

Studio pisania Emacs

Studio pisania Emacs

Praktyczny przewodnik po pisaniu i publikowaniu tekstów

Peter Prevos



Third
Hemisphere
Wydawnictwo

wo Kangaroo Flat, Australia

© Peter Prevos (2025)

Kangaroo Flat, Australia peterprevos.com

ISBN y78-0-y87555y-y-7 (książka w miękkiej oprawie)

Niniejsza praca jest objęta licencją Creative Commons *Attributio-Slave Alike 3.0 Austvalia*.
Możesz swobodnie udostępniać, kopiować, rozpowszechniać i przekazywać tę pracę pod następującymi warunkami:

- ① *Atrybucja*: Należy podać autora utworu w sposób określony przez niego (ale nie w sposób sugerujący, że autor popiera użytkownika lub sposób wykorzystania utworu).
- ② *Slave Alike*: Jeśli zmienisz, przekształcisz lub wykorzystasz tę pracę, możesz rozpowszechniać powstałą pracę wyłącznie na podstawie tej samej lub podobnej licencji.

Napisane w Emacs Org Mode. Składane w LaTeX.

Spis treści

Przedmowa	vii
Przedmowa	xi
1 Wprowadzenie	1
2 Dlaczego Emacs?	5
ç Pierwsze kroki z Emacsem	21
4 Korzystanie z Emacs Writing Studio	4 lata
5 Inspiracja: Czytaj, słuchaj i oglądaj	7ç
6 Pomysł: Zapisywanie i zarządzanie pomysłami	y7
7 Produkcja: Przygotowanie manuskryptu	1ç5
8 Publikacja: Podziel się z całym światem	167
y Administracja: Zarządzaj swoimi projektami	1yy

10 Zostań mistrzem Emacs	21y
Dodatek	22y
Referencje	26ç

Przedmowa

Dzięki *Zmaes Wvities Studio* masz wszystko, czego potrzebujesz, aby rozpocząć pisanie. Książka i towarzysząca jej konfiguracja dla Emacsa stanowią solidną podstawę do porządkowania pomysłów, gromadzenia informacji i wyjaśniania swoich myśli.

Książka ta zawiera przegląd możliwości programu Emacs dla pisarzy. Jest napisana w przystępny sposób. Poznasz podstawy, a następnie zostaniesz poprowadzony przez różne aspekty pracy.

Potraktuj *Zmaes Wvities Studio* jako towarzysza w długiej podróży. Pomoże Ci on rozpocząć pracę z Emacsem i będzie dla Ciebie cennym źródłem informacji, gdy będziesz potrzebować wykonać nieco bardziej zaawansowane czynności, ale zapomnisz, jak to zrobić.

Na początku otrzymujesz wyselekcjonowane doświadczenie. Jest to dokładnie to, czego potrzebujesz, aby zabrać się za pisanie: minimalizuje rozpraszające czynniki. Nie traktuj tego jednak jako ograniczenia: nadal korzystasz z Emacsa — potężnego narzędzia, które można przeprogramować lub rozszerzyć, aby uzyskać więcej możliwości w zakresie tekstu i powiązanych wzorców interakcji.

Konfiguracja *Zmaes Wvities Studio* składa się z rozsądnych ustawień domyślnych. Nie będziesz ograniczony do systemu dostosowanego do indywidualnych potrzeb. Jest to standardowe doświadczenie Emacsa i możesz je dowolnie modyfikować według własnych upodobań.

Zawsze możesz znaleźć więcej zasobów, które pomogą Ci rozwiązać każdy problem. Oprócz oficjalnej instrukcji Emacsa odkryjesz bogaty zbiór wiedzy stworzony przez

społeczności. Znajdziesz tam wpisy na blogu, rysunki i prezentacje wideo. Wszystkie one mają zastosowanie do tego, co masz tutaj.

Żadna konfiguracja nie zastąpi pracy, którą włożysz w naukę. Dotyczy to zarówno czasu spędzonego na korzystaniu z Emacsa, jak i ogólnie pisania. Nie oczekuj, że *Zmaes Wvitię Studio* zdziała dla Ciebie cuda: nie sprawi, że z dnia na dzień staniesz się ekspertem w zakresie Emacsa i nie pobudzi Twojej kreatywności bez Twojego udziału.

Jest to narzędzie, które zostało przetestowane i sprawdziło się w praktyce. Jak każde narzędzie, musi być właściwie używane przez osobę posiadającą odpowiednie umiejętności lub gotową je nabyć poprzez ciągłą praktykę. Jako początkujący nie będziesz wiedzieć zbyt wiele: ta książka ma na celu zapewnienie Ci informacji potrzebnych do kontynuowania nauki.

Możesz oczekiwać, że *Zmaes Wvitię Studio* dotrzyma obietnicy dostarczenia zestawu narzędzi do tworzenia kolejnych dzieł. Zrobi to w niezawodny sposób. Reszta zależy od Ciebie: przeprowadź badania i zacznij tworzyć.

Oprócz kwestii technicznych, książka ta skłania do refleksji nad metodami pracy. Czyni to pośrednio poprzez opisywane funkcje. Następnie należy rozważyć, w jaki sposób każda z nich wpisze się w Państwa proces pracy.

Składanie zestawu narzędzi lub procedur jest częścią procesu odkrywania, przez który muszą przejść autorzy. Metodą prób i błędów dowiadujesz się, co działa w Twoim przypadku. Następnie możesz się tego trzymać i zdecydowanie preferować to rozwiązanie zamiast alternatywnych podejść. Nie ma znaczenia, czy Twoje metody są odpowiednie dla innych: wiesz, że nie ma jednego rozwiązania dla wszystkich.

Emacs to idealne narzędzie dla osób, które chcą mieć pełną kontrolę nad tym, co piszą. Jest wysoce konfigurowalny i będzie rozwijał się wraz z użytkownikiem, aby zawsze odpowiadać jego aktualnemu poziomowi. Jeśli kiedykolwiek będziesz miał okazję nauczyć się języka Emacs Lisp, zdasz sobie sprawę, że możesz zrobić naprawdę wiele, aby dostosować każdy najmniejszy szczegół dokładnie tak, jak chcesz.

Kluczem do opanowania obsługi programu Emacs jest cierpliwość i metodyczne podejście. Wypróbuj jedną funkcję na raz, naucz się jej używać, a następnie przejdź do kolejnej. Właśnie dlatego *studio Zmaes Wvitię* jest tak pomocne: pozwala eksperymentować we własnym tempie, oferując jednocześnie gotowe do użycia rozwiązania.

Pamiętaj, że dopiero zaczynasz i będziesz tu długo. Powodzenia!

Protesilaos „Prot” Stavrou

Przedmowa

Moja przygoda z pisaniem rozpoczęła się na początku XXI wieku od papierowych dzienników, w których zapisywałem swoje przypadkowe myśli i relacje z podróży. Papierowe zeszyty są świetne i nadal używam ich jako podstawowego miejsca do gromadzenia myśli. Jedyną wadą papierowych notatników jest to, że znalezienie starych informacji i połączenie ich ze sobą wymaga dużo pracy. Oprócz notatników zgromadziłem również archiwum papierowe, bibliotekę i kolekcję fotografii. Jako entuzjasta komputerów od tamtej pory szukałem idealnego narzędzia cyfrowego, które pozwoliłoby mi połączyć wszystkie te informacje w doskonały system zarządzania wiedzą osobistą.

Przez lata wypróbowałem wiele aplikacji komputerowych, aby osiągnąć ten idealny stan. Chciałem czegoś, co przechowuje informacje w trwałym formacie, niezależnym od marki oprogramowania. Idealnie byłoby, gdyby można było w nim wyszukiwać informacje i gdyby była to jedna aplikacja do wykonywania różnych zadań. Kiedyś przeskakiwałem z aplikacji do aplikacji, przechodziłem od listy zadań do harmonogramu, a następnie przechodziłem do edytora tekstu, arkusza kalkulacyjnego, czytnika plików PDF itp. Zarządzałem złożoną siecią narzędzi programowych, aby zrealizować jeden projekt. Czy nie byłoby miło, gdyby istniał jeden program, który pomagałby w realizacji wszystkich zadań?

Moje poszukiwania w Internecie zawsze wskazywały na Emacs. Próbowałem go używać na początku XXI wieku, ale szybko zrezygnowałem ze względu na stromą krzywą uczenia się. Około osiem lat temu spróbowałem ponownie i tym razem pomogły mi liczne strony internetowe i filmy instruktażowe, dzięki którym udało mi się

rozpocząłem. W kolejnych latach studiowałem Emacsa i skonfigurowałem go do pisania i publikowania artykułów oraz książek. W miarę zgłębiania możliwości Emacsa odkryłem nie tylko potężne narzędzie do pisania, ale także tętniącą życiem społeczność entuzjastów i programistów, którzy nieustannie przyczyniają się do jego rozwoju. To poczucie przynależności wzbogaciło moje doświadczenie i zainspirowało mnie do przekraczania granic tego, co mogę osiągnąć dzięki Emacsowi. Emacs jest dla mnie jak fabryka czekolady Willy'ego Wonki, gdzie prawie wszystko jest możliwe.

Opracowałem *Zmaes Wviteg Studio* (EWS), aby zaspokoić moje konkretne potrzeby związane z pisaniem tej książki i innymi projektami. Moje zaangażowanie w tworzenie konfiguracji EWS, filmów wideo na stronie internetowej i tej książki wynika z chęci podzielenia się korzyściami płynącymi z mojej podróży z innymi autorami. EWS to nie tylko narzędzie, ale także zwieńczenie lat nauki i eksperymentów. Moim celem jest napisanie książki, którą sam chętnie przeczytałbym, gdy po raz pierwszy zetknąłem się z Emacssem.

Ta książka, będąca zarówno przewodnikiem po EWS, jak i produktem jego możliwości, została rygorystycznie przetestowana i udoskonalona, aby spełniała wymagania dużych projektów pisarskich.

Emacs jest znany ze swojej stromej krzywej uczenia się. Nie pozwól, aby ta reputacja zniechęciła Cię do wypróbowania tego systemu. Głównym celem tej książki jest spłaszczenie krzywej uczenia się poprzez zapewnienie skonfigurowanego systemu jako punktu wyjścia. Przedsiębiorca i autor Seth Godin napisał niedawno na swoim blogu:

Przyjęliśmy podejście „zbyt zajęci, żeby się uczyć”. W rezultacie preferujemy narzędzia, które dają nam szybkie rezultaty, a nie te, których warto się nauczyć.

Podziękowania

Rozwój i utrzymanie złożonego systemu, takiego jak Emacs, jest świadectwem siły społeczności. Jestem głęboko wdzięczny niezliczonym wolontariuszom, którzy przyczynili się do powstania kodu jego podstawowego systemu i bogactwa pakietów. To właśnie ich wspólny wysiłek sprawia, że Emacs jest kwitnącym ekosystemem.

Motywacją do napisania tej książki była inspirująca praca takich osób, jak Protesilaos (Prot) Stavrou, David Wilson z *System Cvaftvs* i wielu innych, którzy pomogli mi spłaszczyć krzywą uczenia się podczas mojej przygody z Emacsem poprzez swoje blogi i filmy online.

Użytkownicy Mastodona i Emacsa Harold Kirsch, Thomas Montfort, Ben Finney, Antonio Simón (Quijote Libre), Bob Irving i Frédéric Vachon zrecenzowali wczesne wersje książki. Ich opinie pomogły uczynić tę książkę zarówno łatwiejszą w czytaniu, jak i bardziej kompleksową, co jest trudnym do osiągnięcia balansem. Dogłębne pytania Erin pomogły mi poprawić wyjaśnienia zawarte w tej książce.

Podziękowania dla Olivera Romana, Stefana Kangasa, Fredrika Salomonssona i Tomasza Hołubowicza za poprawienie literówek.

Peter Prevos
Kangaroo flat, 2025

Rozdział 1

Wprowadzenie

Na przestrzeni tysięcy lat pisanie przeszło od tabliczek glinianych i rzeźbienia w kamieniu do papieru, komputerów, a ostatnio do sztucznej inteligencji. Przed pojawieniem się pisania elektronicznego autor potrzebował jedynie notesu, pióra i maszyny do pisania. Bycie autorem w erze cyfrowej jest pod wieloma względami bardziej złożone niż w czasach gliny, kamienia czy papieru.

Współcześni pisarze są często przytłoczeni mnogością narzędzi elektronicznych, z których każde pełni określoną funkcję. Informacje są rozproszone na różnych platformach i przechowywane w niekompatybilnych formatach, co prowadzi do złożonego i czasochłonnego procesu pisania. Ulga od tej złożoności jest jedną z kluczowych zalet programu Emacs.

Niestety, poprzedni akapit opisuje sposób, w jaki wielu studentów, autorów, badaczy i innych pisarzy zarządza swoją pracą. A co, gdybyś mógł zrobić to inaczej i używać tego samego programu do tworzenia prac pisemnych, od pomysłu do publikacji? Ta książka przedstawia Emacs, potężne narzędzie, które może pomóc we wszystkim, od zainicjowania początkowych pomysłów po opublikowanie gotowego artykułu, książki lub strony internetowej. Emacs pozwala porzucić zęgotowanie oprogramowaniem i skupić się na pisaniu.

Ta książka jest przeznaczona dla pisarzy, którzy chcą uwolnić się od ograniczeń tradycyjnych narzędzi do pisania i skorzystać z bardziej elastycznego, wydajnego i dostosowanego do indywidualnych potrzeb środowiska.

Niezależnie od tego, czy jesteś pisarzem tworzącym skomplikowane opowieści fabularne, czy badaczem opracowującym szczegółowe publikacje non-fiction, *Zmaes Wvities Studio* (EWS) oferuje kompleksowy przewodnik, który pomoże Ci zmienić proces pisania.

1.1 Pisarze fikcji

Jako pisarz beletrystyczny Twoja wyobraźnia jest Twoim najcenniejszym atutem. Potrzebujesz narzędzia, które nadażą za płynnością Twoich pomysłów, pozwalając Ci skupić się na tworzeniu bogatych postaci, skomplikowanych fabuł i żywych światów bez rozpraszania uwagi. Dzięki Emacs możesz utrzymywać uporządkowane notatki, szkice i szkice postaci w jednym środowisku.

Potężne funkcje wyszukiwania i zamiany w połączeniu z konfigurowalnymi szablonami pozwalają w łatwy sposób zachować spójność nazw postaci, lokalizacji i innych kluczowych szczegółów w całym manuskrypcie. Kiedy pojawia się natchnienie, minimalistyczny interfejs Emacsa pozwala od razu zabrać się do pisania bez rozpraszania uwagi zbędnymi elementami graficznymi.

Neal Stephenson, amerykański pisarz zajmujący się fantastyką naukową, nazywa Emacsa „termonuklearnym edytorem tekstu” (Stephenson, 1yyy), nie w sensie, że niszczy on Twoje teksty w kuli ognia, ale w odniesieniu do jego ogromnej mocy w porównaniu z konwencjonalnymi narzędziami do pisania.

1.2 Autorzy literatury faktu

Pisanie literatury faktu wymaga precyzji, organizacji i skrupulatnych badań. Niezależnie od tego, czy pracujesz nad rozprawą doktorską, raportem technicznym, czy biografią historyczną, Emacs zapewnia narzędzia do efektywnego zarządzania ogromnymi ilościami informacji. Możesz efektywnie porządkować swoje notatki badawcze, zarządzać bibliografią i tworzyć konspekty swojej pracy.

Emacs pozwala na płynną integrację tych elementów w jednej aplikacji, zapewniając, że wszystkie wyniki badań są pod ręką podczas tworzenia szkicu pracy. Możliwość eksportowania

manuskryptu do różnych formatów, takich jak PDF lub HTML, gwarantuje, że Twoja praca jest gotowa do publikacji lub udostępnienia innym bez konieczności konwersji formatów.

1.4 Wielozadaniowość pisarzy

W dzisiejszym szybko zmieniającym się świecie wielu pisarzy zarządza wieloma projektami. Emacs jest idealnym narzędziem do zarządzania różnorodnym portfolio pisarskim.

Możesz bez wysiłku przełączać się między różnymi projektami, czy to opowiadaniem, pracą naukową, czy nawet codziennym dziennikiem. Potężne funkcje zarządzania zadaniami Emacsa pomagają śledzić terminy i zadania oraz monitorować postępy we wszystkich przedsięwzięciach związanych z pisaniem.

1.4 Rozproszeni pisarze

Jeśli łatwo rozpraszają Cię funkcje nowoczesnego oprogramowania do pisania, Emacs oferuje odświeżającą zmianę. Dzięki skupieniu się na zwykłym tekście i poleceniach sterowanych klawiaturą, Emacs zapewnia środowisko wolne od rozpraszających czynników, w którym liczą się tylko Twoje słowa, pozwalając Ci skupić się na pisaniu bez zakłóceń.

Pisanie w Emacs przenosi Cię z powrotem do czasów, kiedy między pisarzem a napisanymi słowami stała maszyna do pisania. Ale Emacs ma też potężne możliwości edycji, więc nie ma potrzeby używania korektora.

1.5 Ciekawscy pisarze

Wreszcie, ta książka jest przeznaczona dla tych, którzy nie tylko szukają narzędzia, ale także możliwości nauki i rozwoju jako użytkownicy technologii. Emacs to nie tylko oprogramowanie, to sposób na życie. Ucząc się korzystać z jego potężnych funkcji, rozwijasz nowe umiejętności, które usprawniają proces pisania i ogólną produktywność, inspirując Cię do odkrywania nowych możliwości w swojej pisarskiej podróży.

Emacs został pierwotnie stworzony dla programistów, ale nie trzeba być geniuszem komputerowym, aby korzystać z niego jako autor. Ta książka ułatwia naukę obsługi programu, pomagając w pełni wykorzystać potencjał Emacsa jako najlepszego pomocnika w pisaniu.

Rozdział 2

Dlaczego Emacs?

Oficjalnym sloganem Emacsa jest to, że jest to „rozszerzalny, samodokumentujący się edytor tekstu”. Te nieco niejasne słowa nie oddają w pełni zalet Emacsa, ponieważ skupiają się na jego pierwotnym przeznaczeniu jako narzędzia do tworzenia oprogramowania. Emacs to wielofunkcyjne środowisko komputerowe, które może pomóc w zarządzaniu informacjami, śledzeniu projektów, pisaniu i publikowaniu artykułów, książek, stron internetowych oraz wszelkich innych czynnościach związanych z tekstem. Emacs nie jest hackiem zwiększającym produktywność; jest systemem hakującym produktywność. Emacs jest szwajcarską piłą łańcuchową wśród narzędzi zwiększających produktywność.

Rozwój systemów Emacs (skrót od „editor macros”) rozpoczął się w latach 70. Tak stare oprogramowanie może wydawać się przestarzałe, ale prężna społeczność programistów nieustannie ulepsza ten system. Rozszerzalność Emacsa oznacza, że użytkownicy mogą konfigurować go zgodnie z własnymi potrzebami. Konfiguracja Emacsa instruuje system, aby zachowywał się zgodnie z Twoimi oczekiwaniami, np. w zakresie skrótów klawiaturowych i dodatkowych funkcji. Emacs jest również rozszerzalny dzięki tysiącom bezpłatnych pakietów. Pakiet Emacsa to wtyczka, która dodaje nowe możliwości do systemu lub ulepsza istniejące, podobnie jak aplikacja w telefonie.

Na przestrzeni dziesięcioleci powstało wiele wersji programu Emacs. Obecnie najczęściej używaną wersją jest GNU Emacs, po raz pierwszy wydana przez Richarda Stallmana w 1984 roku (Johnson, 2022;

Stallman, 1y81b). GNU Emacs (zwany dalej Emacs) to darmowe oprogramowanie wydane przez fundację Free Software Foundation. Darmowe oprogramowanie jest czasami nazywane „oprogramowaniem libre”, aby podkreślić aspekt wolności ponad wartość pieniężną. Fundacja luźno definiuje darmowe oprogramowanie jako:

„Wolne oprogramowanie” oznacza oprogramowanie, które szanuje wolność użytkowników i społeczności. Ogólnie rzecz biorąc, oznacza to, że użytkownicy mają swobodę uruchamiania, kopiowania, rozpowszechniania, badania, modyfikowania i ulepszania oprogramowania. Zatem „wolne oprogramowanie” jest kwestią wolności, a nie ceny. Aby zrozumieć tę koncepcję, należy myśleć o „wolności” w znaczeniu „wolności słowa”, a nie „darmowego piwa”.

Emacs jest oficjalnie edytorem tekstu, ale dla autorów nie ma to sensu. Z punktu widzenia autora Emacs jest *edytorem* tekstu, ponieważ edycja jest tylko jednym z etapów procesu pisania. Edytor tekstu jest narzędziem dla programistów do pisania kodu, a edytor tekstu jest narzędziem dla autorów do pisania prozy. EWS to specjalnie dostosowana konfiguracja, która przekształca Emacsa w narzędzie dla badaczy i autorów.

2.1 Dlaczego warto używać Emacsa?

Podczas pracy nad projektem pisarskim autorzy potrzebują zestawu narzędzi, które pomogą im wykonać zadanie. Robią notatki w narzędziu badawczym i skrupulatnie tworzą bibliografie w bazie danych. Następnie piszą w znanym sobie edytorze tekstu. Aby dotrzymać terminów, korzystają z narzędzia zwiększającego produktywność do zarządzania projektami. W końcu, po wielu godzinach skupionej pracy, badacz może zrelaksować się, grając w Tetrisa, aby odpocząć od aplikacji.

Problem z tym dobrze znanym sposobem pracy polega na tym, że każdy program wymaga opanowania nowych umiejętności, zrozumienia innej logiki działania i dostosowania się do z góry ustalonego sposobu pracy. Większość oprogramowania jest nieelastyczna i zmusza użytkownika do dostosowania się do wizji twórców programu, poza kilkoma opcjami konfiguracyjnymi dostępnymi w postaci listy pól wyboru.

Emacs oferuje rewolucyjne podejście. Możesz pisać notatki badawcze, zarządzać bibliografią, a nawet grać w Tetrisa — wszystko w jednym, zunifikowanym środowisku. Wyobraź sobie wygodę opanowania jednego zestawu poleceń zamiast zmagania się z wieloma programami. Emacs umożliwia konfigurację i dostosowanie go do własnych preferencji, przekształcając go z zwykłego narzędzia do pisania w rozszerzenie osobistego przepływu pracy. Porzuć zonglowanie oprogramowaniem i skup się na tym, co naprawdę ważne: pisaniu.

To stwierdzenie może być mylące, ponieważ Emacs wymaga pomocy innego oprogramowania, aby funkcjonować jako studio pisarskie. Emacs jest również interfejsem do innego wolnego oprogramowania. Konieczne będzie zatem zainstalowanie dodatkowego oprogramowania, aby Emacs mógł odczytywać i eksportować pliki do formatów binarnych, takich jak PDF oraz pliki audio lub wideo. Emacs korzysta również z zewnętrznego oprogramowania do sprawdzania pisowni, zaawansowanego wyszukiwania i generowania diagramów.

Chociaż Emacs różni się od nowoczesnego oprogramowania graficznego, które cieszy oko, jego pozorna prostota ma swoje uzasadnienie. Nie daj się zwieść surowej fasadzie. Pod powierzchnią kryje się solidne i starannie zaprojektowane współczesne środowisko komputerowe, które można przekształcić w narzędzie do pisania bez rozpraszania uwagi.

Kolejną zaletą jest długowieczność tego narzędzia. Sposób, w jaki korzystasz z Emacsa teraz, będzie taki sam za kilkadziesiąt lat. Czytanie podręcznika Emacsa z 1981 roku jest prawie jak czytanie najnowszej wersji, ponieważ podstawowe funkcje uległy jedynie niewielkim zmianom (Stallman, 1981a).

Wielu pisarzy narzekało na ograniczenia komercyjnych edytorów tekstu podczas pracy z dużymi dokumentami. Korzystanie z tego oprogramowania może być frustrującym doświadczeniem. Programy te zostały opracowane po raz pierwszy, gdy świat rządziły papierowe notatki i raporty, i od tego czasu niewiele się zmieniły. Oprogramowanie graficzne emuluje drukowane dokumenty papierowe, mimo że większość ludzi pisze na nośnikach elektronicznych. Emacs wyłamuje się z tego paradygmatu, oddzielając treść od projektu. To wyzwalające podejście pozwala skupić się na tworzeniu pomysłów bez zagłębiania się w projekt produktu końcowego. Dodatkową zaletą Emacsa jest to, że może on bez wysiłku przekształcić ten sam plik tekstowy w gotowy do druku plik PDF, stronę internetową lub e-book.

2.2 Oprogramowanie plastyczne

Emacs jest platformą „plastycznego oprogramowania”, co oznacza, że można dowolnie zmieniać i ulepszać sposób jego działania. Podstawową zasadą plastycznego oprogramowania jest łatwość wprowadzania zmian. Zaawansowani użytkownicy Emacsa mogą tworzyć aplikacje dostosowane do własnych potrzeb, korzystając z wersji języka LISP dla Emacsa, zwanego również Elisp (Monnier & Sperber, 2020). Zadanie to może wydawać się zniechęcające, ale chodzi tu o samą możliwość. Nowi użytkownicy Emacsa mogą skonfigurować prawie wszystko w systemie bez znajomości Elisp.

W niniejszej książce przedstawiono wersję Emacsa bez kodowania. Ostatni rozdział i dodatek zawierają wskazówki dotyczące rozpoczęcia korzystania z Elisp, ale jako autor możesz korzystać z Emacsa bez pisania kodu.

Aby korzystać z bardziej zaawansowanych aplikacji, konieczne będzie opanowanie języka Emacs Lisp. Wymóg ten może wydawać się przeszkodą, ale znajomość tego języka zapewni niemal nieograniczoną kontrolę nad sposobem użytkowania komputera. To oprogramowanie powinno dostosowywać się do użytkownika, a nie odwrotnie. Większość użytkowników Emacsa dzieli się tym, co stworzyli, więc możesz swobodnie kopiować ich prace. Możesz również rozszerzać i konfigurować Emacsa za pomocą jednego z tysięcy dostępnych bezpłatnie pakietów. EWS to wyselekcjonowany zbiór takich pakietów, dostosowany do potrzeb autorów.

Zaletą tego podejścia jest całkowita swoboda korzystania z oprogramowania. Można mu zlecić wykonanie niemal każdej czynności i skonfigurować je zgodnie z własnymi potrzebami, o ile można to zrobić za pomocą tekstu. Wadą jest to, że wymaga ono innego podejścia niż współczesne oprogramowanie. Korzystanie z Emacsa przywraca pierwotny cel używania komputera i prawdziwą łatwość obsługi. Czy jesteś gotowy, aby zmienić sposób korzystania z komputera? Parafrazując słynną scenę z filmu Matrix:

Jeśli weźmiesz niebieską pigułkę Microsoftu, historia się kończy i wszystko pozostaje bez zmian. Jeśli weźmiesz fioletową pigułkę Emacsa, pozostaniesz w Krainie Czarów, a ja pokażę ci, jak głęboka jest królicza nora.

2.ç Redefining user-friendliness

Brak eleganckiego interfejsu graficznego w Emacsie może zniechęcać nowych użytkowników. Niestety, większość ludzi myli łatwość obsługi z eleganckim wyglądem i możliwością korzystania z myszki. Jednak podejście graficzne wcale nie jest przyjazne dla użytkownika, ponieważ ogranicza swobodę. Oprogramowanie oparte na grafice jest jak złota klatka. Praca w nim może być przyjemna, ale nadal pozostaje klatką.

Emacs to edytor tekstu, który skupia się na znaczeniu semantycznym znaków na ekranie, a nie na tym, jak ostatecznie będą one wyglądały na stronie lub ekranie. Tekst zwykły nie jest tym samym, co zwykły język angielski; odnosi się on do sposobu przechowywania informacji. Tekst zwykły jest przeciwieństwem tekstu sformatowanego, który ukrywa definicje rozmiarów czcionek, kolorów i innych atrybutów.

Tekst zwykły najczęściej ma rozszerzenie .txt i nie zawiera żadnych formatowań, takich jak pogrubienie. Użytkownicy systemu Windows mogą znać znane oprogramowanie Notatnik. Istnieją jednak inne formaty tekstu zwykłego, takie jak HTML, Markdown, LaTeX i Org, które oferują szeroki zakres możliwości przekształcania tekstu zwykłego w dzieło sztuki.

Zwykły tekst można odczytać na wszystkich systemach komputerowych, więc nie musisz się martwić, że Twoje teksty zostaną zablokowane w zastrzeżonym formacie lub że będziesz skazany na korzystanie z konkretnego pakietu oprogramowania. Wszystko, co napiszesz w Emacsie, można odczytać za pomocą Notatnika, TextEdit lub innego podobnego oprogramowania. Jediną różnicą jest to, że inne programy nie mają takiej wszechstronności jak Emacs. Zwykły tekst nie jest aplikacją niszową. Format ten pozostaje zasadniczo niezmieniony od dziesięcioleci i jest mało prawdopodobne, aby zniknął w przyszłości.

Tryby tekstowe mogą również wyświetlać „grafikę”. Kiedy w latach 70. chodziłem do szkoły podstawowej, nasz nauczyciel z dumą pokazał nam kilka przykładów grafiki komputerowej. Grafika składała się z wydrukowanych znaków alfanumerycznych, które przypominały obraz, na przykład tego kotka (źródło: asciart.eu).

```
/\_/\ (
o o )
==_Y_==
"-"
```

Niektórzy nadal tworzą sztukę ASCII, jak się ją nazywa, aby ozdobić kod komputerowy. Nie ma jednak potrzeby uciekania się do tych starych technik, ponieważ Emacs może również wyświetlać obrazy w standardowych formatach, takich jak PNG i JPG.

Interfejsy graficzne symulują świat fizyczny, sprawiając, że obiekty na ekranie wyglądają jak kartki papieru i teczki na biurku. Wskazujesz, klikasz i przeciągasz dokumenty do folderów; dokumenty wyglądają tak, jak na papierze, a po zakończeniu są wysyłane do kosza. Interfejsy graficzne to magiczna sztuczka, która sprawia, że użytkownik ma wrażenie, iż wykonuje czynności fizyczne (Tognazzini, 1yy3). Takie podejście może być wygodne, ale uniemożliwia zrozumienie działania komputerów. W edytorach tekstu ekran wygląda jak wydrukowana strona. Chociaż może to być estetyczne, odwraca uwagę autora od tworzenia treści i motywuje go do zajmowania się formatowaniem.

Oprogramowanie graficzne działa zgodnie z zasadą „*Włat You See is Włat You Get*” (WYSIWYG). Oznacza to, że na ekranie wyświetla się dokument przypominający stronę papierową. Technika ta ma znaczenie tylko w przypadku tworzenia dokumentów przeznaczonych do druku. Jednak obecnie tylko niewielka część tekstów elektronicznych jest tworzona z myślą o druku, więc podejście WYSIWYG nie ma większego sensu w erze cyfrowej.

Podejście graficzne odwraca uwagę od treści. Skłania użytkownika do edycji stylu zamiast pisania tekstu. Tekst w oprogramowaniu WYSIWYG nazywany jest tekstem sformatowanym, ponieważ obejmuje zarówno treść, jak i wygląd. Instrukcje formatowania w tekście sformatowanym są niewidoczne dla użytkownika, co może powodować problemy z uzyskaniem pożądanego efektu końcowego. Pracownicy biurowi na całym świecie tracą mnóstwo czasu na formatowanie i składanie dokumentów w środowiskach graficznych.

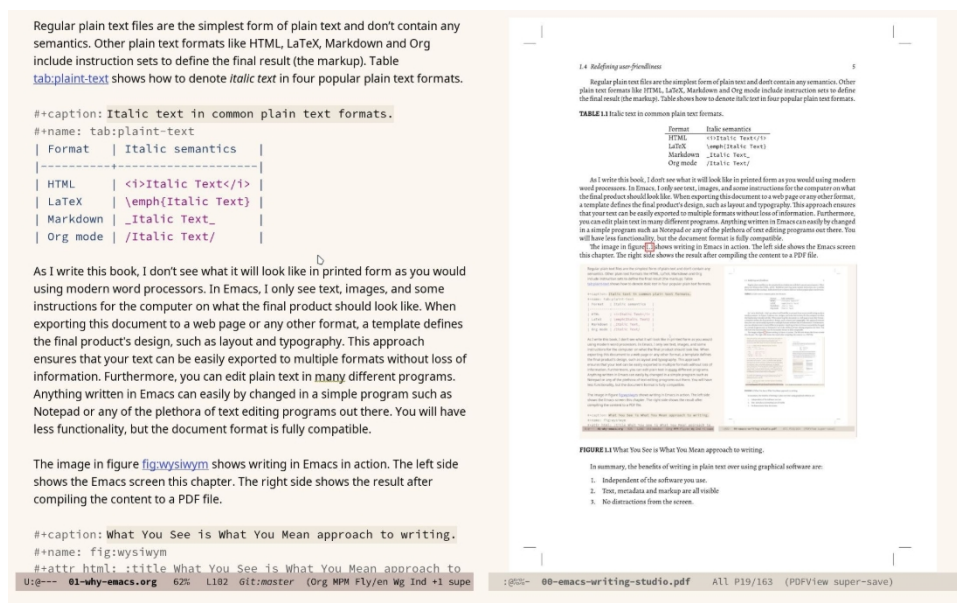
Zwykły tekst wykorzystuje podejście „*Włat You See is Włat You Meae*” (WYSIWYM). Zamiast skupiać się na projekcie dokumentu, edytor WYSIWYM zachowuje zamierzone znaczenie każdego elementu. Sekcje, akapity, ilustracje i inne elementy dokumentu są oznaczane jako takie przy użyciu różnych konwencji (Khalili & Auer, 2015). W zwykłym tekście treść i semantyka są bezpośrednio widoczne i mogą być zmieniane przez użytkownika.

Zwykłe pliki tekstowe są najbardziej podstawowym formatem i nie zawierają żadnych elementów semantycznych. Inne formaty tekstu zwykłego, takie jak HTML, LaTeX (wymawiane jako „*la-teet*”, pochodzące

od greckiego słowa *typos*), Markdown i Org, zawierają zestawy instrukcji służące do definiowania wyniku końcowego (znaczników). Tabela 2.1 pokazuje, jak oznaczyć *tekst kursywą* w czterech popularnych formatach zwykłego tekstu.

Tabela 2.1: Tekst kursywą w formatach zwykłego tekstu.

format	Semantyka kursywy
HTML	<code><i>Tekst kursywą</i></code>
LaTeX	<code>\emph{Tekst kursywą}</code>
Markdown	<code>_Tekst kursywą_</code>
Tryb Org	<code>/Tekst pochylony/</code>



Rysunek 2.1: To, co widzisz, jest tym, co zatwierdzasz.

Korzystanie z czystego tekstu pomaga zwiększyć produktywność, ponieważ nie musisz martwić się o wygląd dokumentu, dopóki nie ukończysz jego treści. Główną zaletą korzystania z czystego tekstu w porównaniu z tekstem sformatowanym jest to, że zapewnia on środowisko do pisania wolne od rozpraszających elementów. Pisząc tę książkę, nie widzę, jak będzie wyglądała w formie drukowanej, tak jak w przypadku korzystania z nowoczesnych edytorów tekstu.

sors. Emacs wyświetla tylko tekst, obrazy i niektóre instrukcje dotyczące wyglądu produktu końcowego. Podczas eksportowania tego dokumentu do strony internetowej lub innego formatu szablon określa układ i typografię produktu końcowego. Takie podejście gwarantuje, że tekst można łatwo eksportować do wielu formatów. Obrazek na rysunku 2.1 pokazuje pisanie w Emacsie w akcji. Po lewej stronie widoczny jest ekran Emacs'a z fragmentem tego rozdziału. Po prawej stronie widoczny jest wynik po skompilowaniu treści do formatu PDF.

Wskaźniki zwykłego tekstu dla tabel, podpisów, odniesień itp. są konwertowane do odpowiedniego formatu, zgodnie z definicją szablonu. Takie podejście pozwala skupić się na treści. Po zdefiniowaniu szablonu wszystkie operacje związane z składaniem i formatowaniem zostaną wykonane przez funkcję eksportu. Podobne wyniki można osiągnąć w przypadku dokumentów HTML (w tym e-booków) i formatów edytorów tekstu.

Podsumowując, zalety pisania w zwykłym tekście w porównaniu z używaniem oprogramowania graficznego są następujące:

1. Niezależność od używanego oprogramowania.
2. Tekst, metadane i znaczniki są widoczne.
3. Brak elementów rozpraszających uwagę na ekranie.
4. Możliwość eksportowania do dowolnego formatu.

2.4 Krzywa uczenia się

Emacs ma stromą krzywą uczenia się ze względu na ogromną liczbę możliwych konfiguracji. Aby Emacs działał zgodnie z Twoimi oczekiwaniami, musisz nauczyć się podstawowych zasad i niektórych powiązanych pakietów dodatkowych. Emacs jest bardziej złożony niż inne edytory tekstu, ale jest też znacznie potężniejszy niż jakiekolwiek inne narzędzie. Jednak z tą wielką mocą wiąże się wielka odpowiedzialność, więc musisz nauczyć się kilku nowych umiejętności, aby używać go jako narzędzia do pisania.

Celem EWS jest spłaszczenie krzywej uczenia się. Najlepszym podejściem jest dać się przytłoczyć mnogością możliwości i opanować tylko te funkcje, które są

potrzebne do wykonania danego zadania. Nawet bez żadnej konfiguracji Emacs może osiągnąć bardzo wiele.

Metody i słownictwo Emacsa wydają się obce w porównaniu z innymi współczesnymi programami. Głównym powodem tych różnic jest fakt, że prace nad nim rozpoczęto w latach 70., kiedy to informatyka znacznie różniła się od tej, którą znamy obecnie. Słownictwo Emacsa jest reliktem, pozostałością po wcześniejszej epoce w ewolucji informatyki. Na przykład otwieranie pliku to „odwiedzanie pliku”. Wklejanie tekstu to „yanking”, a wycinanie to „killing”. Terminologia Emacsa jest bardziej poetycka niż terminy rzemieślnicze, takie jak wycinanie, wklejanie i przenoszenie plików między folderami, jakby były kawałkami papieru. Różnice te są nie tylko częścią uroku Emacsa, ale także jego mocy. Przekonasz się, że odpowiedniki tych znanych zadań w Emacsie są bardziej wydajne niż te, które są powszechne w nowoczesnym oprogramowaniu. Jednak ta stroma krzywa uczenia się jest warta swojej ceny. Moja osobista mantra brzmi:

Im bardziej stroma krzywa uczenia się, tym większa nagroda.

2.5 Zalety i ograniczenia Emacsa

Podsumowując, oto kilka istotnych zalet korzystania z Emacsa do tworzenia treści pisemnych:

1. Jedno oprogramowanie do wykonywania większości czynności komputerowych zwiększa produktywność, ponieważ wystarczy opanować tylko jeden system.
2. Wszystkie informacje są przechowywane w plikach tekstowych. Nigdy nie będziesz mieć problemów z ezoterycznymi formatami plików.
3. Możesz modyfikować prawie wszystko w oprogramowaniu, aby dostosować je do swojego stylu pracy.
4. Emacs działa na wszystkich głównych systemach operacyjnych: GNU/Linux, Windows, Chrome i macOS.
5. Emacs jest darmowym (wolnym) oprogramowaniem wspieranym przez dużą społeczność chętną do pomocy.

Po wysławianiu zalet tego wielofunkcyjnego edytora można by pomyśleć, że Emacs jest wszechmocnym bogiem oprogramowania. Niektórzy ludzie stworzyli nawet *Cłuveł of Zmaes* jako fałszywą religię, aby wyrazić swój podziw dla tego niezwykle elastycznego środowiska oprogramowania. Pomimo tego podziwu Emacs ma pewne ograniczenia.

Emacs może wyświetlać obrazy i integrować je z tekstem, ale ma ograniczoną funkcjonalność w zakresie tworzenia lub modyfikowania plików graficznych. Jeśli potrzebujesz tworzyć lub edytować obrazy, rozważ użycie GIMP (GNU Image Manipulation Program). Treści wideo nie są obsługiwane, z wyjątkiem hiperłączy do plików lub stron internetowych. Ograniczenia te są usprawiedliwione, biorąc pod uwagę, że podstawową funkcją Emacsa jest przetwarzanie tekstu.

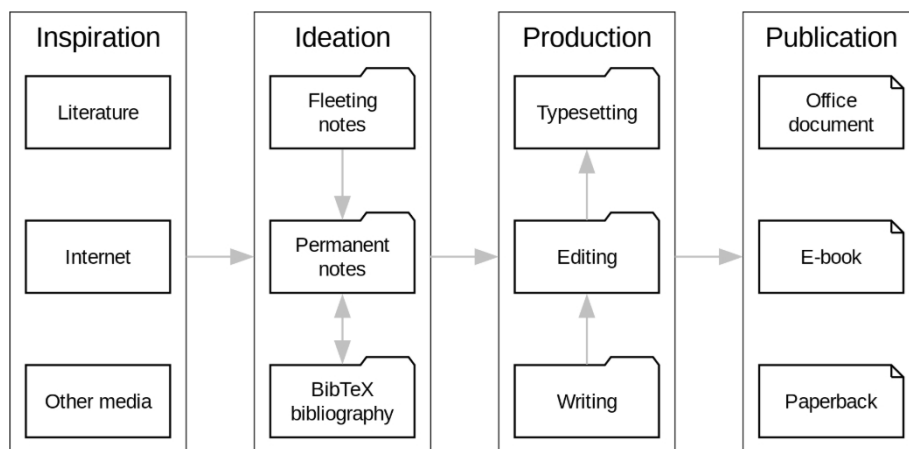
Drugą wadą jest to, że Emacs nie zawiera w pełni funkcjonalnej przeglądarki internetowej. Można surfować po Internecie w ramach Emacsa, ale tylko w ramach ograniczeń interfejsu tekstowego. Czytanie stron internetowych w formacie tekstowym ma również pewne zalety, zapewniając bezpieczne przeglądanie stron bez rozpraszania uwagi.

Wreszcie, Emacs może stać się czynnikiem obniżającym produktywność. To, że można skonfigurować wszystko, nie oznacza, że należy to robić. Nie poświęcaj zbyt wiele czasu *na* dostosowywanie swojego środowiska pracy. Wykorzystaj ten czas na pisanie. Większość trików zwiększających produktywność nie ma istotnego wpływu na wydajność, ponieważ piszesz umysłem, a nie klawiaturą.

2.6 Przepływ pracy w *studiu pisarskim Emacs*

Książka ta przedstawia typowy przebieg pracy naukowców i autorów podczas przygotowywania, pisania i publikowania manuskryptu. W rzeczywistości proces pisania jest zazwyczaj złożony i chaotyczny, ponieważ obejmuje kolejne cykle iteracyjne. Kiedy jednak spojrzymy na niego z dystansu, dostrzeżemy uporządkowany schemat. Czytamy literaturę i czerpiemy inspirację, opracowujemy nowe pomysły, tworzymy nowe prace i publikujemy wyniki. Choć rzeczywistość nigdy nie jest tak linearna, jak sugeruje ta lista, stanowi ona pomocny przewodnik do uporządkowania treści niniejszej książki (rysunek 2.2).

Podstawową zasadą tego procesu jest gromadzenie przez autorów informacji z literatury, Internetu, filmów i innych źródeł (*iespivatioe*), które następnie przetwarzają w notatkach.



Rysunek 2.2: Przepływ pracy w Zmaes Wwities Studio.

Notatki te stanowią centralne repozytorium informacji i inspiracji i mogą być powiązane z bibliografią (*ideatioe*). Pomysły i notatki stanowią podstawę procesu pisania (*pvoduetioe*). Autor ostatecznie publikuje manuskrypt w ostatecznej formie (*publieatioe*).

Są to cztery produktywne fazy typowego projektu pisarskiego, ale należy również uwzględnić pewne koszty dodatkowe. Pod koniec długiego dnia pisania i redagowania autorzy muszą również wykonać pewne *czynności administracyjne* związane z zarządzaniem projektami.

lespivatioe

Pomysły nie pojawiają się w głowie znikąd. Nasze myśli, plany i inspiracje wynikają z naszych doświadczeń życiowych oraz tego, co czytamy, słyszymy lub oglądamy.

Emacs posiada rozbudowane funkcje umożliwiające odczytanie dowolnego formatu tekstowego oraz wyświetlanie plików PDF, e-booków i obrazów. Jednak jako edytor tekstu nie posiada funkcji umożliwiających bezpośrednią pracę z tymi formatami, dlatego Emacs pełni rolę interfejsu do innych programów typu open source. Odtwarzanie podcastów lub oglądanie filmów nie jest możliwe w ramach Emacs'a, ale może on zapewnić interfejs

do integracji z aplikacjami multimedialnymi.

Można również utrzymywać bibliografię w celu uporządkowania i uzyskania dostępu do swojej kolekcji literatury elektronicznej. Emacs może również przeglądać internet w postaci zwykłego tekstu.

Rozdział 5 omawia sposób czytania e-booków, surfowania po Internecie i korzystania z plików multimedialnych za pomocą programu Emacs.

Ideatioe

Wartość przyswajania wszystkich tych nowych pomysłów wynika wyłącznie z konieczności rejestrowania nowych inspiracji. Dlatego też prowadzenie notatek jest niezbędne do ułatwienia procesu tworzenia pomysłów. Notatka może być ulotną myślą lub trwałym pomysłem, który warto zarchiwizować.

Emacs jest idealnym narzędziem do przechowywania notatek w postaci zwykłego tekstu. Dostępnych jest kilka pakietów do zarządzania cyfrowym mózgiem. Ten etap pracy w systemie EWS opiera się na pakiecie Denote autorstwa Protesilaosa (Prot) Stavrou.

Nie musisz stosować żadnych konkretnych metod sporządzania notatek, takich jak *Zettelkasten* czy *Bullet Journal*. Moja osobista kolekcja notatek to pierwotna mieszanka pomysłów, skategoryzowana za pomocą organicznie tworzonych tagów i oportunistycznie łączących się plików. Oprócz cyfrowych przemyśleń, do Denote można dodawać wszystko, co warto zachować, w tym pliki binarne, takie jak PDF-y lub zdjęcia.

W rozdziale 5 omówiono, jak korzystać z Org i pakietu Denote w celu opracowania osobistego systemu zarządzania wiedzą.

Produetioe

Kiedy już uporządkujesz swoje myśli, czas zacząć pisać. Org jest idealny do pisania artykułów i książek lub tworzenia stron internetowych. Twórcy Emacs'a opublikowali również wiele dodatkowych narzędzi ułatwiających proces pisanie, w tym autouzupełnianie, sprawdzanie gramatyki, słownik, tezaurus i inne niezbędne narzędzia. Podczas produkcji możesz również chcieć współpracować z innymi autorami, co wymaga pewnej kontroli nad różnymi wersjami.

Rozdział 7 opisuje, jak używać Org do pisania artykułów, stron internetowych i książek oraz zarządzania dużymi projektami.

Publikacja

Nadszedł wspaniały moment, w którym możesz opublikować efekty swojej pracy. Org posiada zaawansowane funkcje eksportowania tekstu do różnych formatów, w tym przede wszystkim do dokumentów edytora tekstu w celu udostępnienia, plików PDF do książek drukowanych, ePub do e-booków, HTML do stron internetowych oraz ODT do dokumentów korporacyjnych.

Org eksportuje pliki do gotowych do druku plików PDF za pomocą systemu przygotowywania dokumentów LaTeX, który jest popularny wśród autorów technicznych i wydawców, ale może być używany do każdej fizycznej książki. Zaletą korzystania z Org w porównaniu z innymi edytorami LaTeX jest to, że nie trzeba znać składni LaTeX, aby rozpocząć pracę.

W rozdziale 8 omówiono, jak używać Org do konwersji dokumentu tekstowego na publikację elektroniczną lub fizyczną, którą można udostępnić światu.

Administracja

Praca nad projektem pisarskim to fantastyczna podróż twórczej ekspresji, ale wiąże się również z pewnymi nakładami pracy związanymi z zarządzaniem projektami. Emacs współpracuje z innym oprogramowaniem GNU, pomagając w zarządzaniu plikami za pomocą potężnego edytora katalogów (Dired). Emacs może być również używany do porządkowania zdjęć i obrazów dzięki wbudowanemu pakietowi Image-Dired.

Wreszcie, praca nad dużym projektem oznacza śledzenie wielu zadań. Org posiada w pełni funkcjonalny system zarządzania zadaniami, który pomaga śledzić postępy w realizacji projektów. Można wdrożyć własny system pracy lub skorzystać z podejścia Getting Things Done (GTD).

Rozdział y omawia sposoby zarządzania plikami i projektami, które pomogą Ci śledzić postępy w realizacji projektów pisarskich.

2.7 Jak czytać tę książkę

Ta książka nie jest technicznym podręcznikiem Emacsa, ale przewodnikiem dla autorów. Opisuje typowe przypadki użycia związane z badaniami, pisanem i publikowaniem oraz sposoby ich realizacji za pomocą Emacsa. Każdy rozdział zawiera odniesienia do obszernej wbudowanej pomocy, dzięki czemu nieustraszeni czytelnicy mogą zgłębić zawartość książki w bardziej szczegółowy sposób. Wiedza zawarta w tej książce wystarczy, aby rozpocząć projekt pisarski, a sam Emacs zawiera całą dokumentację potrzebną do zostania mistrzem klawiatury.

W następnym rozdziale wyjaśniono zasady korzystania z nieskonfigurowanego systemu GNU Emacs, aby ułatwić rozpoczęcie nauki. Jednak żaden użytkownik Emacsa nie korzysta z oprogramowania w stanie nieskonfigurowanym. Konfiguracja EWS zmienia wygląd i funkcjonalność Emacsa, wprowadzając ulepszenia ułatwiające wyszukiwanie istotnych informacji. EWS wdraża również niektóre pakiety Emacsa (wtyczki) wspomagające autorów, takie jak Citar do uzyskiwania dostępu do bibliografii i Denote do robienia notatek. Rozdział 4 wyjaśnia zasady konfiguracji EWS.

Rozdziały od piątego do ósmego opisują typowy przebieg pracy nad projektem pisarskim: od badań, przez pisanie, aż po publikację. Rozdziały te pokazują, jak czytać artykuły, książki i strony internetowe (rozdział 5) oraz jak przekształcić Emacsa w osobisty system zarządzania wiedzą (rozdział 5). Rozdziały te wyjaśniają również, jak przygotować manuskrypty do publikacji (rozdział 7) i eksportować je do różnych formatów (rozdział 8).

Rozdział 9 dotyczy zadań administracyjnych, w tym zarządzania projektami i plikami. Pakiet Org mode oferuje zaawansowane funkcje ułatwiające zarządzanie kalendarzem i listami zadań, co pozwala zachować wysoką wydajność pracy. W ostatniej części tego rozdziału pokazano, jak Emacs może zarządzać plikami. Praca nad dużymi projektami niewątpliwie wiąże się z kopiowaniem, wklejaniem, zmianą nazw i wykonywaniem innych zadań związanych z plikami komputerowymi. Pakiet Dired (Di-rectory Editor) zapewnia wszystko, co jest potrzebne do realizacji tych zadań.

Ostatni rozdział 10 zawiera porady dotyczące tego, jak stać się mistrzem Emacsa, przedstawiając wskazówki dotyczące pogłębiania wiedzy, w tym krótkie wprowadzenie do języka Emacs Lisp.

W dodatku do książki znajduje się opatrzona komentarzami konfiguracja EWS wraz z instrukcjami dotyczącymi interpretacji i modyfikacji kodu zgodnie z konkretnymi potrzebami.

Najlepszym sposobem na przeczytanie tej książki jest usiąść przed komputerem i wypróbować opisane w niej rozwiązania. Eksperymentuj z różnymi opcjami, utwórz kilka plików i po prostu się baw.

Rozdział 3

Pierwsze kroki z Emacsem

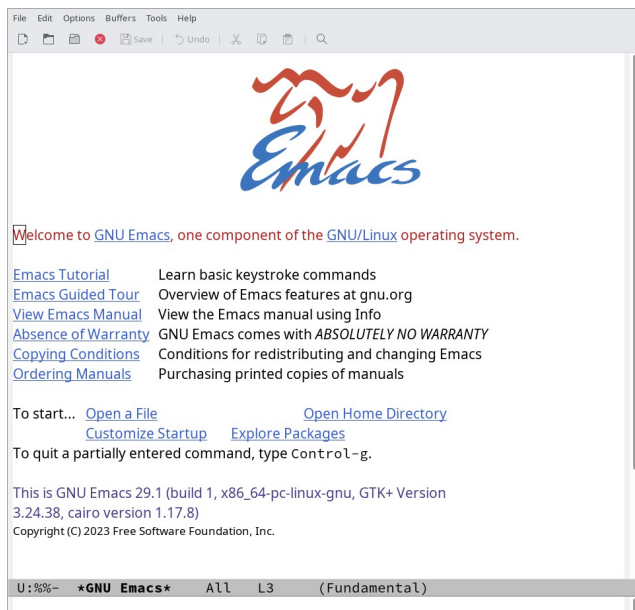
Odpalcie silniki — czas zacząć korzystać z Emacs. W tym rozdziale przedstawiono sposób korzystania z Emacsa bez żadnej konfiguracji, znanego również jako vanilla Emacs. Te podstawowe umiejętności będą pomocne podczas wprowadzania bardziej zaawansowanych funkcji w dalszej części książki.

Proces instalacji programu Emacs zależy od systemu operacyjnego. Witryna internetowa GNU Emacs (emacs.org) zawiera instrukcje dotyczące instalacji programu Emacs w najpopularniejszych systemach operacyjnych. Należy pamiętać, że potrzebna będzie najnowsza wersja Emacs, która w momencie pisania tego tekstu to wersja 30. Instalacja Emacs i całego wymaganego oprogramowania jest bardziej skomplikowana w systemie Windows niż w systemie Linux lub macOS. Strona internetowa Emacs Writing Studio zawiera wskazówki dotyczące instalacji dodatkowego oprogramowania w tym systemie operacyjnym. Po zainstalowaniu oprogramowania należy otworzyć Emacs i zapoznać się z jego funkcjami.

Pierwszą rzeczą, która się pojawia, jest ekran powitalny z linkami do plików pomocy i innych informacji (rysunek 3.1). Kliknij dowolny link, aby przeczytać samouczek, lub naciśnij przycisk q, aby zamknąć ekran (w języku Emacs „kill”). Naciśnięcie q jest standardową metodą zamykania ekranów tylko do odczytu.

Po zamknięciu ekranu powitalnego pojawia się „bufor Scratch”, który można wykorzystać do tworzenia tymczasowych notatek. W terminologii Emacs bufor to obszar w pamięci komputera, w którym przechowywane są treści, które mogą być powiązane z plikiem. Bufor jest dynamiczną wersją

, podczas gdy plik pozostaje niezmieniony do momentu zapisania bufora w powiązonym z nim pliku. Emacs nie zapisuje zawartości bufora Scratch Buffer po zamknięciu programu, więc nie zaczynaj jeszcze pisać swojej pracy doktorskiej.



Rysunek 3.1: Zmaes 29.1 splash seveee.

ç.1 Skrócona instrukcja obsługi programu Emacs

Nie musisz wiedzieć zbyt wiele o Emacsie, aby zacząć z niego korzystać. Działa on bardziej intuicyjnie, niż niektórzy sugerują. Zacznijmy od utworzenia dokumentu Org i wyeksportowania go do strony internetowej, aby zapoznać się z Emacssem.

W pasku menu wybierz *Plik* > *Odwiedź nowy plik*, wybierz katalog i wpisz nazwę pliku z rozszerzeniem *.org*, na przykład: *test.org*. Odwiedzanie pliku to termin używany w programie Emacs, który odnosi się do wczytania pliku do bufora. Plik ten będzie w trybie Org, specjalnym typie pliku tekstowego. Teraz zacznij pisać tak, jak zwykle w każdym innym programie tekstowym.

Możesz również użyć składni Org, aby dodać strukturę i metadane do dokumentu. Na przykład dodaj `#+title: Hello World` jako pierwszą linię, aby zdefiniować tytuł. Aby wstawić nagłówek, rozpocznij linię od pojedynczej gwiazdki, np. `* Rozdział. Użycie dwóch lub więcej gwiazdek powoduje wyświetlenie podnagłówków.`

```
#+title: Hello World
```

```
* Rozdział pierwszy
```

```
Dolor sit amet, consectetur adipiscing elit nulla varius.
```

```
** Sekcja
```

```
Nullam ut consequat lacus. Praesent porttitor urna eget semper.
```

Zauważysz, że te dwa wiersze mają inny kolor niż zwykły tekst. Kolory te nie są rzeczywistymi kolorami tekstu w końcowym produkcie, ale są semantycznymi wskazówkami wskazującymi strukturę dokumentu. Kolory te ułatwiają szybkie przeglądanie dokumentu. Teraz dodaj zwykły tekst poniżej nagłówka. Na koniec zapisz wyniki za pomocą *opcji Plik*

> *Zapisz.* W ten sposób utworzyłeś swój pierwszy dokument tekstowy w Emacsie.

Jeśli zauważysz, że Emacs nie zawija długich zdań na granicy ekranu, nie martw się. Możesz łatwo zmienić to domyślne zachowanie, przechodząc do paska menu i wybierając *Opcje > Liece Wvappieg ie Tłis Buffev > Wovd Wvap (Tryb wizualny Liece)*. Teraz Emacs zachowuje się bardziej jak oprogramowanie do pisania, do którego jesteś przyzwyczajony. Jednak po otwarciu nowego pliku będziesz musiał ponownie włączyć tę opcję. Kiedy zaczniemy konfigurować Emacs, zostanie on ustawiony tak, aby tryb wizualnej linii był włączony dla wszystkich plików tekstowych.

Następnie eksportujemy ten prosty przykład do strony internetowej. Org jest systemem *Włat You See is Włat You Meae*, więc sposób wyświetlania tekstu na ekranie niekoniecznie odpowiada temu, jak będzie on wyglądał w ostatecznej wersji publikacji. Tryb Org przekształca składnię na pożądaną układ typograficzny podczas eksportowania dokumentu przy użyciu szablonu, który w przypadku strony internetowej jest kaskadowym arkuszem stylów (CSS).

Więc każdy tekst w linii `#+title:` jest tytułem dokumentu (tag `<title>`), a każda linia zaczynająca się od jednej lub więcej gwiazdek staje się nagłówkiem (`<h1>`, `<h2>` itd.).

Moduł eksportujący konwertuje te słowa kluczowe na odpowiedni wynik. W kolejnych rozdziałach przedstawiono dodatkową składnię służącą do dodawania obrazów, tabel, formuł i innych elementów.

Ostatnim krokiem jest utworzenie strony internetowej. W menu *Ovg* wybierz *Zxpovt/Publist* i naciśnij klawisze *h* oraz *o*. Org przetłumaczy dokument na plik HTML i otworzy domyślną przeglądarkę, aby wyświetlić wyniki.

Presto, to jest podstawowy przebieg pracy przy pisaniu i publikowaniu dokumentu w trybie Emacs Org. Oczywiście jest tego znacznie więcej, a Emacs posiada niezliczone wtyczki i opcje dostosowywania, które pozwalają stworzyć dokument dokładnie taki, jaki chcesz.

ć.2 Praca z klawiaturą

Emacs jest w większości aplikacją obsługiwana za pomocą klawiatury. Można używać myszy i paska menu do wykonywania sporadycznych zadań, jak pokazano w poprzedniej sekcji, ale nie ma potrzeby korzystania z tego urządzenia peryferyjnego. Chociaż nie ma formalnego porozumienia co do tego, czy bardziej wydajne jest używanie klawiatury, czy myszy (Omanson, Miller, Young i Schwantes, 2010; Tognazzini, 1yy2), większość użytkowników Emacsa woli trzymać ręce na klawiszach.

Może się wydawać, że kliknięcie ikony na pasku menu wymaga mniejszego wysiłku umysłowego niż zapamiętanie sekwencji naciśnień klawiszy. Jednak praktycznym problemem związanym z paskami ikon jest to, że nie ma wystarczająco dużo miejsca, aby pomieścić wszystkie dostępne funkcje. Skróty klawiaturowe są łatwe do zapamiętania, ponieważ szybko stają się częścią pamięci mięśniowej.

Korzystanie ze skrótów klawiaturowych zamiast myszy pozwala uniknąć konieczności ciągłego przenoszenia rąk między dwoma urządzeniami wejściowymi. Na przykład, gdy popełnisz błąd ortograficzny w standardowym edytorze tekstu, przenosisz rękę z klawiatury na mysz, przesuwasz kursor do błędnego słowa, klikasz myszą i wybierasz żądaną pisownię. W Emacsie wystarczy jedno naciśnięcie klawisza, aby zmienić literówkę na najbardziej prawdopodobną poprawną wersję i kontynuować pisanie.

Najważniejszą rzeczą, o której należy pamiętać w debacie na temat klawiatury kontra mysz, jest to, że pisanie polega bardziej na myśleniu niż na wciskaniu tysięcy klawiszy na minutę, więc używanie myszy nie jest grzechem. W Emacsie można używać myszy do niektórych zadań, takich jak zaznaczanie tekstu.

lub przesuwając kursor. Główną zaletą systemu menu jest to, że pomaga on odkrywać funkcje programu Emacs. Nie potrzebujesz myszki, aby uzyskać dostęp do menu. Naciśnij klawisz F10 i użyj klawiszy strzałek, aby poruszać się po menu rozwijanym i odkrywać funkcje programu Emacs.

Ponieważ program Emacs został opracowany przed standaryzacją interfejsów komputerowych, sposób jego interakcji z klawiaturą nieco różni się od tego, do czego jesteś przyzwyczajony. Zacznijmy od podstaw. Standardowa klawiatura komputerowa ma pięć rodzajów klawiszy:

1. Alfnumeryczne: litery, cyfry i znaki interpunkcyjne.
2. Edytowanie: takie jak klawisze strzałek i backspace
3. funkcyjne i multimedialne
4. Esc
5. Klawisze modyfikujące: Shift, Control, Alt, Windows/Command

Naciśnięcie klawisza alfanumerycznego powoduje dodanie znaku do pamięci komputera i wyświetlenie go na ekranie. Jest to skomplikowany sposób na powiedzenie, że dodają one znaki do ekranu. Klawisze edycji, takie jak klawisze strzałek, przewijania w górę i w dół, usuwania i cofania, działają zgodnie z tym, co wskazują ich etykiety. Klawisze funkcyjne i multimedialne wykonują bardziej złożone zadania. Na przykład naciśnięcie klawisza F3 w programie Emacs powoduje zapisanie makra. Klawisze multimedialne są zazwyczaj definiowane przez system operacyjny i aktywują zadania, takie jak zwiększenie jasności ekranu lub odtwarzanie muzyki. Klawisz Escape jest najpotężniejszym elementem klawiatury. Podobnie jak rubinowe pantofelki Dorotki w *Czarnoksiężniku z krainy Oz*, trzykrotne naciśnięcie tego klawisza pozwala wyjść z kłopotów, gdy utkniesz.

Są to jedyne klawisze potrzebne do pisania tekstu, ale chcemy zrobić coś więcej niż tylko wstawiać i edytować tekst. Klawiatury komputerowe posiadają również klawisze modyfikujące, które są specjalnymi klawiszami, które po naciśnięciu tymczasowo modyfikują standardowe działanie innego klawisza.

Klawisze modyfikujące w nowoczesnych klawiaturach komputerów PC i Apple to Shift, Control, Alt / Op-tion oraz Command / Windows. Komputery Chromebook mają te same klawisze modyfikujące.

ale nie ma odpowiednika klawisza Windows / Command. Niektóre mniejsze klawiatury mają również dodatkowe klawisze modyfikujące, takie jak Fn, które rozszerzają dostępne opcje. Klawisze modyfikujące nie mają żadnego efektu, gdy są naciskane samodzielnie. Jak sama nazwa wskazuje, klawisze te modyfikują działanie innych klawiszy, gdy są naciskane jednocześnie.

Niektóre terminy Emacsa dotyczące tych klawiszy pochodzą z czasów, kiedy obecny standardowy układ klawiatury jeszcze nie istniał. Klawisz, który obecnie nazywamy klawiszem Alt, był kiedyś klawiszem *Meta*. Klawisz Windows na klawiaturach komputerów PC lub klawisz Command w systemach Apple odpowiada dawnemu klawiszowi *Super*. System operacyjny wykorzystuje klawisz Windows lub Command do różnych zadań, więc Emacs nie używa go domyślnie. Starsze klawiatury posiadały również klawisz „Hyper”, który Emacs nadal rozpoznaje, ale nie jest już dostępny w nowoczesnych komputerach. Zaawansowani użytkownicy komputerów mogą przypisać ten klawisz do innego modyfikatora.

Dokumentacja Emacsa stosuje skróty sekwencji klawiszy zgodnie z przyjętą konwencją. Podczas korzystania z paska menu, zgodnie z opisem w poprzedniej sekcji, niektóre pozycje zawierają skrócone skróty klawiaturowe w swoich opisach.

Na przykład C-a oznacza jednocześnie naciśnięcie klawiszy Control i a. Myślnik oznacza, że pierwszy klawisz modyfikuje drugi klawisz. Natomiast spacja między klawiszami oznacza, że należy je nacisnąć kolejno. Klawisz spacji i inne nazwane klawisze są oznaczone nawiasami kątowymi, tak jak tutaj <spc> lub <Tab>. Brak nawiasów kątowych oznacza, że należy wpisać litery „spc” lub „tab”.

Każdy klawisz modyfikujący ma swój skrót, jak pokazano w tabeli 3.1. Klawisze modyfikujące można łączyć, co może prowadzić do niezręcznych kombinacji, takich jak C-M-S-a (Control, Alt i Shift a), wymagających zręcznych palców artysty-iluzjonisty, aby wykonać je płynnie. Klawisz Shift zazwyczaj nie jest wskazywany, ponieważ C-M-A jest tym samym co C-M-S-a. Klawisz Escape może również pełnić funkcję klawisza modyfikującego. Jednokrotne naciśnięcie klawisza Escape jest tym samym, co przytrzymanie klawisza Meta: <esc>-x jest tym samym co M-x.

Najważniejszym skrótem klawiaturowym, o którym należy wiedzieć, jest C-g, który anuluje częściowo wpisane polecenie. W przeciwieństwie do potrójnego klawisza escape, polecenie to może również zakończyć działanie funkcji.

Wszystkie naciśnięcia klawiszy w Emacsie wykonują funkcję, co oznacza, że realizują zadanie. Wspomniany powyżej klawisz c-g wykonuje polecenie keyboard-quit. *Funkcja* to funkcja, którą można wywołać interaktywnie, zazwyczaj za pomocą skrótów klawiszowych, wyborów z menu lub za pomocą m-x. *Funkcja* wykonuje określone zadanie i jest wywoływana z innych funkcji lub poleceń, ale nie bezpośrednio przez użytkownika. Wszystkie polecenia są funkcjami, ale nie wszystkie funkcje są poleceniami, a w niniejszej książce terminy te są używane zamiennie.

Tabela 3.1: Klawisze modyfikujące Zmaes.

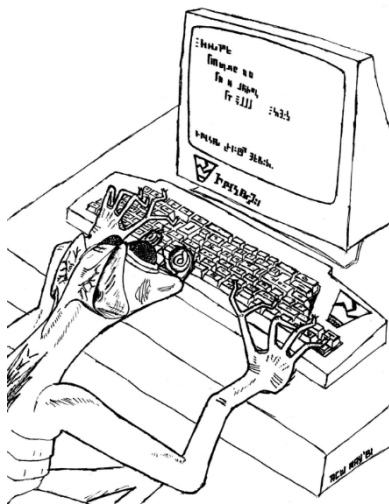
Modyfikator	Przykład	funkcja
Shift	S-8	* znak na klawiaturze amerykańskiej
Control	C-e	Koniec linii
Alt / Option	M-d	Usuń (wytrzyj) słowo
Windows / Polecenie	s	Używane przez system operacyjny
Hyper	H	Nie przypisane do standardowych klawiszy

Większość książek technicznych wyświetla nazwy funkcji czcionką maszynową, aby odróżnić je od standardowego tekstu. Funkcje Emacs'a są najczęściej zapisywane z myślnikami zamiast spacji między słowami, co hakerzy nazywają „kebab-case”. Nie wszystkie funkcje mają skróty klawiaturowe, ale jeśli skrót jest dostępny, to też jest pokazany czcionką maszynową. Znajomość nazw funkcji i odpowiadających im skrótów klawiaturowych pomaga lepiej zrozumieć, jak działa Emacs. Musisz również znać nazwę funkcji, ponieważ skróty klawiaturowe mogą się zmieniać, ponieważ są w pełni konfigurowalne. Ale to nie wszystko. Emacs używa również klawiszy prefiksowych. Po ich naciśnięciu system czeka na dalsze polecenia. Na przykład C-x C-f oznacza, że najpierw należy nacisnąć klawisze Control i x, a następnie Control i f, co jest domyślną sekwencją służącą do wyszukiwania (otwierania lub tworzenia) pliku. za pomocą polecenia find-file.

Po naciśnięciu klawisza prefiksowego Emacs wyświetla go na dole ekranu (w obszarze echo), oczekując na dalsze wprowadzenie danych. Po naciśnięciu kolejnego klawisza jest to albo kolejny prefiks, albo zakończona sekwencja klawiszy. Długość sekwencji klawiszy jest teoretycznie nieograniczona, ale w praktyce zazwyczaj nie przekracza trzech lub czterech klawiszy. Standardowe klawisze prefiksowe to:

- C-x: używany do wbudowanych poleceń Emacs'a
- C-c: używany przez pakiety Emacs'a
- C-h: funkcje pomocy
- M-x: wykonywanie poleceń

Jeśli chcesz wiedzieć, jakie skróty są dostępne po naciśnięciu klawisza prefiksowego, użyj C-h po prefiksie. Na przykład C-c C-h otwiera nowy bufor z listą wszystkich dostępnych skrótów zaczynających się od C-c. Nazwy poleceń mogą być niejasne. Kliknij nazwę funkcji, aby wyświetlić plik pomocy. W Emacsie pomoc jest zawsze pod ręką.



Rysunek 3.2: Okładka wydania podręcznika Emacs z 1981 roku.

Ze względu na swoje dawne korzenie Emacs nie jest zgodny ze standardem Common User Access (CUA) dla interfejsów użytkownika (Berry, 1988). Standard ten definiuje znane skróty klawiaturowe, takie jak C-c i C-x, służące do kopiowania lub wycinania elementów do schowka. Emacs używa tych klawiszy jako prefiksów lub do innych funkcji. Można skonfigurować Emacs'a tak, aby rozpoznawał te popularne skróty klawiaturowe, ale EWS pozostaje przy domyślnym zachowaniu.

Warto wspomnieć jeszcze o jednym klawiszu przedrostkowym. Niektóre polecenia mają alternatywne stany, co oznacza, że ta sama funkcja może mieć różne wyniki. Alternatywny stan aktywuje się, dodając C-u (argument uniwersalny) przed zwykłą sekwencją klawiszy. Emacs powtarza akcję cztery razy, gdy funkcja nie ma alternatywnego stanu dla argumentu uniwersalnego. Tak więc użycie C-u <w górę> przesuwa kursor o cztery linie w górę. Użycie podwójnego argumentu uniwersalnego powoduje przesunięcie o szesnaście linii i tak dalej. Po wpisaniu C-u C-u C-u # Emacs wstawia 54

(4³) symbole hashtagu. Można również powtarzać naciśnięcia klawiszy, dodając cyfrę po klawiszu Control

lub Alt powtarza następne naciśnięcie klawisza. Na przykład M-80 dodaje osiemdziesiąt myślników do tekstu. Ten szczegółowy opis sposobu, w jaki Emacs wykorzystuje klawiaturę, może Cię oszołomić. Okładka podręcznika Emacsa w wersji 1y81 sugerowała nawet, że Emacs jest najlepiej obsługiwany przez kosmitów o niezmiernie zwinnych palcach (rysunek 3.2). Nie martw się, po przeczytaniu tej książki

tej książki będziesz grać na klawiaturze jak wirtuoz fortepianu.

§.§ Wydawanie poleceń

Klawisze modyfikujące i prefiksowe zapewniają mnóstwo skrótów do wydawania poleceń w Emacsie, ale liczba klawiszy nie jest nieograniczona, więc niektóre polecenia nie mają skrótu. Jeśli polecenie nie ma przypisanego skrótu klawiszowego, można przypisać własny; należy tylko uważać, aby nie spowodować konfliktu z istniejącymi skrótami, jak wyjaśniono w dodatku. W niniejszej książce zawsze podano zarówno domyślny skrót klawiszowy EWS, jak i nazwę polecenia.

Funkcje bez skrótu klawiszowego należy wywoływać po nazwie. Standardowym sposobem wykonywania poleceń jest użycie M-x, a następnie wpisanie nazwy polecenia i naciśnięcie klawisza Return/Enter (<Ret>). Po wpisaniu M-x na dole ekranu (w minibufferze) pojawi się M-x, czekające na dalsze instrukcje. Minibuffer służy do wprowadzania danych i instrukcji. Na przykład, aby zagrać w Tetris, wpisz M-x tetris <Ret>. Nie rozprasza się zbytnio; po prostu naciśnij kilka razy q, aby wyjść z gry i powrócić do pracy. Badania eksperymentalne sugerują, że długotrwała gra w Tetris może wpływać na sny (Stickgold, Malia, Maguire, Roddenberry i O'Connor, 2000).

Wpisywanie pełnej nazwy funkcji za każdym razem jest zbyt pracochłonne dla osób poszukujących maksymalnej wydajności. System uzupełniania minibuffera pomaga znaleźć poszukiwane polecenia. Podczas wpisywania częściowej nazwy funkcji lub pliku można nacisnąć klawisz Tab. Emacs wyświetli propozycje uzupełnienia w minibufferze. Na przykład, aby wdrożyć zawijanie linii w pliku tekstowym, należy wykonać funkcję `visual-line-mode`, wpisując M-x visu <Tab>.

Aby zobaczyć, jak działa to uzupełnianie, użyj klawisza Tab po każdej literze wpisanej do minibuffera. Zauważysz, że Emacs zawęży listę potencjalnych uzupełnień w miarę zbliżania się do pożądanego wyboru, aż pozostanie tylko jedna opcja. Zasada ta działa również w przypadku nazw zmiennych i nazw plików. Klawisz Tab jest Twoją tajną bronią, która pomaga zapamiętać i odkrywać funkcje, zmienne, nazwy plików, nazwy buforów i inne potencjalne opcje. Możesz uzyskać dostęp do menu i pasków narzędzi za pomocą myszy, ale zawierają one tylko ograniczony wybór dostępnych funkcji, ponieważ ekran nie jest wystarczająco duży, aby pomieścić je wszystkie.

W dalszej części tej książki podano jedynie nazwy poleceń, bez dodawania części M-x i <Ret>. Gdy w tekście sugeruje się użycie funkcji lub polecenia o nazwie `example-command`, należy to zrobić za pomocą M-x `example-command` <Ret>. Wskazano również wszystkie dostępne skróty klawiaturowe, w którym to przypadku można użyć skrótu, aby uzyskać dostęp do funkcji.

§.4 Tryby główne i pomocnicze programu

Emacs to wszechstronne narzędzie, które realizuje specjalistyczne zadania poprzez tryby edycji zmieniające podstawowe zachowanie. Tryb edycji może być trybem głównym lub pomocniczym. Tryb główny przypomina otwarcie aplikacji w środowisku Emacs, podobnie jak otwieranie aplikacji w telefonie. Najpopularniejszym trybem głównym jest Org, który zapewnia system pisanie i publikowania. Org jest trybem głównym omawianym w całej tej książce.

Bardziej nietypowym trybem Emacsa jest tryb Artist. To narzędzie umożliwia tworzenie rysunków w postaci zwykłego tekstu za pomocą myszy i klawiatury. Spróbuj, uruchom tryb `artist-mode`.

i zacznij rysować myszką. Więcej informacji na temat korzystania z pakietu Artist można znaleźć w wbudowanym systemie pomocy, wybierając C-h P artist.

Wszystkie główne tryby mają te same podstawowe funkcje Emacs, takie jak kopiowanie i wklejanie (usuwanie i wstawianie) oraz otwieranie plików, ale dodają też specjalne zadania, na przykład eksportowanie do strony internetowej. Główny tryb określa podstawową funkcjonalność otwartego bufora. Bufor to część pamięci, która przechowuje tekst, nad którym pracujesz, lub inną zawartość. Każdy bufor ma jeden główny tryb, a każdy główny tryb ma swoją własną, odrębną funkcjonalność, w tym określone skróty klawiszowe i menu rozwijane.

Tryby pomocnicze zapewniają dodatkowe funkcje, takie jak sprawdzanie pisowni, uzupełnianie tekstu lub wyświetlanie numerów linii. Tryb pomocniczy to program pomocniczy, który rozszerza funkcjonalność trybu głównego. Chociaż każdy bufor ma tylko jeden tryb główny, bufor może mieć kilka trybów pomocniczych.

W wielu przypadkach Emacs rozpoznaje główny tryb na podstawie rozszerzenia pliku. Wszystkie pliki Org mają rozszerzenie .org, więc Emacs automatycznie włącza tryb Org podczas otwierania takiego pliku. Nazwa głównego trybu jest wyświetlana w linii poniżej okna. Tryby pomocnicze muszą być wyraźnie włączone, albo dla konkretnego bufora, globalnie dla wszystkich buforów, albo podpięte do głównego trybu.

Aby zrozumieć te zasady, otwórz plik test.org utworzony w pierwszej części tego rozdziału za pomocą C-x C-f. Zauważysz, że tytuł i nagłówki są zaznaczone innym kolorem, a w pasku menu dostępna jest nowa opcja. Teraz zapisz ten plik pod nową nazwą (test.txt) za pomocą C-x C-w (write-file). Bufor nie jest już plikiem Org, ale zwykłym plikiem tekstowym. Bufor zmienia się, ponieważ tytuł i nagłówki nie są już kolorowe. Dodatkowo opcja Org w pasku menu jest niedostępna, a wiersz poniżej otwartego bufora wyświetla teraz „text” zamiast „Org”. Aby powrócić do Org, należy wydać polecenie org-mode.

Dostępne skróty klawiaturowe (mapy klawiszy) i menu rozwijane zależą od aktywnych w danym momencie trybów głównych i pomocniczych. Niektóre mapy klawiszy mają charakter globalny i dotyczą całego programu Emacs. Inne mapy są specyficzne dla danego trybu. O ile tryb nie nadpisuje ich, niektóre skróty pozostają takie same dla wszystkich trybów. Pakiety mogą modyfikować lub dodawać skróty,

w zależności od wymaganej funkcjonalności. Skrót taki jak C-c C-c jest używany przez różne tryby do różnych czynności, w zależności od kontekstu, w jakim jest używany.

§.5 Otwieranie i zapisywanie plików

Otwieranie plików w Emacs nazywa się „odwiedzaniem pliku” i odbywa się za pomocą funkcji find-file (C-x C-f). W praktyce znalezienie, otwarcie i odwiedzenie pliku mają ten sam efekt. Emacs otwiera plik i wyświetla jego zawartość w buforze, gotową do pisania i edycji. Po wpisaniu nazwy, która jeszcze nie istnieje, Emacs tworzy nowy plik. Po otwarciu katalogu Emacs wyświetla zawartość tego folderu w menedżerze plików Emacs (The Directory Ed-itor lub „Dire”) (patrz rozdział y). Alternatywnie można utworzyć plik za pomocą ikony na pasku narzędzi lub poprzez pasek menu.

Emacs prosi o wybranie pliku lub folderu w minibufferze. Wpisywanie pełnej ścieżki do poszukiwanego pliku byłoby uciążliwe, dlatego Emacs pomaga w automatycznym uzupełnianiu, co wyjaśniono w sekcji 3.3. Należy pamiętać, że ścieżki plików w Emacsie używają ukośników, a nie odwrotnych ukośników, jak ma to miejsce w systemie Windows (C:/Users/Freud/, a nie C:\Users\Freud\).

Po znalezieniu pliku Emacs uruchamia się w folderze aktualnie aktywnego bufora. Możesz usunąć tekst przed kursorem, aby przejść do wyższych poziomów drzewa katalogów. Nie musisz usuwać wszystkich podkatalogów. Aby znaleźć plik w katalogu domowym, zignoruj bieżący tekst w minibufferze i wpisz tyldę, a następnie ukośnik (~/) i <Tab>. Aby rozpocząć wyszukiwanie w folderze głównym dysku, wpisz dwa ukośniki (/). Na komputerze z systemem Windows najlepszą metodą jest wpisanie litery dysku, a następnie dwukropka i ukośnika (c:/). Po dwukrotnym naciśnięciu klawisza Tab wszystkie dostępne pliki i foldery pojawią się w minibufferze.

Utwórz nowy plik (C-x C-f), aby poćwiczyć i zacząć pisać do bufora. Po dodaniu tekstu należy zapisać swoją pracę w pliku. Zawartość pliku pozostaje niezmienną do momentu zapisania bufora. Po zakończeniu edycji, C-x C-s zapisuje bufor do powiązanego z nim pliku. Aby zapisać bufor pod nową nazwą, można użyć C-x C-w (tabela 3.2). Można sprawdzić, czy bufor różni się od powiązanego

plik w linii trybu. Jeśli na początku zawiera dwa gwiazdki, oznacza to, że plik wymaga zapisania. Dwa myślniki oznaczają, że zawartość pliku jest identyczna z zawartością bufora.

Tabela 3.2: Najczęściej używane formaty plików eommoely.

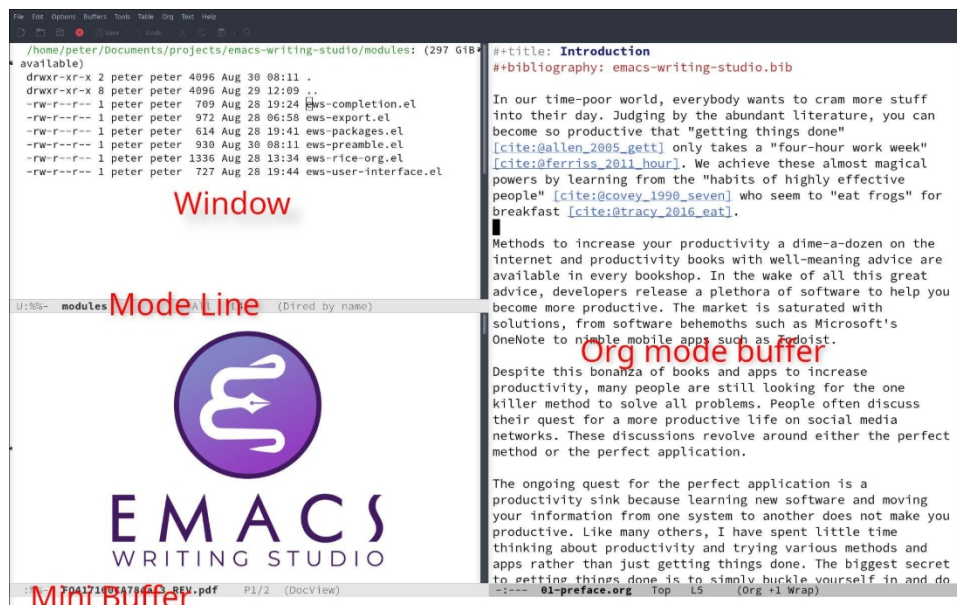
Klawisz	Funkcja	Opis
C-x C-f	find-file	znajdź (otwórz lub utwórz) plik
C-x C-s	zapisz bufor	Zapisz bieżący bufor
C-x C-w	write-file	Zapisz bieżący bufor (Zapisz jako)

3.6 Bufory, ramki i okna

Po uruchomieniu programu Emacs oprogramowanie działa w ramce (rysunek 3.3). Może to wydawać się mylące, ponieważ w większości systemów operacyjnych ramka jest często nazywana oknem. Aby jeszcze bardziej skomplikować sprawę, ramkę programu Emacs można podzielić na okna. Można również otworzyć wiele ramek na pulpicie, na przykład po jednej na każdym monitorze.

Domyślny ekran Emacsa zawiera pasek menu u góry i pasek narzędzi z ikonami bezpośrednio pod nim. Okna zaczynają się poniżej paska narzędzi. Każde okno zawiera bufor, który zawiera tekst, interfejs użytkownika lub dane wyjściowe z funkcji. Linia trybu poniżej każdego okna wyświetla nazwę bufora lub powiązanego z nim pliku, a także inne metadane. Każda ramka ma obszar echa na dole, w którym Emacs wyświetla informacje zwrotne. Echo to termin informatyczny odnoszący się do wyświetlania informacji, takich jak komunikaty o błędach i inne informacje zwrotne. W dolnej części ekranu znajduje się również minibuffer, rozszerzalna sekcja, w której Emacs oczekuje na wprowadzenie danych przez użytkownika, na przykład podczas wybierania bufora lub pliku.

Podobnie jak w przypadku każdego oprogramowania do pisania, pracujesz nad wersją znajdującą się w pamięci (buforze), a poprzednia wersja znajduje się na dysku (w pliku). Możesz mieć otwartych jednocześnie wiele buforów, co pozwala na łatwe przełączanie się między nimi. Aktywny bufor to ten, nad którym aktualnie pracujesz. Nazwy specjalnych buforów, takich jak *Messages*, są otoczone gwiazdkami. Tego typu bufory nie są powiązane z żadnym plikiem.



Rysunek 3.3: Okno z trzema okienkami, buforem podzielonym, buforem obrazu i buforem Ovg.

Emacs jest bardzo stabilny, a niektórzy użytkownicy mają setki otwartych buforów, ponieważ rzadko muszą restartować program. Skrót C-x b (przełącz do bufora) wybiera inny bufor jako aktywny. Za pomocą sekwencji klawiszy C-x lewo i C-x prawo (poprzedni bufor i następny bufor) można przechodzić między buforami w chronologicznej kolejności aktywacji.

Domyślnie ramka ma jedno okno. Możesz podzielić bieżące okno poziomo (split-window-below) lub pionowo (split-window-right), naciskając C-x 2 lub C-x 3. Skrót C-x 0 (delete-window) usuwa bieżące okno, ale bufor pozostaje w pamięci, a C-x 1 usuwa wszystkie okna z wyjątkiem tego, w którym znajduje się kursor (delete-other-windows), więc bieżący bufor obejmuje całą ramkę. Aby przechodzić między oknami, użyj skrótu C-x o (other-window). Ta funkcja przełącza między dostępnymi oknami.

Podczas pionowego podziału okna ten sam bufor pojawia się dwukrotnie. Każde okno może mieć własną pozycję kursora, dzięki czemu można łatwo odwoływać się do innych części tekstu bez konieczności przeskakiwania między oknami i utraty koncentracji. Aktywacja trybu śledzenia powoduje przepływ tekstu bufora, dzięki czemu okna zawierające ten bufor stają się kolumnami tego samego dokumentu. Gdy kursor przesunie się poniżej dolnej krawędzi lewego okna, pojawia się ponownie w prawym oknie, więc wszystkie okna mają jeden kursor. Aby wyłączyć tryb śledzenia, należy ponownie uruchomić tę samą funkcję.

Tabela 3.3: Funkcje okien buforowych.

Klawisz	Funkcja	Opis
C-x b	przełącz do bufora	Wybierz inny bufor
C-x <lewa strzałka>	poprzedni bufor	Przejdź do poprzedniego aktywnego bufora
C-x <prawa strzałka>	next-buffer	Przejdź do następnego aktywnego bufora
C-x 0	delete-window	Usuń bieżące okno
C-x 1	delete-other-windows	Usuń wszystkie pozostałe okna
C-x 2	split-window-below	Podziel bieżące okno w poziomie
C-x 3	split-window-right	Podziel bieżące okno w pionie
C-x o	other-window	Przejdź do następnego okna
	follow-mode	Pokaż bufor w wielu oknach

§.7 Pomoc Finding

Emacs posiada rozbudowany wbudowany system pomocy z różnymi sposobami dostępu do informacji, dostępny za pomocą klawisza C-h. Pełna instrukcja obsługi Emacs'a jest dostępna za pomocą C-h r (info-emacs-manual). Instrukcja ta otwiera się w trybie Info, który jest specjalnym trybem przeznaczonym dla instrukcji obsługi. Pełna instrukcja obsługi Emacs'a nie jest lekturą na dobranoc, ale raczej zbiorem wiedzy, z którego można czerpać w razie potrzeby. Klawisz g pozwala przeskoczyć do rozdziału lub sekcji tekstu (Info-goto-node), korzystając z omówionego wcześniej uzupełniania minibuffera. Na przykład C-h r g help <Ret> przenosi do części podręcznika poświęconej systemowi pomocy. Podczas czytania podręcznika w systemie info spacja przewija ekran w górę, dzięki czemu można przeglądać podręcznik strona po stronie (Info-scroll-up). Przycisk Backspace lub S-<spc> powoduje powrót do poprzedniego ekranu (Info-scroll-down). Podręcznik zawiera hiperłącza w spisie treści i w całym tekście. Można

Kliknij na nie myszką lub naciśnij klawisz Enter, gdy kursor znajduje się na linku. Aby przejść do poprzedniego lub następnego rozdziału, możesz użyć funkcji Info-up i Info-down przypisanych do klawiszy u i d. Jeśli szukasz czegoś konkretnego, funkcja Info-search (s) pozwala znaleźć konkretne terminy. Jak zawsze, q powoduje zamknięcie ekranu.

Niektóre pakiety w Emacsie mają własne podręczniki. Listę dostępnych podręczników można wyświetlić za pomocą C-h R (info-display-manual). Dodatkowo można skorzystać z funkcji uzupełniania minibuffera, aby znaleźć podręcznik tutaj. Podręczniki te można czytać w taki sam sposób, jak opisano w poprzednim akapicie.

Nie wszystkie pakiety Emacsy mają obszerne instrukcje obsługi. Inną metodą wyszukiwania informacji o pakiecie jest funkcja describe-package (C-h P), która wyodrębnia informacje z kodu źródłowego opisującego pakiet.

System pomocy posiada również inne polecenia umożliwiające wyszukiwanie bardziej szczegółowych opisów. Jeśli chcesz dowiedzieć się, które polecenie jest powiązane z danym skrótem klawiszowym, użyj C-h k i wprowadź sekwencję klawiszy. Emacs wyświetla komunikat na dole ekranu, gdy wprowadzisz sekwencję klawiszy, która nie ma przypisanej funkcji, np. „C-c k is undefined”. Aby dowiedzieć się więcej o zmiennej, użyj C-h v (describe-variable) i wpisz jej nazwę. Aby dowiedzieć się więcej o poleceniu, użyj C-h x (describe-command). Pojawi się okno z opisem odpowiedniej zmiennej lub polecenia, które można zamknąć za pomocą q.

W pozostałej części książki znajdują się regularne odniesienia do odpowiednich części systemu pomocy Emacs dla czytelników, którzy chcieliby poznać więcej szczegółów na temat tego systemu. Nie musisz czytać instrukcji od początku do końca, ponieważ ta książka zawiera wszystko, co musisz wiedzieć, aby rozpocząć pracę jako autor Emacsy. Odniesienia do dokumentacji Emacsy są przeznaczone dla osób zainteresowanych poznaniem więcej szczegółów na temat działania tego oprogramowania.

ę.8 Pisanie w Emacsie

Wiesz już wystarczająco dużo, aby zacząć pisać bardziej złożone dokumenty. Otwórz istniejący plik tekstowy lub utwórz nowy i zacznij pisać. Aby osiągnąć pełną wydajność, musisz zrozumieć kilka podstawowych zasad trybu tekstowego, podstawowego trybu głównego dla

pisanie prozy. Dokumentacja Emacsa opisuje tryb tekstowy jako tryb służący do pisania tekstu przeznaczonego dla ludzi, w przeciwieństwie do trybu Prog, który służy do pisania kodu odczytywanego przez komputery. Tryb tekstowy stanowi podstawę wszystkich innych formatów prozy, takich jak Org, Markdown czy fountain. Oznacza to, że wszystkie główne tryby dla autorów wykorzystują tę samą podstawową funkcjonalność do pisania. Po włączeniu Org tryb tekstowy jest również włączany automatycznie.

W tej sekcji podsumowano najczęściej używane polecenia do pisania tekstu. Podręcznik Emacsa zawiera szczegółowy opis wszystkich funkcji związanych z pisanem języków ludzkich, który można przeczytać za pomocą poleceń `C-h r g basic` i `C-h r g text`.

Movieg avoued a buffev

Dokumentacja Emacsa czasami odnosi się do kursora jako do „punktu”. Kursor to znak wyświetlany na ekranie (linia lub ramka), a punkt wskazuje miejsce, w którym pojawi się następny wpisany znak. Punkt ma większe znaczenie podczas pisania funkcji Emacsa, dlatego niniejsza książka skupia się na kursorze, ponieważ to właśnie tam odbywa się pisanie.

Oprócz standardowych metod poruszania się po buforze, Emacs oferuje dodatkowe funkcje ułatwiające nawigację po manuskrypcie. Na przykład naciśnięcie klawiszy `C-p` (poprzednia linia) działa tak samo jak klawisz `<w górę>` (tabela 3.4). Niektórzy użytkownicy preferują te klawisze, ponieważ dzięki nim ręce pozostają w pozycji domyślnej, co pozwala na szybkie pisanie bezwzrokowe.

Tabela 3.4: Poruszanie się po buforze za pomocą klawiszy.

Klawisz	Funkcja	Kierunek
<code>C-b</code> , <code><w lewo></code>	lewy znak	W lewo
<code>C-f</code> , <code><prawa></code>	prawy znak	Prawa
<code>C-p</code> , <code><w górę></code>	poprzednia linia	W górę
<code>C-n</code> , <code><w dół></code>	następna linia	W dół
<code>M-b</code> , <code>C-<w lewo></code>	poprzednie słowo	Poprzednie słowo
<code>M-f</code> , <code>C-<prawa></code>	następne słowo	Następne słowo
<code>C-v</code> , <code><pagedown></code>	scroll-down-command	Przewiń w dół
<code>M-v</code> , <code><pageup></code>	scroll-up-command	Przewiń w górę
<code>C-a</code> , <code><home></code>	przenieś-początek-wiersza	Początek linii
<code>C-e</code> , <code><end></code>	przesuń-koniec-wiersza	Koniec linii
<code>M-<</code> , <code>C-<home></code>	początek bufora	Początek bufora
<code>M-></code> , <code>C-<end></code>	koniec bufora	Koniec bufora

Łatwo jest zgubić się w morzu słów na ekranie. Kilka prostych skrótów klawiszowych może pomóc Ci szybko skupić wzrok. Naciśnięcie klawiszy C-l (ponowne wyśrodkowanie góra-dół) powoduje przesunięcie linii, na której znajduje się kursor, do środka ekranu. Powtórzenie tego skrótu spowoduje przesunięcie kursora do górnej części ekranu. Trzykrotne wykonanie tej czynności spowoduje przesunięcie kursora do dolnej części ekranu.

Czasami trzeba przejść z jednej części dokumentu do drugiej, a potem wrócić do miejsca, z którego się wyszło, ale gubi się wtedy miejsce. Można to zrobić sprawniej, ustawiając znacznik.

Znacznik to zakładka dla pozycji (punktu) w tekście. Ustawienie znacznika jest jak umieszczenie pinezki na mapie. Możesz ustawić znacznik, aby zapamiętać miejsce, do którego chcesz przejść, co jest niezwykle przydatne podczas edycji dużych plików. Znacznik ustawia się za pomocą C-<spc> C-<spc> (set-mark-command), co powoduje zapisanie aktualnej pozycji kursora w pierścieniu znaczników. Pierścień znaczników to sekwencja znaczników dla bieżącego bufora. Teraz można przejść do innej części dokumentu i edytować lub przeczytać potrzebne fragmenty. Powrót do poprzedniego znacznika odbywa się za pomocą kombinacji C-u C-<spc>. Podczas gdy C-<spc> (set-mark) zapisuje bieżącą pozycję w pierścieniu znaczników, dodanie argumentu uniwersalnego powoduje wyodrębnienie tej pozycji i przeniesienie kursora do niej. Wielokrotne naciśnięcie C-u C-<spc> powoduje przejście przez wszystkie znaczniki zapisane w pierścieniu. Po osiągnięciu pierwszej zapisanej wartości powracasz do ostatniej, stąd nazwa „pierścień znaczników”.

Seaveł aed veplae

Chociaż poruszanie się po tekście za pomocą klawiszy strzałek i innych funkcji jest doskonałe, czasami dokładnie wiesz, czego potrzebujesz. Funkcja wyszukiwania i zamiany w Emacsie jest potężna, a ta sekcja tylko powierzchownie opisuje jej możliwości.

Najpopularniejszą metodą wyszukiwania w Emacsie jest wyszukiwanie przyrostowe (isearch-forward). Wyszukiwanie przyrostowe (C-s) rozpoczyna się zaraz po wpisaniu pierwszej litery wyszukiwanego hasła. Podczas wpisywania zapytania Emacs pokazuje, gdzie znalazł tę sekwencję znaków. Wielokrotne naciskanie C-s powoduje przechodzenie między wynikami w buforze. Gdy

Po zidentyfikowaniu żadanego miejsca można zakończyć wyszukiwanie za pomocą klawisza C-g, a kursor powróci do pierwotnej pozycji. Klawisz Enter lub Return zatrzymuje kursor w bieżącej pozycji, umożliwiając edycję tekstu.

Skrót C-s wyszukuje stopniowo od kursora. Można przechodzić między wynikami wyszukiwania, naciskając wielokrotnie C-s. Użycie C-r (isearch-backward) wyszukuje tekst przed kursorem. Emacs zapisuje wyszukiwane hasła w pierścieniu wyszukiwania. Wpisanie C-s C-s powoduje ponowne użycie poprzedniego hasła wyszukiwania. Użycie M-p i M-n pozwala przewijać poprzednie hasła wyszukiwania w pierścieniu.

Aby szukać i zamienić tekst w buforze, użyj M-% (query-replace). Ta funkcja podświetla wszystkie wystąpienia tekstu, który ma zostać zamieniony, i udostępnia szereg opcji dla każdego wystąpienia. Wpisz spację lub y, aby zamienić zaznaczone dopasowanie, oraz delete lub n, aby przejść do następnego. Wykrzyknik zastępuje wszystkie wystąpienia bez dalszego potwierdzenia. Jeśli coś pójdzie nie tak, użyj klawisza u, aby cofnąć ostatnią zmianę, lub klawisza U, aby cofnąć wszystkie zmiany wprowadzone podczas tego wyszukiwania. Klawisz Enter lub q powoduje zakończenie procesu zamiany. Dostępnych jest więcej opcji, które można wyświetlić, naciskając klawisz z znakiem zapytania.

Kopiowanie i wklejanie tekstu

Pisanie jest przyjemne, ale czasami bardziej efektywne jest skopiowanie czegoś, co napisałeś wcześniej, lub skopiowanie tekstu od kogoś innego (oczywiście z podaniem źródła), a może nawet tekstu wygenerowanego za pomocą modelu językowego. System kopiowania i wklejania tekstu działa inaczej niż nowoczesne systemy, ale ma znacznie większą moc.

Aby zaznaczyć fragment tekstu (oznaczyć w języku Emacs), należy najpierw ustawić znacznik za pomocą C-<spc>, a następnie przesunąć kursor, aby podświetlić żądany fragment. Aby zaznaczyć cały akapit, należy użyć klawisza M-h. W kontekście zwykłego tekstu akapit to linia tekstu oddzielona pustymi liniami. Wielokrotne naciśnięcie M-h zaznacza kolejne sekcje. Za pomocą C-x h zaznaczasz cały tekst w buforze, a C-g anuluje wszelkie zaznaczenia. Po zaznaczeniu tekstu możesz go usunąć, skopiować lub przenieść. W niektórych trybach można zaznaczać za pomocą klawiszy Shift i strzałek, ale ta funkcja jest wyłączona w Org, ponieważ klawisze te aktywują inne funkcje.

Nowoczesne odpowiedniki funkcji kopiowania, wycinania i wklejania zostały wymyślone przez Harry'ego Teslera w 1974 roku (Tesler, 2012). Terminologia Emacsa jest bardziej prozaiczna. Kopiowanie tekstu jest równoznaczne z zapisaniem go w „kill-ring”, a wyciągnięcie tekstu powoduje jego pobranie z tego pozornie ponurego miejsca. Podczas gdy schowek w większości systemów zachowuje tylko ostatni wpis, kill ring zapewnia dostęp do wszystkich „zabójczych” wpisów. Innymi słowy, Emacs przechowuje historię wszystkich tekstów skopiowanych i wyciętych z bufora do kill ring. Domyślna długość tej historii wynosi sześćdziesiąt wpisów. Gdy pierścień wycinania jest pełny, najstarszy element znika.

Polecenia kill kopią lub przenoszą tekst do pierścienia kill. Dwa polecenia yank kopią wpis z pierścienia kill do bieżącego bufora. Polecenie yank-pop (M-y) przechodzi cyklicznie przez zawartość pierścienia kill, umożliwiając dostęp do historii. Tabela 3.5 zawiera listę skrótów klawiaturowych służących do kopiowania i przenoszenia tekstu z i do pierścienia kill.

Tabela 3.5: Kopiowanie i wklejanie.

Klawisz	Funkcja	Opis
M-w	kill-ring-save	Skopiuj zaznaczenie do pętli kill
C-w	kill-region	Przenieś zaznaczenie do bufora wycinków
C-y	yank	Wstaw ostatni wpis z bufora wycinków do bufora
M-y	yank-pop	Zastąp wyciągnięty tekst wpisem z pętli kill ring

Corveetieg błędy

Starożytne rzymskie przysłowie mówi nam, że popełnianie błędów jest rzeczą ludzką (*Zwawe łumaeum est*), ale ciągle ich powtarzanie jest diabelskie. Emacs nie przejmuje się tymi wrażliwościami i zapewnia szeroki wachlarz opcji, które pozwalają poprawiać swoje błędy. Najwygodniejszym aspektem pisania elektronicznego jest to, że łatwo można zmienić zdanie lub poprawić błąd bez uciekania się do korektorów lub innych archaicznych metod. Dostępnych jest szereg poleceń edycyjnych, które pozwalają modyfikować tekst i poprawiać literówki (tabela 3.5). Polecenia zaczynające się od kill- przechowują usunięty tekst w pierścieniu kill, dzięki czemu w razie potrzeby można go ponownie wstawić do bufora.

Oprócz usuwania niechcianych znaków i słów, można również zamieniać ich pozycje. Jeśli przypadkowo odwrócisz dwie litery w słowie, możesz zamienić ich kolejność za pomocą

Tabela 3.6: Usuwanie znaków.

Klawisz	funkcja	Czynność
C-d, <delete>	delete-char	Usuń znak po kursorze
<backspace>	delete-backward-char	Usuń znak przed kursorem
C-x C-o	delete-blank-lines	Usuń puste linie poniżej kursora
M-d, C-<delete>	kill-word	Usuń następne słowo
C-k	kill-line	Usuń do końca linii

transpose-char polecenie z kursorem między nimi (C-t). Zamiana słów odbywa się szybko za pomocą polecenia transpose-words (M-t).

Emacs może pomóc Ci, jeśli popełnisz błąd podczas pisania wielkiej litery. Trzy poniższe polecenia zmieniają słowo znajdujące się pod kursorem. Jeśli jesteś w środku słowa, najpierw przesuń się na początek. Dodanie argumentu ujemnego (M--, Alt / Option i klawisz minus) przed tymi poleceniami modyfikuje litery przed kursorem. Ta funkcja jest przydatna, gdy właśnie skończyłeś pisać słowo i zdajesz sobie sprawę, że powinno ono zaczynać się wielką literą. Wpisanie M-- M-c poprawia to za Ciebie bez konieczności przeskakiwania po tekście lub sięgania po mysz.

- M-l: Zamień następujące słowo na małe litery (downcase-word).
- M-u: Zamień następujące słowo na wielkie litery (upcase-word).
- M-c: Zamień kolejne słowo na wielką literę (capitalize-word).

Jeśli zaznaczyłeś fragment tekstu i chcesz zmienić rodzaj czcionki, użyj C-x C-l, żeby zamienić wybrany tekst na małe litery (downcase-region) albo C-x C-u, żeby zamienić go na wielkie litery (upcase-region). Kiedy używasz tych poleceń po raz pierwszy, Emacs wyświetla ostrzeżenie, bo mogą one być mylące dla nowych użytkowników. Po prostu potwierdź i zapisz to na przyszłość.

Polecenie cofania w Emacsie jest przypisane do C-/. Jeśli chcesz cofnąć ten krok, użyj C-? (undo-redo). Emacs zachowuje się inaczej niż inne programy w kwestii cofania i ponawiania edycji, co wymaga pewnego wyjaśnienia. W standardowych edytorach tekstu poprzednie

cofnięte teksty są tracone, jeśli cofniesz coś, wprowadzisz zmiany, ale potem zmienisz zdanie.

Na przykład wpisz „Sokrates”, usuń to za pomocą C-<backspace>, zmień na „Platon”, a następnie cofnij tę zmianę, aby powrócić do „Sokrates” i dodaj trochę więcej tekstu. W standardowych edytorach tekstu nie można powrócić do stanu, w którym tekst zawierał imię „Platon” (stan B na rysunku 3.4). W Emacsie dostępne są wszystkie poprzednie stany. Możesz powrócić do dowolnego poprzedniego stanu za pomocą kolejnych poleceń cofania w Emacsie. Kolejne polecenia cofania i ponawiania powtarzają sekwencję pokazaną na rysunku 3.4, nie tracąc niczego, co wpisałeś. Takie zachowanie może być mylące, ale po pewnym czasie użytkownika pokochasz je, ponieważ nigdy nie tracisz żadnych zmian.



Rysunek 3.4: Stany Emacsu.

Kolejną cechą systemu cofania w Emacsie jest to, że można go zastosować tylko do wybranego obszaru. Załóżmy, że właśnie ukończyłeś pierwszy rozdział i zacząłeś pisać następny. Następnie zdajesz sobie sprawę, że musisz cofnąć niektóre zmiany wprowadzone w rozdziale pierwszym. Jeśli użyjesz funkcji cofania, najpierw cofnie ona wszystkie zmiany wprowadzone w rozdziale drugim, a dopiero potem zmieni rozdział pierwszy. Problem ten można rozwiązać, zaznaczając odpowiedni fragment tekstu w rozdziale pierwszym, a następnie wydając polecenie cofania tylko dla tego fragmentu.

Laeguages ottev tlae Zeglist

Dla większości świata angielski nie jest pierwszym językiem. Kiedy ustawisz klawiaturę w swoim systemie operacyjnym na inny język, Emacs może się pogubić podczas używania klawiszy modyfikujących. Wpisanie M-x na ukraińskim komputerze powoduje pojawienie się cyrylicy Che zamiast x, czego Emacs nie potrafi przetworzyć.

Emacs obsługuje szeroki zakres metod wprowadzania danych, umożliwiających pisanie w wielu różnych językach używanych na całym świecie. Aby zapoznać się z przeglądem różnych języków obsługiwanych przez Emacs, uruchom `view-hello-file` (C-h h). Metoda wprowadzania danych albo konwertuje znaki z klawiatury

znaki bezpośrednio lub konwertuje sekwencję liter na jeden znak. Na przykład, używając jednej z metod do pisania po chińsku, zaczynasz wpisywać, a w minibufferze pojawia się menu, z którego możesz wybrać żądany znak.

Aby wybrać metodę wprowadzania danych dla bieżącego bufora, użyj polecenia `set-input-method` za pomocą `C-x <Ret> C-\`, które pozwala wybrać preferowaną metodę w minibufferze. Początek linii trybu wskazuje aktualnie używaną metodę wprowadzania danych. Możesz tymczasowo wyłączyć wybraną metodę za pomocą `C-\`. Ponowne użycie tego klawisza powoduje powrót do poprzednio wybranej metody wprowadzania danych.

Aby uzyskać bardziej szczegółowe informacje na temat używania klawiatury do pisania w innym języku, użyj `C-h I`, co uruchamia funkcję `describe-input-method`. Aby wyświetlić listę wszystkich dostępnych metod wprowadzania danych, uruchom polecenie `list-input-methods`, a pojawi się nowy bufor z długą listą języków świata. Podręcznik Emacs zawiera szczegółowe informacje na temat różnych metod wprowadzania danych za pomocą `C-h r g input`.

Modyfikowanie wyświetlania

Wygląd bufora na ekranie zależy od trybu głównego, motywu, konfiguracji i aktywnych pakietów. Użytkownik ma pewną interaktywną kontrolę nad rozmiarem tekstu. Aby tymczasowo zwiększyć wysokość tekstu w bieżącym buforze, należy wpisać `C-x C++`. Aby zmniejszyć rozmiar, należy wpisać `C-x C--` (`text-scale-adjust`). Aby przywrócić domyślną (globalną) wysokość czcionki, należy wpisać `C-x C-0`.

Domyślny tryb tekstowy w Emacs nie skraca wierszy jak zwykły edytor tekstu, ale działa do momentu naciśnięcia klawisza `Enter`. W Emacs logiczny wiersz to sekwencja znaków zakończona znakiem powrotu karetki. Wiersz wizualny odnosi się do sposobu wyświetlania w Emacs. Domyślnym ustawieniem jest kontynuowanie logicznych wierszy poza granicami ekranu. Choć może to być przydatne podczas pisania kodu, może powodować zamieszanie podczas pisania prozy.

Emacs posiada kilka funkcji zawijania linii, z których tryb wizualnej linii (`Visual Line Mode`) jest najbardziej przydatny podczas pisania długich tekstów. Aby aktywować ten tryb, należy wykonać polecenie `visual-line-mode`. Wykonywanie tej czynności podczas pracy z buforem może być uciążliwe i właśnie tutaj pojawia się konfiguracja

. Musimy skonfigurować system tak, aby domyślnie włączał zawijanie linii dla wszystkich trybów tekstowych.

ę.y Konfiguracja Emacsa

W poprzednich sekcjach wyjaśniono, jak korzystać z programu Emacs w stanie podstawowym, bez konfiguracji, zwanym potocznie „vanilla Emacs”. Oprogramowanie to bez żadnej konfiguracji pozwala autorom na wykonywanie wszystkich niezbędnych czynności, ale nie jest to rozwiązanie idealne.

Jako system plastyczny, Emacs można konfigurować niemal w nieskończoność, dzięki czemu można dostosować jego działanie do własnych potrzeb. Ponadto użytkownicy Emacsa dzielą się swoimi konfiguracjami i publikują tysiące pakietów zwiększających funkcjonalność programu. W tym rozdziale omówiono zasady konfiguracji Emacsa.

Korzystanie z komercyjnego oprogramowania przypomina wynajmowanie umeblowanego domu, natomiast korzystanie z Emacsa jest bardziej jak posiadanie własnego domu. Jednak Twoje cyfrowe mieszkanie wymaga pewnych zmian, takich jak świeża farba, nowe dywany i meble, abyś mógł poczuć się w nim jak w domu.

Niektóre systemy, takie jak Doom Emacs, Spacemacs i SciMax, zapewniają przydatne zestawy startowe. Chociaż konfiguracje te są pomocne, czasami oferują wszystko oprócz przysłowiowego zlewu kuchennego. Z drugiej strony, konfigurujesz swój system od podstaw, co może stać się hamulcem produktywności, ponieważ musisz przedzierać się przez niezliczone opcje.

Konfiguracja EWS to minimalny zestaw startowy dla autorów. EWS zapewnia elementy składowe, które można modyfikować zgodnie z własnymi preferencjami. Zanim jednak zainstalujesz konfigurację EWS, zapoznaj się z zasadami konfiguracji Emacsa.

Plik Tłe ieitialisatioe

Po uruchomieniu programu Emacs ładowany jest plik inicjalizacyjny, znany również jako plik init. Plik ten zawiera kod Lisp, który ładuje dodatkowe pakiety i konfiguracje podczas uruchamiania programu Emacs. Program Emacs można uruchomić bez pliku init, ale z pewnością będziesz chciał zmodyfikować ustawienia domyślne.

Przy pierwszym uruchomieniu Emacs utworzy folder konfiguracyjny, w którym znajduje się plik inicjujący. Folder ten zawiera również pakiety potrzebne do personalizacji systemu. Emacs szuka pliku o nazwie `.emacs`, `.emacs.el` lub `init.el`. Kropka przed nazwą pliku oznacza, że jest on ukryty, co zapobiega bałaganowi w katalogach.

Zmaes packages

System bazowy Emacs zapewnia szeroki zakres funkcji, ale można rozszerzyć jego możliwości za pomocą jednego z tysięcy pakietów. Wiele osób tworzy i udostępnia oprogramowanie w języku Emacs Lisp, aby ulepszyć lub rozszerzyć możliwości systemu. Twórcy tych pakietów dystrybuują je głównie za pośrednictwem publicznych repozytoriów pakietów, czyli stron internetowych, które umożliwiają łatwe pobieranie i instalowanie pakietów. Dwa najważniejsze z nich to:

- ELPA: GNU Emacs Lisp Package Archive — domyślne archiwum (elpa.gnu.org).
- MELPA: Milkypostman's Emacs Lisp Package Archive (melpa.org).

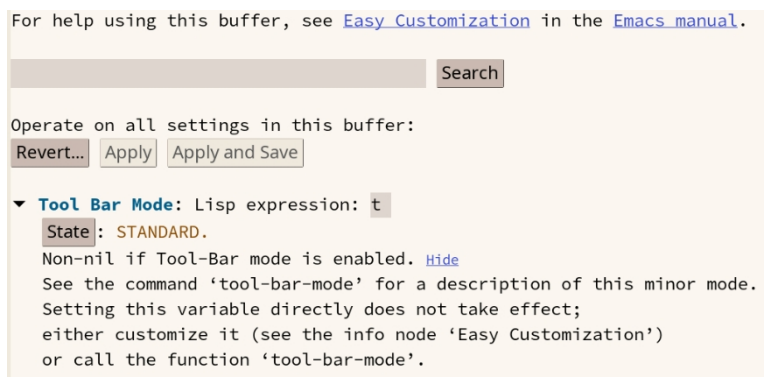
Podstawowa różnica między tymi dwoma repozytoriami polega na tym, kto posiada prawa autorskie. Fundacja Wolnego Oprogramowania posiada prawa autorskie do wszystkich pakietów w ELPA. W przypadku pakietów MELPA prawa autorskie pozostają własnością autora. Dla użytkownika końcowego wynik jest taki sam, ponieważ wszystkie pakiety są licencjonowane jako wolne oprogramowanie. Listę pakietów można przeglądać za pomocą polecenia `list-packages`.

Pakiety są stale aktualizowane przez ich twórców. Aby mieć pewność, że korzystasz z najnowszej wersji, użyj funkcji `package-upgrade-all`. Ta konwencja nazewnictwa może wydawać się sprzeczna z intuicją, ponieważ użycie `upgrade-all-packages` jest bardziej zrozumiałe z językowego punktu widzenia. Jednak zgodnie z konwencją nazewnictwa funkcji Emacs Lisp pierwsze słowo oznacza nazwę pakietu, w tym przypadku `package`. Ta konwencja nazewnictwa ułatwia grupowanie funkcji według pakietów.

Dostosowywanie Zmaes

Oprócz tworzenia własnej konfiguracji w Emacs Lisp lub korzystania z zestawu startowego, Emacs posiada menu dostosowywania, które pozwala skonfigurować system bez konieczności pisania kodu. Załóżmy, że chcesz usunąć pasek narzędzi z widoku, ponieważ do wydawania poleceń używasz wyłącznie klawiatury.

Wpisz M-x customise-variable <Ret> tool-bar-mode, a pojawi się nowe okno z opcjami dostosowywania tej zmiennej (rysunek 3.5). Możesz używać myszki do poruszania się po ekranie konfiguracyjnym. Za pomocą klawiszy <Tab> i S-<Tab> można przesuwać kursor między elementami ekranu.



Rysunek 3./: Ekran dostosowywania trybu paska narzędzi.

W tym przypadku zmienna jest zmienną logiczną, co oznacza, że może przyjmować wartość true (t) lub false (nil). Należy pamiętać, że Lisp nie używa wartości false lub f, ale nil. W Lispie pusta zmienna jest równoznaczna z wartością false, a każda zawartość (lub t) jest interpretowana jako „prawda”. Dokumentacja Emacsa często stosuje mylące podwójne zaprzeczenie. Ustawienie zmiennej na „non-nil” jest równoznaczne z ustawieniem jej na wartość true. Inne zmienne mogą wymagać różnych typów danych wejściowych, takich jak lista rozwijana, pola wyboru lub dowolny tekst.

Przycisk [Zastosuj] powoduje natychmiastowe wprowadzenie tej zmiany, ale zostanie ona zresetowana po ponownym uruchomieniu programu Emacs. Kliknięcie przycisku [Zastosuj i zapisz] powoduje zastosowanie nowego ustawienia i

zapisuje je w pliku `init.el`, dzięki czemu zostaną one aktywowane przy następnym uruchomieniu programu Emacs. Klawisz „q” zamyka ekran dostosowywania.

Użyj funkcji `~customise-group~`, aby wyświetlić przegląd wszystkich parametrów udostępnianych przez pakiet, a następnie wybierz ten, który chcesz zbadać. Przesuwaj kursor między dostępnymi zmiennymi za pomocą klawisza Tab i użyj klawisza Return/Enter, aby dostosować interesującą Cię zmienną.

W niniejszej książce przedstawiono sugestie dotyczące dostosowywania zmiennych. Zamiast pisać w języku Emacs Lisp, można skorzystać z systemu dostosowywania, który pozwala skonfigurować Emacs bez konieczności pisania kodu. Gdy w niniejszej książce pojawia się sugestia dotycząca dostosowania zmiennej, należy wykonać polecenie `customise-variable` (C-c w v), wpisać nazwę zmiennej, wypełnić ekran konfiguracyjny i kliknąć „Apply and Save” (Zastosuj i zapisz).

§.10 Wyjście z Emacsa

Praca z Emacsem jest tak przyjemna, że możesz nie chcieć go nigdy wyłączać. Jednak wszystko, co dobre, kiedyś się kończy, więc czasami może zajść potrzeba wyłączenia (lub zamknięcia) Emacsa.

Skrót klawiszowy C-x C-c (`save-buffers-kill-terminal`) zamyka sesję Emacsa, ale najpierw sprawdza, czy nie ma niezapisanych buforów. Istnieje kilka opcji, które pozwalają uniknąć utraty niezapisanych buforów.

Ta funkcja wyświetla wszystkie niezapisane pliki w obszarze echo i udostępnia opcje dotyczące postępowania z każdym z nich lub wszystkimi naraz. Najbezpieczniejszą opcją jest naciśnięcie klawisza ! i zapisanie wszystkich buforów, w których wprowadzono zmiany, bez zadawania dalszych pytań. Aby wyjść z tej funkcji bez zamykania programu Emacs i utraty tekstu, należy użyć sprawdzonego skrótu C-g. Naciśnięcie klawisza C-h powoduje wyświetlenie komunikatu pomocy opisującego wszystkie dostępne opcje.

§.11 Kolejne kroki

Jeśli chcesz dowiedzieć się więcej o podstawowych zasadach działania Emacsa, dobrym punktem wyjścia jest wbudowany samouczek, który można znaleźć za pomocą polecenia C-h t. Jeśli chcesz zobaczyć, jak inni używają

Emacsa, w Internecie znajdziesz wiele przydatnych filmów instruktażowych.

Jednak najlepszym sposobem nauki obsługi programu Emacs nie jest czytanie o nim ani oglądanie filmów w Internecie, ale po prostu korzystanie z niego. Dobrym miejscem, gdzie można znaleźć interesujące pliki tekstowe do ćwiczeń, jest projekt Gutenberg (gutenberg.org).

Książka ta została napisana i opublikowana przy użyciu opisanej w niej konfiguracji, więc została dokładnie przetestowana w rzeczywistych warunkach. Repozytorium GitHub dla EWS zawiera również folder dokumentów, w którym znajdują się pliki źródłowe Org dla tej książki. Pliki te można pobrać jako przykład książki zbadanej, napisanej i opublikowanej przy użyciu Emacsa.

<https://github.com/pprevos/emacs-writing-studio>

Teraz rozumiesz już podstawowe zasady pisania w Emacsie. W następnym rozdziale wyjaśniono, jak zainstalować i używać konfigurację EWS, która instaluje szereg specjalistycznych pakietów dla autorów.

Rozdział 4

Korzystanie z Emacs Writing Studio

W poprzednim rozdziale opisano, jak korzystać z programu Emacs i konfigurować go, aby działał zgodnie z własnymi preferencjami. Konfiguracja *Zmaes Wwiteg Studio* (EWS) przekształca program Emacs w narzędzie dostosowane do potrzeb autorów. W tym rozdziale wyjaśniono, jak korzystać z dodatkowych funkcji oferowanych przez konfigurację EWS, oraz przedstawiono proces pracy od momentu powstania pomysłu do publikacji.

EWS wykorzystuje minimalistyczny interfejs pozbawiony typowych elementów graficznych oprogramowania. Ten surowy wygląd minimalizuje czynniki rozpraszające uwagę na ekranie, dzięki czemu można skupić się na tym, co najważniejsze — wpisywaniu słów do bufora. Drugą istotną zmianą w stosunku do standardowego Emacs jest system uzupełniania. EWS wykorzystuje pakiety *Vertico/Orderless* i *Marginalia*, aby zapewnić ulepszone uzupełnianie w minibufferze, ułatwiając wyszukiwanie funkcji, plików i innych niezbędnych elementów.

Konfiguracja EWS to zestaw opcji, które mogą nie odpowiadać wszystkim. Użytkownicy Emacs'a czasem dyskutują o tym, jaka konfiguracja domyślna jest sensowna. Niezależnie od tego, jak ciekawe mogą być te dyskusje, taka domyślna konfiguracja nie istnieje. To, co dla jednej osoby jest sensowną domyślną konfiguracją, dla innej może być koszmarem komputerowym. Możesz dowolnie modyfikować dowolny aspekt EWS, aby dopasować go do swojego idealnego sposobu pracy. Dodatek zawiera kompletną konfigurację z adnotacjami oraz sugestie dotyczące wprowadzania zmian.

4.1 Instalacja Emacs Writing Studio

Nie musisz uczyć się programowania w Elisp, aby korzystać z Emacs. Możesz zacząć od konfiguracji EWS, aby rozpocząć pracę. Najpierw jednak musisz zainstalować EWS i dodatkowe oprogramowanie. Aby zainstalować konfigurację EWS, pobierz pliki `init.el` i `ews.el` z repozytorium GitHub i zapisz je w folderze konfiguracyjnym:

<https://github.com/pprevos/emacs-writing-studio>

Plik `init.el` zawiera konfigurację EWS, a plik `ews.el` zawiera zbiór niestandardowych funkcji.

Lokalizacja folderu konfiguracyjnego zależy od systemu operacyjnego. Wpisz `C-h v user-emacs-directory`, aby sprawdzić jego lokalizację w wyskakującym okienku pomocy. Okienko to można zamknąć, naciskając klawisz `q`.

Skopiuj pliki z repozytorium EWS do tego katalogu. EWS aktywuje się po wykonaniu polecenia `restart-emacs` lub przy następnym uruchomieniu programu.

Emacs integruje się również z innymi pakietami wolnego oprogramowania, aby rozszerzyć swoją funkcjonalność. Niektóre funkcje oferowane przez EWS wymagają zainstalowania tego oprogramowania. Nie musisz wiedzieć, jak korzystać z tych programów, ponieważ Emacs będzie nimi sterował za Ciebie. Każdy rozdział tej książki zawiera opis niezbędnego oprogramowania.

Funkcja `ews-missing-executables` sprawdza, czy w systemie dostępne jest oprogramowanie zewnętrzne. Funkcja ta uruchamia się, gdy Emacs rozpoczyna konfigurację EWS i wyświetla w minibufferze komunikat informujący, czy brakuje któreś z zalecanych narzędzi. Aby przejrzeć wynik, można przejść do bufora `*Messages*` za pomocą klawiszy `C-h e`. Emacs będzie działał normalnie, nawet jeśli brakuje któreś z wymienionych programów, ale niektóre funkcje mogą być niedostępne.

Jeśli jesteś użytkownikiem systemu Linux lub Chromebook, pakiety te będą dostępne za pośrednictwem menedżera pakietów Twojego systemu. Użytkownicy systemu Windows mogą skorzystać z menedżera pakietów Chocolatey (chocolatey.org) lub MSYS2 (msys2.org), aby uzyskać to oprogramowanie. Użytkownicy Apple mogą zainstalować narzędzia Linux za pomocą menedżera pakietów Homebrew (`brew.sh`).

4.2 Minimalistyczny interfejs

Emacs to miejsce odpoczynku i kontemplacji, z dala od kakofonii współczesnego oprogramowania, pełnego przycisków i funkcji, których nie potrzebujesz. Konfiguracja EWS usuwa pasek narzędzi, pasek menu i paski przewijania. Chociaż menu rozwijane są cennym narzędziem do odkrywania funkcji, nie ma potrzeby, aby były one wyświetlane na ekranie. Dostęp do menu można uzyskać za pomocą klawisza F10 (menu-bar-open), a opcje menu można wybierać za pomocą klawiszy strzałek i klawisza Ret, aby wybrać pozycję. Menu można zamknąć za pomocą klawisza C-g (keyboard-quit). Jednak po pewnym czasie korzystania z Emacsa szybko nabędziesz pamięć mięśniową i zaczniesz używać skrótów klawiaturowych. Jeśli wolisz, aby menu graficzne, ikony i paski przewijania były widoczne przez cały czas, dostosuj następujące zmienne:

- tool-bar-mode
- menu-bar-mode
- tryb paska przewijania

EWS używa C-c w jako domyślnego klawisza przedrostkowego dla swojej specyficznej funkcji, gdzie w jest skrótem od „pisanie”, a t od „motyw”. W EWS funkcja dostosowywania zmiennych jest powiązana z C-c w v.

Tłemes

Motyw to zestaw instrukcji opisujących kolory określonych części tekstu. Kolory w edytorze tekstu pełnią inną rolę niż w edytorze tekstu. Kolory w Emacsie mają znaczenie semantyczne, co oznacza, że wskazują funkcję tekstu, a nie jego wygląd po opublikowaniu. Nagłówek może mieć inny kolor niż tekst lub metadane, co ułatwia poruszanie się po dokumencie.

Edytory tekstu wykorzystują dwa rodzaje motywów: jasny i ciemny. Jasne tło, znane z większości nowoczesnych programów do edycji tekstu, może powodować astenopię (zmęczenie oczu) po długotrwałym patrzeniu na ekran. Ciemne schematy kolorów zwiększają ostrość widzenia i zmniejszają

zmęczenie wzroku, zwłaszcza w słabo oświetlonych pomieszczeniach o złożonym tle (Kim, Erickson, Lambert, Bruder i Welch, 201y). Wiele użytkowników edytorów tekstu preferuje ciemne motywy. Jasne motywy nie są same w sobie złe i sprawdzają się w jasno oświetlonych pomieszczeniach.

Konfiguracja EWS instaluje i aktywuje najnowszą wersję motywów Modus firmy Prot. Motywy Modus mają dwie podstawowe wersje: motyw modus-operandi jest podstawowym motywem jasnym, natomiast motyw modus-vivendi jest jego ciemnym odpowiednikiem. Podstawowe motywy Modus maksymalizują kontrast między tłem a pierwszym planem, zgodnie z wytycznymi dotyczącymi dostępności treści internetowych (WCAG).

Motywy Modus są zgodne z normą AAA WCAG, która określa minimalny współczynnik kontrastu między tłem a pierwszym planem na poziomie 7:1. Ten wysoki współczynnik kontrastu zapewnia czytelność dla osób z umiarkowanym niedowidzeniem. Każdy z głównych motywów ma trzy zmodyfikowane wersje: wersje dla osób z ślepotą barw czerwonych i zielonych oraz niebieskich i żółtych (deuteranopia i tritanopia) oraz wersję bardziej kolorową (barwioną).

Wersje przyciemnione mają nieco niższy współczynnik kontrastu i są odpowiednie dla osób z normalnym wzrokiem. Motywy Modus nie określają skrótów klawiaturowych, więc EWS definiuje niektóre z nich. Skrót C-c w **t t** przełącza między jasną a ciemną stroną (wstaw tutaj żart z Gwiezdných Wojen).

Pakiet motywów Modus zawiera obszerny podręcznik, który szczegółowo wyjaśnia, jak dostosować wygląd i działanie kolekcji motywów. Podręcznik ten jest dostępny w trybie informacyjnym za pomocą skrótu C-h R modus. Dodatek zawiera dodatkowe informacje na temat dostosowywania motywu do osobistych ustawień.

Użytkownicy Emacs stworzyli zbiór różnorodnych motywów. Aby wybrać swój ulubiony, możesz przejrzeć galerię motywów Emacs (emacsthemes.com). Jeśli motyw jest dostępny w jednym z repozytoriów pakietów, można go zainstalować za pomocą polecenia `package-install`. Wpisz „theme”, aby filtrować listę i wybrać swój ulubiony motyw. Po zainstalowaniu nowego motywu możesz użyć polecenia C-c w **t s** (`consult-theme`). Polecenie to wyświetla menu wyboru wszystkich zainstalowanych motywów, umożliwiając wybranie nowego motywu dla bieżącej sesji.

Po pierwszym uruchomieniu EWS zostanie użyty domyślny motyw Emacs. Aby ustawić preferowany domyślny motyw jasny lub ciemny, uruchom polecenie `customize-themes` i wybierz preferowaną wersję. Postępuj zgodnie z instrukcjami i kliknij przycisk „Zapisz ustawienia motywu”, aby zapisać wybrany domyślny motyw w pliku `custom.el`.

Ustawienia

Domyślną czcionką w Emacs jest czcionka o stałej szerokości (`mono-spaced`) przeznaczona do pisania kodu. W czcionce o stałej szerokości wszystkie znaki mają tę samą szerokość. Litery i lub w zajmują tyle samo miejsca, podobnie jak w maszynach do pisania. Ten rodzaj czcionki, zwany również czcionką o stałej szerokości, jest idealny do pisania kodu lub tabel, ponieważ ułatwia wyrównywanie tekstu. Czcionka o zmiennej szerokości jest bardziej przyjazna dla oka podczas pisania prozy. Nie wszystkie znaki mają taką samą szerokość w czcionce o zmiennej szerokości, co jest powszechne w naturalnym pisaniu.

- Czcionka o stałej szerokości
- Czcionka o zmiennym rozstawie

Najlepiej byłoby połączyć zalety obu rozwiązań i skonfigurować Emacs tak, aby używał najbardziej odpowiedniej czcionki w każdej sytuacji. Emacs może zdefiniować różne czcionki dla poszczególnych części tekstu, dla poszczególnych buforów lub dla trybu głównego. Konfiguracja EWS wykorzystuje tryb zmiennej szerokości znaków, aby osiągnąć idealną kombinację różnych typów czcionek.

Konfiguracja EWS nie określa żadnych konkretnych czcionek i wykorzystuje domyślne czcionki systemu. Możesz skonfigurować swoje ulubione czcionki, pod warunkiem, że są one dostępne na Twoim komputerze. Musisz dostosować trzy zmienne czcionek:

- `default`: ustawienia domyślne (czcionka o stałym rozstawie).
- `stała szerokość`: Czcionka używana w kodzie komputerowym.
- `zmienna szerokość`: Ustawienia dla prozy.

W żargonie Emacsa „face” to zbiór atrybutów służących do wyświetlania tekstu. Definiuje on czcionkę, kolor pierwszego planu, kolor tła, opcjonalne podkreślenie itp. Dostępnych jest wiele atrybutów face, które można konfigurować. Najważniejsze z nich to nazwa czcionki i wysokość czcionki w jednostkach 1/10 punktu. Punkt to dokładnie 1/72 cala, więc jeden punkt to około 0,35 mm.

Możesz skorzystać z menu dostosowywania, wywołując `customize-face` i wybierając domyślny, stały lub zmienny rozmiar czcionki, a następnie wpisując nazwę czcionki w polu *Font Family*. Kliknij [Apply and Save] dla każdej czcionki. Ta czynność zapisuje ustawienia czcionki w pliku `custom.el`, który Emacs ocenia na początku sekwencji startowej. Należy pamiętać, że wszelkie dostosowania zastępują ustawienia motywu, dlatego najlepiej dostosowywać tylko rodzinę czcionek i rozmiar.

Aby sprawdzić, które czcionki są dostępne, użyj menu `set-font`. Gdy używasz GUI do ustawienia domyślnej czcionki, zmiana jest natychmiastowa, ale przejściowa. Aby zapisać domyślną czcionkę i rozmiar w pliku `custom.el`, użyj menu `bar-options-save`. Ta metoda modyfikuje tylko czcionkę domyślną i powinna to być czcionka o stałej szerokości znaków, co w większości nazw czcionek jest oznaczone przedrostkiem „mono”. Możesz również uzyskać dostęp do funkcji, naciskając klawisz F10 i używając klawiszy strzałek, aby znaleźć odpowiednie funkcje w menu opcji.

Należy pamiętać, że czcionki w Emacs nie są czcionkami używanymi w ostatecznej wersji publikacji. Typografia produktu końcowego jest definiowana przez odpowiednie szablony (rozdział 8).

Domyślne ustawienia marginesów mogą sprawiać, że ekran wygląda na nieco przepełniony. Mamy nadzieję, że nie cierpisz na *toivov vaeui* (lęk przed pustą przestrzenią), ponieważ ta konfiguracja instaluje również pakiet Spacious Padding autorstwa Prot Stavrou. Pakiet ten zwiększa przestrzeń wewnątrz okien i ramek, zapobiegając przepełnieniu ekranu symbolami.

4.4 Odkrywanie funkcjonalności Emacsa

Emacs zapewnia dostęp do tysięcy poleceń i setek skrótów klawiaturowych. Po pewnym czasie korzystania z systemu polecenia te stają się drugą naturą, a sekwencje klawiszy zapadają w pamięć mięśniową.

W sekcji 3.3 omówiono system uzupełniania w minibufferze, który pomaga znaleźć polecenia, pliki i inne informacje. EWS instaluje szereg pakietów, które rozszerzają tę funkcjonalność, aby jeszcze bardziej ułatwić znalezienie potrzebnych informacji.

Mieibuffev eompletioe

Nawet pomimo pojawienia się oprogramowania do zamiany mowy na tekst, klawiatura pozostaje najpopularniejszą metodą przekształcania myśli w tekst. Chociaż komputery mogą pewnego dnia nawet czytać w naszych myślach, warto wspomnieć o zaletach używania palców do wyrażania myśli. Kto chciałby, aby jego „eee” i „yyy” lub nieocenzurowany strumień świadomości znalazły się w tekście? Pisanie polega w równym stopniu na myśleniu i tworzeniu strumienia słów, jak i na maksymalizacji liczby uderzeń klawiszy na minutę.

Systemy uzupełniania są podobne do funkcji przewidywania tekstu w telefonie komórkowym. Po wpisaniu kilku znaków komputer pozwala uzupełnić wybór. Emacs posiada rozszerzalny system uzupełniania, który pomaga uzupełniać długie słowa, wyszukiwać pliki, zapamiętywać nazwy funkcji i wykonywać inne proste zadania. Emacs posiada trzy rodzaje systemów uzupełniania:

1. *Mieibuffev eompletioe* pomaga w wyborze opcji w minibufferze, takich jak nazwy funkcji i pliki.
2. *Keyetovd eompletioe*: systemy pomagające w skrótach klawiaturowych.
3. *Text eompletioe* pomaga uzupełniać słowa wpisywane w buforze.

Minibuffer służy do wyszukiwania plików, oceny funkcji i wprowadzania innych informacji. System uzupełniania minibuffera ma na celu ułatwienie znalezienia potrzebnych elementów poprzez zapewnienie mechanizmu wyszukiwania, który wyświetla listę możliwych opcji. Standardowy system uzupełniania minibuffera Emacs koncentruje się na wprowadzaniu funkcji, nazw plików, nazw buforów i wszelkich innych procesów selekcji w minibufferze.

System uzupełniania minibuffera jest wysoce konfigurowalny, a kilka pakietów rozszerza jego podstawową funkcjonalność. Konfiguracja EWS wykorzystuje zestaw powiązanych pakietów opracowanych przez Daniela Mendera, aby zapewnić płynne działanie.

Pakiet Vertico wykorzystuje wyszukiwanie przyrostowe, co oznacza, że lista kandydatów jest skracana, aby dopasować się do wpisywanych znaków, gdy tylko wpiszesz jeden lub więcej znaków. Na przykład, otwierając plik za pomocą C-x C-f, możesz zacząć wpisywać dowolną część nazwy pliku, aby znaleźć plik, którego szukasz.

Pakiet Savehist zapamiętuje wybrane opcje i zapisuje historię minibuffera podczas zamykania programu Emacs. Pakiet ten zapewnia, że najczęściej wybierane opcje pozostają na górze listy, co zwiększa wygodę użytkownika. Aby jeszcze bardziej udoskonalić zdolność programu Emacs do wyszukiwania kandydatów do uzupełnienia, pakiet Orderless dopasowuje wzorce niezależnie od kolejności, w jakiej zostały wpisane. Na przykład wpisanie emacs writing daje takie same wyniki jak wpisanie emacs.

Emacs jest środowiskiem komputerowym z wbudowaną dokumentacją, co oznacza, że każda funkcja i zmienna zawiera opis tekstowy swojego działania. Pakiet Marginalia wyświetla pierwszy wiersz tych tekstów obok propozycji uzupełnień. Pakiet ten pokazuje również dostępne skróty klawiaturowe dla odpowiednich propozycji uzupełnień (rysunek 4.1). Po wpisaniu M-x wyświetli się lista funkcji wraz z krótkim opisem ich działania oraz informacją, czy istnieje skrót klawiaturowy umożliwiający ich wywołanie.

17/8739 M-x	
emacs-uptime	Return a string giving the uptime of this instance of Emacs.
bibtex-sort-buffer	Sort BibTeX buffer alphabetically by key.
normal-mode	Choose the major mode for this buffer automatically.
org-metadown (M-<down>)	Move subtree down or move table row down.
eval-buffer	Execute the accessible portion of current buffer as Lisp code.
scratch-buffer	Switch to the *scratch* buffer.
common-lisp-mode	Major mode for editing programs in Common Lisp and other similar Lisps.
org-wc-count-subtrees	Count words in each subtree, putting result as the property :org-wc on that he...
org-wc-display	Show subtree word counts in the entire buffer.
list-packages	Display a list of packages.

Rysunek 4.1: Uzupełnianie w Emacsie z Vevtio, Ovdevless i Marginalia.

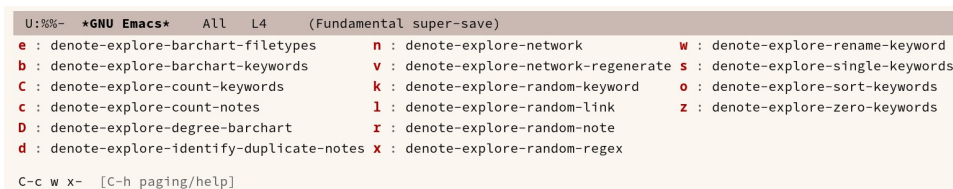
Skróty klawiaturowe

Autouzupełnianie skraca ilość tekstu, który musisz wpisać. Jest to idealne rozwiązanie, aby odkryć funkcje, o których istnieniu być może nie wiedziałeś. Jednak, jak wyjaśniono w poprzednim rozdziale, zazwyczaj nie wpisujemy nazw funkcji, ale używamy skrótów klawiaturowych.

Zapamiętanie potrzebnych skrótów klawiaturowych wymaga pewnego wysiłku. Pakiet Which-Key autorstwa Justina Burketta pomaga zapamiętać, których skrótów klawiaturowych należy używać. Pakiet ten wyświetla skróty klawiaturowe odpowiadające aktualnie wprowadzonym klawiszom prefiksowym w wyskakującym okienku (rysunek 4.2).

Wiele skrótów klawiaturowych składa się z kilku części, np. C-x C-f. Which-Key wyświetla listę wszystkich dostępnych opcji. Na przykład po naciśnięciu C-x menu wyświetli listę wszystkich kolejnych klawiszy i funkcji, z którymi są one powiązane. Wyskakujące okienko z napisem „prefiks” oznacza bardziej szczegółowe informacje. Po naciśnięciu C-c w, prefiksu EWS, wyświetli się lista dostępnych podmenu i funkcji.

Jeśli skróty są zbyt liczne, aby zmieścić się w minibufferze, można przejść do następnej strony za pomocą C-h n, a do poprzedniej strony za pomocą C-h p. Wpisanie C-h w menu Which-Key powoduje wyświetlenie dodatkowych opcji na dole ekranu.



Rysunek 4.2: Okno podręczne Włiet-Key dla C-c w d.

Pomoc

Emacs Writing Studio wykorzystuje pakiet Helpful autorstwa Wilfreda Hughesa. Pakiet ten zapewnia dodatkowy kontekst dla ekranów pomocy, ułatwiając dostęp do informacji. EWS zastępuje standardowe skróty klawiszowe systemu pomocy:

- C-h x: Pomoc dotycząca poleceń (helpful-command)
- C-h k: Pomoc dotycząca skrótu klawiaturowego (helpful-key)
- C-h v: Pomoc dotycząca zmiennych (helpful-variable)

4.4 Ostatnio otwierane pliki i zakładki

Po powrocie do nowej sesji Emacsa można otworzyć ostatnio otwierany plik. Tryb pomocniczy ostatnich plików (`recentf-mode`) wyświetla listę ostatnio otwieranych plików. Aby uzyskać dostęp do tej listy, należy użyć kombinacji klawiszy `C-c w r` i wyszukać żądany plik na liście.

Ten tryb dodatkowo zapisuje listę plików z powiązanymi otwartymi buforami po zamknięciu programu Emacs i przejściu do folderu konfiguracyjnego. Funkcja `recentf-edit-list` otwiera tę listę, dzięki czemu można ją w razie potrzeby modyfikować ręcznie. Tryb ostatnich plików przechowuje ostatnie 50 otwartych plików. Ostatnie pliki są tymczasowe i są stale aktualizowane w miarę otwierania nowych plików.

Aby utworzyć bardziej trwałą listę plików, które chcesz otwierać, użyj zakładek. Plik można zapisać jako zakładkę za pomocą polecenia `C-x r m` (`bookmark-set`). Zakładka zapisuje również położenie kursora, dzięki czemu można utworzyć wiele zakładek dla jednego katalogu lub pliku. Domyślną nazwą zakładki jest nazwa pliku. Przed zapisaniem można również wprowadzić własną nazwę w minibufferze. Aby wyświetlić listę wszystkich dostępnych zakładek w minibufferze i wybrać tę, którą chcesz otworzyć, użyj `C-x r b` (`bookmark-jump`). Jeśli chcesz usunąć zakładkę, która nie jest już potrzebna, użyj funkcji `bookmark-delete`. Ta funkcja nie ma domyślnego skrótu klawiszowego, ale w EWS jest przypisana do `C-x r d`. Zakładki są zapisywane w pliku zakładek w folderze konfiguracyjnym podczas tworzenia lub usuwania zakładki.

4.5 Wprowadzenie do trybu Org

W poprzednim rozdziale wyjaśniono, jak pisać pliki tekstowe. Teraz dodamy nową warstwę funkcjonalności, wprowadzając tryb Org. Ten potężny tryb główny jest domyślnie dostarczany wraz z programem Emacs. Oprogramowanie to zostało pierwotnie opracowane w 2003 roku przez Carstena Dominika, profesora astronomii na Uniwersytecie w Amsterdamie. Od tego czasu niezliczeni inni programiści kontynuowali prace nad rozwojem Org. Wiele osób korzysta z programu Emacs, ponieważ Org stanowi idealne środowisko do pisania.

Tryb Org można wykorzystać do publikowania stron internetowych, artykułów i książek, prowadzenia dziennika, pisania notatek badawczych, zarządzania działaniami i nie tylko. Ponadto jest on intuicyjny w użyciu. W tej sekcji przedstawiono podstawy pisania prozy w trybie Org. W pozostałej części książki wyjaśniono bardziej specjalistyczne funkcje tego rozbudowanego pakietu.

Zacznij od utworzenia pliku z rozszerzeniem .org i zacznij pisać, na przykład C-x C-f test.org. Emacs automatycznie włącza Org dla każdego pliku z rozszerzeniem .org. Org pochodzi z trybu tekstowego, więc wszystko, co wyjaśniono w sekcji 3.8, ma również zastosowanie tutaj.

Każdy dokument Org zaczyna się od nagłówka zawierającego metadane i ustawienia dotyczące bufora. Metadane i ustawienia trybu Org zaczynają się od znaku #+, po którym następuje słowo kluczowe i dwukropek, a następnie metadane. Nagłówek dokumentu może również zawierać inne metadane, takie jak podtytuł, datę i inne informacje. Pakiety Emacs mogą wykorzystywać te informacje podczas publikowania tekstu i do różnych funkcji. Gdyby Szekspir używał Org, przedmowa do *Romea i Julii* wyglądałaby następująco:

```
#+title:    Najwspanialsza tragedia Romea i Julii #+author:
William Shakespeare
#+date:     [1597-05-08 czw.]
```

Doeumeet stvuetuve

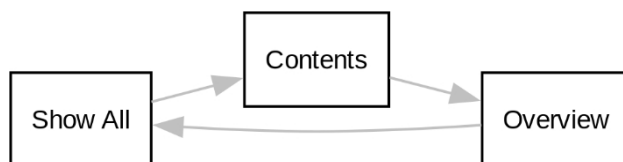
Jedną z nieoficjalnych zasad pisania jest określenie struktury przed napisaniem treści. Książki mają rozdziały, sekcje i akapity; artykuły mają nagłówki; wiersze mają wersy; i tak dalej. Prawie wszystkie formy pisania mają hierarchię. Tryb Org posiada elastyczny zestaw poleceń, które pozwalają szybko określić strukturę projektu pisarskiego. Określenie nagłówków jest tak proste, jak rozpoczęcie linii od gwiazdki, po której następuje spacja. Aby utworzyć głębsze poziomy, dodaj więcej gwiazdek:

```
* Nagłówek 1
** Nagłówek 2
*** Nagłówek 3
```

Po naciśnięciu M-<Ret> następujący wiersz staje się nowym nagłówkiem. Po naciśnięciu C-<Ret> nowy wiersz zostanie dodany po tekście w bieżącej sekcji. Można również przekształcić standardowy akapit w nagłówek za pomocą C-c * (org-toggle-heading). Org ułatwia również przenoszenie i promowanie lub degradowanie istniejących nagłówków oraz powiązanych podnagłówków i tekstu (który w Org jest poddrzewem). Wystarczy użyć klawiszy Alt i strzałek, aby przenieść poddrzewo w obrębie dokumentu. Klawiszy tych można również używać do przenoszenia akapitów.

Poddrzewo nie może zostać przeniesione powyżej poziomu nadrzędnego za pomocą klawiszy Alt i strzałek w górę/w dół. Szybszym sposobem przeniesienia poddrzewa do innej sekcji dokumentu jest użycie polecenia refile, które można wywołać, naciskając C-c C-w (org-refile). Polecenie to wyświetla monit o wprowadzenie nagłówka, aby ponownie zapisać wybrany nagłówek i powiązany z nim tekst, a następnie przenosi go odpowiednio.

Gdy kursor znajduje się na nagłówku, klawisz Tab powoduje zwinięcie tekstu. Wielokrotne naciśnięcie klawisza Tab powoduje wyświetlenie podnagłówków, a następnie ponowne wyświetlenie pełnego tekstu. Aby zwinąć cały dokument, należy dodać klawisz Shift. Naciśnięcie klawiszy S-<Tab> powoduje zwinięcie całego bufora, wyświetlając tylko nagłówki pierwszego poziomu. Ponowne naciśnięcie klawiszy S-<Tab> spowoduje wyświetlenie nagłówków, a powtórzenie tej czynności po raz drugi spowoduje wyświetlenie całego tekstu. Możesz przełączać się między tymi trybami za pomocą klawisza S-<Tab> (rysunek 4.3 i tabela 4.1). Złożone nagłówki można rozpoznać po wielokropkach (...) na końcu linii. Pakiet Org-Modern (sekcja 4.5) zmienia gwiazdki na trójkąty. Gdy trójkąt wskazuje w prawo, nagłówek jest zwinięty, a gdy wskazuje w dół, nagłówek jest otwarty.



Rysunek 4.3: Globalne ustawienia oczu, czyli Org witł S-Tab.

Tryb Org udostępnia również zestaw poleceń ułatwiających nawigację między nagłówkami. Polecenia te umożliwiają przemieszczanie się między nagłówkami tego samego poziomu oraz

Tabela 4.1: Tryb Org stvuetuve editieg.

Skrót	Opis
<Tab> / S-<Tab>	(Rozwijanie/zwijanie nagłówków
M-<w górę> / M-<w dół>	Przesuń nagłówek lub akapit
M-<w lewo> / M-<w prawo>	Podnieś lub obniż nagłówek
M-<Ret>	Wstaw nowy nagłówek
C-c *	Konwertuj akapit na nagłówek
C-c -	Zamień akapit na pozycję listy

przejdź w górę hierarchii. W tabeli 4.1 wymieniono niektóre z dostępnych poleceń związanych ze strukturą dokumentów Org. Rysunek 4.4 przedstawia sposób poruszania się między nagłówkami Org za pomocą klawiszy C-c C-*, gdzie * oznacza b, f, n, p lub u. Tak więc C-c C-n przenosi kursor do następnego nagłówka, niezależnie od jego poziomu w hierarchii, a C-c C-u przenosi do nadrzędnego nagłówka bieżącego nagłówka.



Rysunek 4.4: Przemieszczanie się między nagłówkami w trybie Org za pomocą klawiszy C-c C-*.

Formatowanie tekstu

Pisanie wszystkich słów w tym samym stylu może być nudne, a niektóre fragmenty tekstu wymagają podkreślenia. Aby zmienić sposób wyświetlania tekstu przez Org, należy otoczyć go specjalnymi znakami:

/kursywa/, *pogrubienie*, _podkreślenie_, +przekreślenie+, =dosłownie=

W Vanilla Emacs znaczniki te pozostają widoczne, ale znikają podczas eksportowania dokumentu do formatu publikacji. Konfiguracja EWS ukrywa te znaczniki. Jedynym problemem związanym z ukrywaniem znaczników podkreślenia w ten sposób jest to, że edytowanie tekstu sformatowanego staje się trudne, ponieważ nie jest jasne, czy kursor znajduje się na znaczniku, czy na pierwszym lub ostatnim znaku. Dlatego EWS wykorzystuje pakiet Org-Appear autorstwa Alice Hacker. Narzędzie to wyświetla znaczniki tekstu sformatowanego, gdy kursor znajduje się na słowie, ale w przeciwnym razie je ukrywa, dzięki czemu ekran jest mniej zagracony.

Listy

Pisanie długich akapitów prozy może utrudniać zrozumienie treści, dlatego autorzy literatury faktu używają list, aby zapewnić przejrzystość tekstu. Tworzenie list w Org jest niezwykle proste.

Rozpocznij linię od myślnika i zakończ wpis za pomocą M-<Ret>, aby utworzyć następny wpis. Użycie klawiszy Alt i strzałek w lewo lub w prawo zmienia głębokość elementu. Klawisz Alt w połączeniu z klawiszami strzałek w górę i w dół przesuwa linię w górę lub w dół hierarchii. Prefiks listy można zmienić za pomocą klawiszy Shift i strzałek w lewo/w prawo. Akapit można przekształcić w listę za pomocą C-c - (org-toggle-item). Wielokrotne użycie tego polecenia zmienia typ punktów, podobnie jak klawisze Shift i strzałki.

- Element
 - + następny element
 1. Lista numerowana
 2. I kolejny

Listy numerowane domyślnie zaczynają się od jedynki, ale można dodać plik cookie, aby lista zaczynała się od innej liczby. Na przykład, aby lista zaczynała się od liczby 3, należy dodać [@3], jak pokazano poniżej.

3. [@3] Pierwsza linia
4. Drugi wiersz

Lieks

Tekst, podobnie jak człowiek, nie może istnieć w samotności. Ludzie mają rodziny i przyjaciół, podobnie jak teksty. Być może potrzebujesz linków, aby połączyć powiązane pliki lub odwołać się do tekstu, z którego zaczerpnąłeś pomysły. Relacje między tekstami elektronicznymi są ustalane za pomocą hiperłączy. Tryb Org rozpoznaje szeroki zestaw typów linków, takich jak strony internetowe, identyfikatory obiektów dokumentów (DOI) i pliki wewnętrzne.

Linki w Org pojawiają się między podwójnymi nawiasami kwadratowymi `[[]]` i zaczynają się od identyfikatora, po którym następuje sam link. Na przykład strony internetowe zaczynają się od zwykłego `https:`, pliki od `file:`, a numery DOI, jak można się domyślić, od `doi:`.

Jeśli lokalizacja linku zawiera spacje, należy ją ująć w cudzysłowy, aby stała się aktywna: `[[file:"nazwa pliku"]]`. Linki mogą również zawierać opis przy użyciu następującej składni: `[[type:link][description]]`. Gdy link ma opis, Org ukrywa składnię i formatuje go jak hiperłącze na stronie internetowej. Link do strony internetowej w Org pojawia się jako podkreślony hiperłącze, wyświetlając tylko opis. Polecenie `org-toggle-link-display` włącza i wyłącza to zachowanie. Tak więc składnia linku do Wikipedii wygląda następująco:

```
[[https://wikipedia.org/][Wikipedia]]
```

Aby przejść do linku w Org, kliknij myszką lub naciśnij C-c C-o, gdy kursor znajduje się na tekście linku (`org-open-at-point`). Jeśli użyjesz tego skrótu w dowolnym miejscu tekstu, które nie jest linkiem, Org wyświetli listę linków w bieżącej sekcji.

Nie musisz wpisywać nawiasów kwadratowych ani tworzyć i edytować linków za pomocą C-c C-l (`org-insert-link`). Ta funkcja najpierw prosi o wybranie typu linku, a następnie pełnego adresu linku. Ostatni krok wymaga podania opcjonalnego opisu. Aby usunąć link i zachować tylko opis jako zwykły tekst, użyj tego polecenia i usuń adres linku, zachowując lub modyfikując opis.

Tryb Org domyślnie ukrywa składnię linków. Możesz zmienić to zachowanie za pomocą polecenia `org-toggle-link-display`.

EWS zawiera pakiet Org-Webtools autorstwa Adama Portera, który zapewnia wygodne funkcje do pracy z linkami do stron internetowych. Aby utworzyć link do strony internetowej, skopiuj dowolny adres URL z przeglądarki i użyj `org-web-tools-insert-link-for-url` (C-c w w). Polecenie to tworzy w pełni sformatowany link z adresu URL w kill ring i pobiera opis ze strony internetowej. Aby dowiedzieć się, jakie inne funkcje oferuje ten pakiet, użyj C-h P `org-web-tools`.

Obrazy

Chociaż Emacs jest zwykłym edytorem tekstu, może również wyświetlać obrazy. Org nie osadza obrazów w pliku, ale tworzy link, dzięki czemu tekst i obrazy pozostają oddzielnymi plikami. Linki do obrazów są linkami do innych plików bez opisu. Aby dodać obraz, naciśnij C-c C-l (`org-insert-link`) i wpisz `file:.` Naciśnij Enter i wybierz nazwę pliku obrazu w minibufferze, ale nie wpisuj opisu.

Możesz pominąć część `file:` dodając argument uniwersalny za pomocą skrótu C-u C-c C-l, skąd możesz rozpocząć wybieranie pliku obrazu. Twój bufor będzie teraz zawierał link zaczynający się od `file:`, katalogu i nazwy pliku. Linki do plików mogą być bezwzględne lub względne względem katalogu, z którego pochodzi dokument, do którego linkujesz.

```
[[file:ścieżka/do/obrazu]]
```

Po dodaniu linku można wyświetlić podgląd obrazu za pomocą skrótu C-c C-x C-M-v, który jest powiązany z funkcją `org-redisplay-inline-images`. Aby przełączać podgląd obrazów w całym dokumencie, użyj skrótu C-c C-x C-v (`org-toggle-inline-images`). Konfiguracja EWS umożliwia domyślny podgląd obrazów we wszystkich buforach Org. Podczas dodawania nowego obrazu należy włączyć podgląd za pomocą polecenia `redisplay` (C-c C-x C-M-v).

Obrazy w buforze trybu Org są zawsze wyrównane do lewej lub prawej krawędzi strony, w zależności od kierunku pisania w danym języku. Obrazy mają domyślnie szerokość 300 pikseli w buforze Org. Możesz skonfigurować rozmiar podglądu zgodnie z własnymi preferencjami, używając

dodając linię nad obrazem, na przykład: `#+attr_org: :width 600`. Ta linia zmienia tylko rozmiar obrazu w buforze Org, ale nie w ostatecznym formacie eksportu. Można również dodać podpis i nazwę odniesienia do obrazu. W rozdziale 8 wyjaśniono, jak ustawić rozmiar obrazu, wyrównanie, podpisy i odnośniki w ostatecznej wersji eksportowanej. Poniższy przykład pokazuje, jak obraz wygląda w Org, wraz z podpisem, nazwą odnośnika i atrybutami.

```
#+caption: To jest podpis obrazu. #+name:
          fig-example
#+attr_org: :width 600
[[file:ściezka/do-
obrazu]]
```

Moje projekty pisarskie zawierają oddzielny folder z plikami graficznymi, aby oddzielić je od tekstu. Zmiana nazwy pliku graficznego lub usunięcie go z projektu powoduje powstanie nieaktywnego linku. Emacs posiada również pewne funkcje do zarządzania bibliotekami graficznymi poprzez pakiet Image-Dired, omówiony w rozdziale y.

Tabele

Tabela jest powszechnie stosowanym mechanizmem w publikacjach technicznych, służącym do strukturyzowania informacji zamiast prozy. Tworzenie tabel w Org odbywa się za pomocą intuicyjnej metody dodawania, usuwania i przenoszenia kolumn i wierszy. Aby utworzyć tabelę, należy rozpocząć wiersz od symbolu rury (`|`), wprowadzić treść i kontynuować, aż zdefiniowane zostaną wszystkie kolumny, a następnie zakończyć wiersz ostatnią rurą. Nie trzeba martwić się o wyrównanie tekstu, ponieważ klawisz `Tab` automatycznie dodaje spacje w celu dostosowania rozmiarów kolumn. Dodaje również kolejny wiersz i umieszcza kursor w pierwszej komórce.

Po rozpoczęciu wiersza znakiem `|` i naciśnięciu klawisza `Tab` tworzysz poziomą linię w poprzek tabeli. Zaczynij wypełniać komórki informacjami. Jeśli tabela zostanie źle wyrównana, klawisz `Tab` lub `C-c C-c` ponownie wyrówna odstępy. Możesz również dodać poziomą linię poniżej kursora i przejść do następnego wiersza za pomocą `C-c -`.

```
| Kraj      | Powierzchnia |
|-----+-----|
| Singapur | 735 |
```

Org może również uprościć tworzenie pustej tabeli lub konwersję regionu tekstu do tabeli za pomocą `org-table-create-or-convert-from-region`, które jest powiązane z C-c |. Gdy żaden region nie jest zaznaczony, Org pyta o wymiary tabeli. Wpisanie 3x4 powoduje utworzenie tabeli z trzema kolumnami i czterema wierszami (w tym nagłówkiem) lub dowolnym rozmiarem. Po wybraniu regionu podczas wydawania tego polecenia region zostanie przekształcony w tabelę, a wszystkie przecinki w tekście staną się kolumnami. Ta funkcja jest przydatna podczas kopiowania tabeli z innego dokumentu niebędącego dokumentem Org, takiego jak strona internetowa. Jeśli masz tabelę zapisaną w pliku CSV (Comma-Separated Values), możesz ją zaimportować za pomocą funkcji `org-import-table`.

Przechodź do kolejnych komórek za pomocą klawisza Tab lub strzałek w górę/w dół. Użycie kombinacji S-<Tab> powoduje cofnięcie kursora o jedną komórkę. Aby zmienić układ tabeli, użyj kombinacji klawiszy Alt i strzałek. Tak więc M-<w górę> powoduje przesunięcie wiersza w górę, a M-<w lewo> powoduje przesunięcie kolumny w lewo. Kombinacja klawiszy Alt i Shift wraz ze strzałkami powoduje dodanie lub usunięcie kolumn i wierszy sąsiadujących z kursorem. Nie ma potrzeby ręcznego sortowania tabeli. Jeśli chcesz posortować tabelę alfabetycznie lub numerycznie, użyj funkcji `org-sort` (która sortuje również listy). Funkcja ta wyświetla menu wyboru w minibufferze, aby potwierdzić kolejność sortowania. Na koniec, jeśli uznasz, że wiersze powinny stać się kolumnami i odwrotnie, funkcja `org-table-transpose-table-at-point` wykona to zadanie za Ciebie.

Poruszanie się w obrębie komórki jest również możliwe za pomocą klawiszy M-a i M-e (wstecz-zdanie i dalej-zdanie). Użycie tych poleceń powoduje przesunięcie kursora na początek lub koniec zawartości komórki. Polecenia te można również wykorzystać do przechodzenia między zdaniami w akapicie tekstu poza tabelą.

Domyślna szerokość kolumny to długość najszerzej komórki plus dwa spacje. Kolumny zawierające głównie liczby są automatycznie wyrównane do prawej strony, a pozostałe kolumny do lewej strony. Można zmienić to zachowanie, dodając specjalny wiersz, który wskazuje jego

wyrównanie i szerokość między nawiasami kątowymi ($\langle \rangle$).

Poniższy przykład przedstawia trzy największe hrabstwa pod względem powierzchni w km^2 . Szerokość pierwszej kolumny jest ograniczona do pierwszych 13 znaków i wyśrodkowana. Użyj skrótu C-c $\langle \text{Tab} \rangle$, aby przełączać się między kolumnami skróconymi a pełnej szerokości. Możliwe skróty wyrównania to $\langle \text{L} \rangle$, $\langle \text{C} \rangle$ i $\langle \text{R} \rangle$, które można łączyć z szerokością kolumny, jak pokazano w poniższym przykładzie. Użycie C-c $\langle \text{Tab} \rangle$ bez skrótu szerokości spowoduje zminimalizowanie kolumny do jednego znaku.

#+podpis: Trzy największe kraje pod względem powierzchni.

Kraj	...	Powierzchnia
$\langle \text{c13} \rangle$	\dots	$\langle \text{l} \rangle$
Rosja	\dots	17098242
Kanada	\dots	9984670
Chiny	\dots	9640011

lesevtieg speeial ełavaetevs

Tekst to często coś więcej niż zbiór liter, cyfr i znaków interpunkcyjnych. Istnieją dwie metody wstawiania znaków niealfanumerycznych do tekstu.

Standardową metodą w Emacsie jest wstawianie znaków specjalnych bezpośrednio do tekstu. Funkcja insert-char (C-x 8 $\langle \text{Ret} \rangle$) udostępnia menu z setkami opcji. Jeśli potrzebujesz któregoś z dostępnych rodzajów strzałek, wyszukaj żadaną strzałkę za pomocą klawiszy strzałek i wybierz tę, której potrzebujesz. Wybrane glify muszą oczywiście być dostępne w używanej czcionce. To podejście pozwala również na wstawianie do dokumentu emoji, czyli XXI-wiecznej wersji hieroglifów. Klawisz prefiksowy C-x 8 e wyświetla menu z funkcjami umożliwiającymi wybór i wstawianie emoji.

Tryb Org oferuje również metodę wstawiania znaków specjalnych inspirowaną składnią LaTeX. Na przykład, gdy wpiszesz π , Org wyświetli to jako π , jeśli włączona jest opcja ładnych znaków, która jest domyślna w EWS. Możesz przełączać to zachowanie za pomocą klawiszy C-c C-x \

(org-toggle-pretty-entities). Dostępne są greckie litery (`\alpha` do `\omega`) i wiele innych symboli.

Aby wyświetlić listę wszystkich tych elementów i ich tłumaczenia na LaTeX lub HTML, należy użyć polecenia `org-entities-help`. Aby wstawić znak specjalny, należy użyć elementu Org. Tabela 4.2 przedstawia niektóre znaki specjalne dostępne w Org. Należy zauważyć, że większość elementów Org jest identyczna z ich odpowiednikami w LaTeX. Sekcja 8.5 zawiera więcej szczegółowych informacji na temat używania LaTeX w Org.

Tabela 4.2: Przykłady jednostek trybu Org.

Symbol	Jednostka Org
Δ	<code>\Delta</code>
\aleph	<code>\aleph</code>
\dots	<code>\dots</code>
€	<code>\EUR</code>

Aby zapisać indeksy górne i dolne, należy użyć znaku karetki i podkreślenia, np. `m^{2}` lub `CO_{2}`, które są wyświetlane jako m^2 i CO_2 . Domyślnie Org nie wymaga używania nawiasów klamrowych dla indeksów górnych i dolnych. Może to jednak powodować niejasności, jeśli użytkownik preferuje pisanie tekstu przy użyciu „snake_case”. Konfiguracja EWS ogranicza stosowanie indeksów górnych i dolnych do znaków w nawiasach klamrowych.

Matematical notation

Autorzy tekstów technicznych często korzystają z notacji matematycznej, która w Org jest zapisywana w składni LaTeX. Wzór jest otoczony jednym lub dwoma znakami dolara. Pojedynczy znak dolara oznacza wzór wbudowany, natomiast podwójny znak dolara powoduje wyświetlenie wzoru jako oddzielnego akapitu z większymi symbolami. Aby pokazać, jak wyglądają wzory LaTeX, poniżej przedstawiono wzór Ramajunana na π , zarówno w formie graficznej, jak i w notacji LaTeX.

$$\pi = \frac{1}{9801} \sum_{n=0}^{\infty} \frac{(-1)^n (4n)!}{(n!)^4} \times \frac{26390n + 1103}{3964n}$$

Pełne wyjaśnienie notacji formuł LaTeX wykracza poza zakres niniejszej książki. Możesz przeanalizować ten przykład, aby poznać zasady notacji matematycznej LaTeX.

```


$$\frac{1}{\pi} = \frac{\sqrt{8}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!}{(n!)^4} \times \frac{26390n + 1103}{396^{4n}}$$


```

Gdy włączona jest opcja ładnych znaków, niektóre symbole LaTeX są konwertowane na notację matematyczną, dlatego podczas pisania wyrażeń matematycznych należy wyłączyć tę opcję za pomocą C-c C-x

\.

Tryb Org może wyświetlać podgląd fragmentów LaTeX jako obrazy, jeśli dostępny jest program dvi2png (dołączony do LaTeX). Aby wyświetlić podgląd fragmentu pod kursorem, naciśnij klawisze C-c C-x C-l (org-latex-preview). Proces ten konwertuje formuły LaTeX do pliku SVG przechowywanego w podkatalogu o nazwie ltximg.

Pakiet Org-fragtog autorstwa Benjamina Levy'ego zapewnia wygodną funkcję przełączania między fragmentami tekstu zwykłego LaTeX a podglądem obrazów. Gdy kursor znajduje się wewnątrz wzoru, Emacs wyświetla tekst zwykły; gdy znajduje się poza wzorem, wyświetla wersję graficzną, eliminując potrzebę ręcznego przełączania się między nimi.

Tryb Rieieg Ovg

Ricing to slangowe określenie używane przez programistów, odnoszące się do znacznego dostosowywania wyglądu edytora. Upiększanie może obejmować motywy, czcionki i inne poprawki wizualne w celu stworzenia unikalnego stylu. Vanilla Emacs to brzydkie kaczątko, które można przekształcić w pięknego łabędzia. Plik konfiguracyjny EWS zawiera pewne modyfikacje interfejsu użytkownika.

Główną różnicą między zwykłym edytorem tekstu a edytorem WYSIWYG jest to, że w Emacsie wygląd tekstu (czcionka, kolor itp.) przekazuje znaczenie, a nie tylko wygląd. Motyw Emacs'a określa kolory i czcionki dla dokumentu.

Celem tego formatowania jest ułatwienie poruszania się po dokumencie. Wygląd dokumentu w buforze nie jest taki sam, jak po wyeksportowaniu do produktu końcowego.

Aktywny motyw, konfiguracje i pakiety określają wygląd bufora trybu Org. Emacs definiuje wygląd bufora poprzez tryb `font-lock-mode`. Blokowanie czcionek oznacza (lub „fontyfikuje” w języku Emacsa) fragmenty tekstu przy użyciu reguł logicznych. Ocena trybu `font-lock-mode` przełącza między w pełni skonfigurowaną wersją pliku trybu Org a wersją zwykłego tekstu. Uruchom tę funkcję w istniejącym widoku trybu Org, aby zobaczyć różnicę między zwykłym tekstem a tekstem z fontami. Aby pójść o krok dalej, możesz otworzyć plik w trybie Org i uruchomić tryb tekstowy, aby wyłączyć wszystkie funkcje trybu Org i zobaczyć plik w jego surowej postaci. Aby wrócić do bezpiecznego trybu, uruchom tryb `org`, aby przywrócić plik.

EWS wykorzystuje części pakietu `Org-Modern` autorstwa Daniela Mendlera. Pakiet ten implementuje nowoczesny styl dla buforów Org przy użyciu blokowania czcionek i właściwości tekstu. Jednak niektóre elementy stylizacji nie zostały zaimplementowane, ponieważ początkującym użytkownikom łatwiej jest widzieć pełną składnię. W załączniku wyjaśniono, jak skonfigurować ten pakiet.

4.6 Sprawdzanie pisowni

Pisanie z użyciem sprawdzania pisowni stało się dla autorów nieodzownym zabezpieczeniem. Bez fałistej czerwonej linii moje teksty byłyby pełne literówek. Połączone pakiety `Ispell` i `flyspell` Emacs zapewniają interfejs do oprogramowania sprawdzającego pisownię `Hunspell`, więc należy upewnić się, że jest ono dostępne na komputerze, wraz z co najmniej jednym słownikiem. Konfiguracja EWS włącza tryb `flyspell minor` (sprawdzanie pisowni na bieżąco) dla wszystkich trybów tekstowych.

Istnieją zasadniczo dwa sposoby poprawiania tekstu. Można albo kontynuować pisanie i sprawdzić cały tekst po zakończeniu sesji, albo poprawiać wykryte błędy ortograficzne na bieżąco.

Funkcja `ispell` (`C-c w s s`) przegląda wszystkie podejrzone błędy ortograficzne w bieżącym buforze lub zaznaczonym obszarze. Funkcja ta wyświetla proponowane poprawki w górnej części okna. Możesz wybrać preferowaną poprawkę, wprowadzając odpowiedni numer. Minibuffer udostępnia menu do zarządzania błędem. Zignoruj literówkę za pomocą

spacja; zaakceptuj ją dla tej sesji za pomocą `a`, dodaj ją do swojego osobistego słownika za pomocą `i` i zapoznaj się z innymi opcjami, które wyświetla `C-h`.

`flyspell` posiada również przydatną funkcję umożliwiającą przechodzenie między podejrzanymi błędami ortograficznymi. Funkcja `flyspell-goto-next-error` (`C-;`) umieszcza kursor na następnym słowie wymagającym sprawdzenia. Dodanie argumentu uniwersalnego (`C-u C-;`) powoduje przejście do poprzedniego błędu. Aby poprawić podejrzanę słowo, należy użyć funkcji `ispell-word` (`M-$`) i opcji dostępnych w opisanym wcześniej menu.

Sprawdzanie pojedynczych słów wymaga przesunięcia kursora na dane słowo, a następnie powrotu do pisania. `flyspell` oferuje wygodniejszy sposób poprawiania słów w locie za pomocą `flyspell-auto-correct-previous-word` (`C-;`). Polecenie to zastępuje pierwszy wykryty błąd ortograficzny przed kursorem widocznym na ekranie najbardziej prawdopodobną alternatywą.

Najbardziej prawdopodobnym kandydatem jest słowo ze słownika, które ma najkrótszą odległość od podejrzanego typu. Odległość między dwoma słowami jest z grubsza określona przez liczbę kroków potrzebnych do przekształcenia jednego w drugie. Wielokrotne naciskanie klawisza `C-;` powoduje przechodzenie przez dostępne opcje, aż do powrotu do oryginalnej wersji. Obszar echa pokazuje listę możliwych poprawek. Wciśnięcie dowolnego innego klawisza przerywa łańcuch. Aby więc poprawić dwa błędy ortograficzne, należy najpierw poprawić jeden, wcisnąć dowolny klawisz inny niż `C-;` i poprawić kolejne problematyczne słowo.

Oprogramowanie `Hunspell` ma dostęp do zbioru słowników, w tym różnych odmian języka angielskiego, które należy zainstalować osobno. Domyślnym słownikiem dla EWS jest angielski australijski (`en_AU`). Jeśli nie jesteś Australijczykiem, musisz dostosować zmienną `ews-hunspell-dictionaries`, aby zmienić domyślny język, używając formalnej nazwy słownika `Hunspell`, na przykład `de_CH` dla niemieckiego szwajcarskiego lub `nl_NL` dla holenderskiego.

Jeśli piszesz w więcej niż jednym języku, dodaj wiele słowników, dodając je do zmiennej `ews-hunspell-dictionaries` w postaci ciągu znaków oddzielonych przecinkami. Na przykład, aby zaakceptować zarówno słowa australijsko-angielskie, jak i holenderskie, dostosuj tę zmienną do „`en_AU,nl_NL`” i uruchom ponownie Emacs (`restart-emacs`). Wszelkie zmiany w słownikach

zmiany wezmą efekt tylko wtedy, gdy wybrany słownik jest dostępny w systemie. Zmienna `ispell-hunspell-dict-paths-alist` zawiera listę zainstalowanych słowników.

4.7 Więcej informacji

Są to podstawowe informacje na temat funkcji, które EWS dodaje do podstawowego systemu Emacs. Pozostała część tej książki opisuje proces pracy z EWS, przedstawiając liniowy proces pisania od inspiracji do publikacji, jak wyjaśniono w sekcji 2.5.

Repozytorium EWS GitHub zawiera również plik `org-demo.org`, który zawiera przykłady funkcji opisanych w tym rozdziale. Najlepszym sposobem nauki jest praktyka, więc uruchom komputer i przygotuj się na przygodę.

Inspiracja: czytaj, słuchaj i oglądaj

Często zadawane pytanie autorom i innym twórczym osobom dotyczy tego, skąd czerpią swoje pomysły. Odpowiedzi na to proste pytanie są często mistyczne i niejasne, ale znaczna część odpowiedzi jest prosta. Autorzy potrzebują trzech źródeł inspiracji: wyobraźni, umiejętności pisarskich i dzieł innych pisarzy. Myśli i pomysły innych myślicieli stanowią fundament, na którym autorzy budują swoje dzieła. W 1575 roku sir Isaac Newton napisał list do Roberta Hooke'a, w którym zawarł słynne zdanie: „Jeśli widziałem dalej, to tylko dlatego, że stałem na ramionach gigantów”. Cytat ten może wydawać się banalny, ale pozostaje prawdziwy nawet w erze generatywnej sztucznej inteligencji.

Opieranie się na pracach innych twórców jest niezbędne nie tylko w pisaniu akademickim, które podlega ścisłym konwencjom dotyczącym cytowania. Wszyscy autorzy muszą czytać, słuchać i oglądać, aby skutecznie pisać. Dlatego pierwszym krokiem w procesie pracy *Zmaes Wvities Studio* (EWS) jest przyswajanie informacji stworzonych przez innych poprzez czytanie, słuchanie lub oglądanie treści.

Mimo że Emacs jest edytorem tekstu, może wywoływać inne programy do wyświetlania e-booków i odtwarzania plików multimedialnych. Oprócz tego, że jest to oprogramowanie elastyczne, Emacs jest również interfejsem do innych

oprogramowania. Oznacza to, że możemy używać Emacsa do zadań innych niż pisanie i edytowanie tekstu. Możesz również używać Emacsa do słuchania muzyki lub podcastów oraz oglądania filmów przy pomocy dodatkowego oprogramowania.

Śledzenie swojej kolekcji literatury fizycznej i elektronicznej to kolejne istotne zadanie naukowca. Emacs może również pomóc w tworzeniu przeszukiwalnej bibliografii i archiwum multimediów, aby ułatwić dostęp do zasobów.

W tym rozdziale wyjaśniono, jak używać Emacsa do czytania dokumentów elektronicznych w najpopularniejszych otwartych formatach, surfowania po Internecie i korzystania z plików multimedialnych. W rozdziale tym pokazano również, jak utworzyć bazę danych tekstową do zarządzania zbiorami książek elektronicznych i plików multimedialnych.

5.1 Czytanie e-booków

W erze przedcyfrowej ludzie dzielili się informacjami w formie fizycznej, od glinianych tabliczek po zwoje i papierowe książki. Przed pojawieniem się internetu jedynym sposobem na znalezienie nowych informacji było odwiedzanie biblioteki i spędzanie godzin i dni na przeglądaniu katalogów i książek. W ciągu ostatnich dwóch dekad dokumenty cyfrowe stały się normą, a internet praktycznie wyeliminował potrzebę odwiedzania biblioteki.

Większość e-booków nie jest dystrybuowana w postaci zwykłego tekstu, ale w formatach takich jak PDF, ePub lub DjVu. Godnym uwagi wyjątkiem jest Project Gutenberg (gutenberg.org), obszerna biblioteka elektroniczna zawierająca ponad 70 000 e-booków wolnych od praw autorskich. Biblioteka ta powstała w 1971 roku i dystrybuje książki w postaci zwykłego tekstu oraz innych otwartych formatów. Oprócz literatury klasycznej dystrybuje również różnorodne gatunki i tematy, co czyni ją cennym źródłem literatury wolnej od praw autorskich.

Emacs może renderować wyłącznie e-booki w formatach otwartych. Format otwarty, w przeciwieństwie do formatu zastrzeżonego, ma publicznie dostępną specyfikację, co pozwala każdemu na tworzenie oprogramowania do tworzenia i odczytywania tych plików. Najważniejszą zaletą formatu otwartego jest to, że każde urządzenie może wyświetlać tekst. Emacs nie może odczytywać formatów zastrzeżonych, takich jak Apple iBooks i Kindle. Zastrzeżone formaty plików stanowią poważne zagrożenie dla ciągłości kulturowej

. Dlatego niektóre rządy wymagają otwartych formatów dokumentów, aby egzekwować suwerenność cyfrową i uwolnić się od dominacji dużych korporacji (Pohle & Thiel, 2020).

Emacs potrzebuje pomocy innego oprogramowania, aby wyświetlać e-booki i dokumenty edytora tekstu. Schemat na rysunku 5.1 ilustruje programy niezbędne do renderowania stron z obsługiwanych formatów plików. Oprogramowanie to konwertuje każdą stronę na bufor zwykłego tekstu lub obraz PNG, który Emacs może wyświetlić.

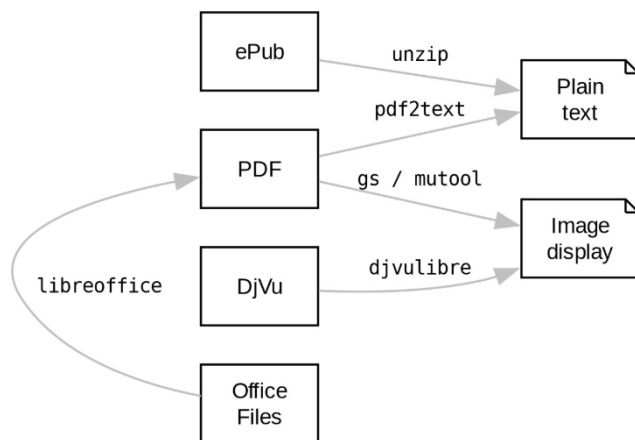
Do odczytu najpopularniejszych formatów e-booków wymagane jest następujące oprogramowanie:

- Ghostscript (gs) i/lub MuPDF (mutool): te dwa pakiety mogą konwertować pliki PDF na obrazy.
- Poppler (pdftotext): kolejne narzędzie do renderowania plików PDF i konwertowania ich na zwykły tekst.
- LibreOffice (soffice): pakiet oprogramowania biurowego do przeglądania dokumentów biurowych.
- DjVuLibre (ddjvu): czytanie e-booków w formacie DjVu.
- Rozpakuj: Wyodrębnij pliki z plików ZIP, aby móc czytać książki w formacie ePub.

Konfiguracja EWS wyświetla ostrzeżenie, gdy którykolwiek z tych pakietów jest niedostępny. Emacs będzie działał normalnie, ale niektóre funkcje mogą być niedostępne. Ostrzeżenia są wyświetlane w buforze komunikatów, który otwiera się w innym oknie za pomocą C-h e (view-echo-area-messages).

Pliki PDF

Portable Document Format (PDF) to wszechstronny system opracowany przez firmę Adobe na początku XXI wieku. Format PDF zapewnia spójną prezentację dokumentów, niezależnie od oprogramowania, sprzętu lub



Rysunek 1: Gotowe dokumenty do wydruku w formacie Zmaes.

system operacyjny używany do ich przeglądania. Format PDF jest skodyfikowany w międzynarodowej normie (ISO 32000) i stał się głównym otwartym formatem literatury elektronicznej.

Każdy plik PDF zawiera kompletny opis dokumentu o stałym układzie. Plik zawiera tekst, czcionki, układ, typografię, obrazy i inne informacje potrzebne do wyświetlenia treści.

Plik PDF jest zgodny z tradycyjnym układem fizycznym i typografią, zakładając, że dokumenty są drukowane. Dokument PDF nie jest rzeczywistym e-bookiem, ponieważ jego układ jest stały, w przeciwieństwie do innych formatów elektronicznych, które dostosowują się do rozmiaru ekranu.

Emacs może wyświetlać pliki PDF dzięki wbudowanemu trybowi DocView, z pewną pomocą GhostScript lub MuPDF, które konwertują pliki na obrazy. Otwórz plik PDF tak samo, jak każdy inny dokument. Aby poruszać się po pliku, użyj klawiszy strzałek oraz Page Up / Page Down. Dostępnych jest kilka innych skrótów klawiaturowych, które ułatwiają poruszanie się po dokumencie (tabela 5.1).

Jeśli w systemie dostępny jest program mutool, można również użyć polecenia imenu (M-g i) do wyświetlenia spisu treści i przejścia do rozdziału, zakładając, że plik PDF zawiera zakładki.

Aby umożliwić przeszukiwanie pliku PDF, potrzebne jest narzędzie `pdftotext`, które jest częścią pakietu oprogramowania Poppler. Ciekawostka: nazwa tego narzędzia pochodzi od odcinka serialu animowanego *Futuvama*.

W dokumencie można wyszukiwać za pomocą polecenia `doc-view-search` (przypisanego do klawisza C-s), które tworzy listę wszystkich pasujących stron i pokazuje, ile stron zawiera wyszukiwane hasło. Po zakończeniu wyszukiwania można przejść do następnej strony zawierającej pasujące wyniki, naciskając ponownie klawisz C-s.

DocView nie podświetla wyszukiwanego terminu, ale naciśnięcie C-t powoduje wyświetlenie wyników wyszukiwania dla tej strony w podpowiedzi (`doc-view-show-tooltip`). Poppler umożliwia również wyświetlenie pliku PDF jako zwykłego pliku tekstowego za pomocą skrótu klawiszowego C-c C-t (`doc-view-open-text`). Opcja ta ułatwia wyszukiwanie i kopiowanie odpowiednich fragmentów tekstu do notatek. Aby powrócić do graficznego widoku tekstu, należy dwukrotnie nacisnąć klawisze C-c C-c.

Tabela /.1: Skróty klawiaturowe Doe-View.

Klawisz	Opis
P	Powiększ do pełnego ekranu
W	dopasuj szerokość do okna
H	dopasuj wysokość do okna
+ / -	Powiększ i pomniejsz
M-g g	Przejdź do strony
M-< / M->	Przejdź do pierwszej lub ostatniej strony
k	Zamknij (usuń) plik
?	Plik pomocy

Oficjalne dokumenty

Pakiet DocView może również odczytywać dokumenty Microsoft Office i LibreOffice. Aby włączyć tę funkcję, należy zainstalować pakiet oprogramowania LibreOffice, bezpłatny pakiet oprogramowania biurowego typu open source.

Podczas otwierania dokumentu biurowego Emacs uruchamia LibreOffice (`soffice`) w celu konwersji pliku do formatu PDF i wyświetlenia go w DocView (rysunek 5.1), wykorzystując funkcje opisane w poprzedniej sekcji. Metodę tę można stosować nie tylko do otwierania plików edytora tekstu

, ale także prezentacje i arkusze kalkulacyjne, które przed wyświetleniem są konwertowane do formatu PDF.

Dokumenty biurowe są skompresowanymi plikami XML, więc po otwarciu ich za pomocą narzędzia do archiwizacji można wyświetlić ich zawartość w postaci zwykłego tekstu. Niestety istnieją dwa standardy dokumentów biurowych. Darmowe oprogramowanie, takie jak LibreOffice, wykorzystuje otwarty standard, podczas gdy komercyjni dostawcy oprogramowania często stosują inną wersję. Różnice między tymi standardami mogą powodować niewielkie problemy z formatowaniem podczas odczytu plików utworzonych za pomocą oprogramowania komercyjnego.

Książki DjVu

DJVU (wymawiane *jako déjà vu*) to format plików przeznaczony do skanowanych książek. Ponieważ plik DJVU może zawierać wysokiej jakości kolorowe obrazy, zdjęcia, tekst i rysunki, jest często używany do archiwizacji dokumentów. Pliki DjVu mają rozszerzenie .djvu lub .djv. DocView może odczytywać pliki DjVu, jeśli w systemie jest dostępna biblioteka DjVuLibre.

DocView wyświetla pliki PDF, dokumenty biurowe i dokumenty DjVu, więc te same funkcje, które są dostępne podczas przeglądania plików PDF, mają również zastosowanie do tych innych formatów. Więcej szczegółowych informacji na temat tego pakietu można znaleźć w podręczniku Emacs, który można szybko znaleźć za pomocą polecenia `C-h r g docu`.

Ograniczenia programu DocView

DocView ma pewne ograniczenia w porównaniu z innymi przeglądarkami dokumentów. Tekst jest wyświetlany jako plik PNG, co ogranicza możliwości wyszukiwania i uniemożliwia kopiowanie tekstu. Pakiet Pdf Tools autorstwa Vedanga Manerikara jest bardziej wszechstronny niż DocView. Niestety nie jest on zawarty w EWS, ponieważ jego instalacja w systemach innych niż Linux jest skomplikowana. Emacs Application framework to kolejne narzędzie, które zapewnia ulepszoną przeglądarkę plików PDF, oparty na języku Python.

W ostatniej części tego rozdziału wyjaśniono, jak skonfigurować Emacs, aby otwierał pliki binarne, takie jak PDF, w innym oprogramowaniu.

Pliki ePub

Plik ePub (Electronic Publication) to powszechnie stosowany otwarty format plików cyfrowych książek, czasopism i innych treści pisemnych. W przeciwieństwie do formatów PDF i DjVu, zawartość dostosowuje się do geometrii ekranu, dzięki czemu idealnie nadaje się do czytników e-booków, tabletów i innych urządzeń. Plik ePub to strona internetowa w pudełku. Możesz wyświetlić surową zawartość pliku ePub po otwarciu go za pomocą narzędzia do archiwizacji. Plik składa się ze zbioru plików HTML, które definiują zawartość i wygląd książki, a także wszelkich plików graficznych użytych do ilustracji. Ten format pliku wymaga pakietu Unzip do wyświetlenia zawartości.

Pakiet nov autorstwa Vasilija Schneidermanna zapewnia funkcjonalność przeglądania książek w formacie ePub w programie Emacs. Otwórz plik ePub i przewijaj go za pomocą spacji, klawiszy strzałek lub klawiszy Page Up/Page Down. Dostępnych jest kilka skrótów klawiaturowych umożliwiających poruszanie się po książce (tabela 5.2).

Tabela 5.2: Skróty klawiaturowe pakietu Nov.

Skrót klawiszowy	Funkcja	Opis
t	nov-goto-toc	Spis treści
n	nov-next-document	Następny rozdział
p	poprzedni-dokument	Poprzedni rozdział
q	zamknij okno	Zamknij
?	tryb opisywania	Bufor pomocy

Aby zwiększyć lub zmniejszyć rozmiar tekstu, użyj skrótów klawiaturowych C-x C-+ i C-x C-- (text-scale-adjust). Podczas powiększania czcionki niektóre fragmenty tekstu mogą zostać przesunięte poza okno. Aby zresetować długość wierszy, naciśnij klawisz g, aby ponownie wyrenderować dokument (nov-render-document).

Możesz kopiować i wklejać tekst z plików ePub do notatek bibliograficznych (rozdział 5) za pomocą polecenia kill-ring-save (M-w). Można również kopiować obrazy z pliku ePub. Otwórz plik jako archiwum za pomocą klawisza a (nov-reopen-as-archive), który pokazuje wewnętrzną strukturę dokumentu. Stąd można przejść do odpowiedniego pliku obrazu, skopiować go za pomocą klawisza C (archive-copy-file) i wybrać nową lokalizację oraz nazwę. Niestety nie ma funkcji podglądu obrazów w archiwum.

5.2 Mamaging – Twoja biblioteka cyfrowa

Posiadanie dużej kolekcji literatury może szybko doprowadzić do chaosu, dlatego większość naukowców korzysta z elektronicznej bibliografii, aby śledzić przeczytane pozycje. Emacs może pomóc w stworzeniu katalogu bibliotecznego w postaci zwykłego tekstu, który ułatwi dostęp do bibliografii. Bibliografia ta może również zawierać linki do notatek (rozdział 5) i ułatwiać cytowanie prac naukowych (rozdziały 7 i 8).

Tworzenie bibliografii i zarządzanie nią wymaga trzech pakietów Emacsa, które płynnie się ze sobą integrują. Wbudowany tryb BibTeX pomaga w tworzeniu i utrzymywaniu bibliografii w postaci zwykłego tekstu. Pakiet Biblio autorstwa Clémenta Pit-Claudela przeszukuje internetowe bazy danych naukowych i wstawia odpowiednie pozycje do bibliografii. Pakiet Citar autorstwa Bruce'a D'Arcusa zapewnia łatwy dostęp do bibliografii za pomocą systemu uzupełniania minibuffera. Te trzy pakiety zmieniają Emacsa w w pełni funkcjonalny system zarządzania literaturą.

Rozpoczęcie pracy z trybem BibTeX

Tryb BibTeX jest głównym trybem służącym do tworzenia bibliografii i zarządzania nimi. Jak sugeruje nazwa, tryb ten wykorzystuje format plików BibTeX jako domyślną bazę danych literatury. BibTeX jest formatem zwykłego tekstu służącym do organizowania literatury i cytatów. Jest on zazwyczaj używany do składu publikacji naukowych za pomocą LaTeX (Lamport, 1yy4). Typowy wpis dotyczący książki będzie wyglądał następująco:

```
@article{stallman_1981_emacs,
  tytuł      = {EMACS – rozszerzalny, konfigurowalny
                edytor tekstu z funkcją
                samodokumentacji},
  autor      = {Stallman, Richard M.},
  rok        = 1981,
  czasopismo = {Biuletyn ACM SIGOA}, tom
                = 2,
  numer      = {1-2},
  strony     = {147-156},
  doi        = {10.1145/1159890.806466},
```

```
słowa kluczowe      = {Emacs}
plik                 = {informatyka/stallman-1981-emacs.pdf}
}
```

Każdy wpis zaczyna się od znaku @ i typu publikacji (książka, artykuł lub inne typy), po którym następuje nawias klamrowy i unikalny klucz cytatu. Kolejne wiersze zawierają odpowiednie dane dotyczące tego wpisu. BibTeX może przetwarzać różne rodzaje literatury, w tym artykuły i referaty konferencyjne, z których każdy ma swoje własne, specyficzne typy pól. BibTeX ignoruje nieznane pola, co daje możliwość wykorzystania formatu do innych celów, takich jak dołączanie plików i słów kluczowych.

Możesz przechowywać jeden lub więcej plików bibliograficznych w folderze, aby mieć do nich dostęp z dowolnego miejsca w Emacs. W EWS zmienna `ews-bibtex-directory` definiuje ten folder. Należy dostosować tę zmienną do żądanej lokalizacji, która domyślnie jest ustawiona na `~/library`. Tilde na początku wskazuje ścieżkę do katalogu domowego. Po dostosowaniu tej zmiennej należy uruchomić funkcję `ews-bibtex-register`, aby zarejestrować pliki bibliograficzne w tym katalogu. Funkcję tę należy uruchamiać za każdym razem, gdy dodaje się lub usuwa plik BibTeX z tego folderu.

Nie musisz instalować zewnętrznego oprogramowania, aby rozpocząć pracę. Wystarczy utworzyć pusty plik w katalogu bibliografii z rozszerzeniem `.bib`, a Emacs włączy tryb BibTeX po otwarciu pliku.

BibTeX to szanowane narzędzie, które nadal ma znaczenie, choć ma pewne ograniczenia. BibLaTeX to zmodernizowana wersja, która umożliwia różne style wyjściowe i integrację z nowoczesnymi klasami dokumentów LaTeX. Zaawansowani użytkownicy LaTeX mogą również korzystać z wariantu BibLaTeX, ustawiając zmienną `bibtex-dialect` na `biblatex`.

Dodawanie nowych pozycji

Tryb BibTeX w Emacsie wykorzystuje szablony do dodawania nowych wpisów. Aby dodać nowe odniesienie, użyj funkcji

`bibtex-entry` (C-c C-b) i wybierz odpowiedni typ publikacji. Emacs

udostępnia również skróty dla każdego rodzaju literatury. Listę tych poleceń można wyświetlić w pliku BibTeX za pomocą kombinacji klawiszy C-c C-e ?.

Większość pól jest opcjonalna, ale każdy typ literatury ma co najmniej jedno pole obowiązkowe. Pola opcjonalne zaczynają się od „OPT”. Należy wypełnić co najmniej jedno pole zaczynające się od „ALT”, np. autor lub redaktor. W przypadku książki, jak pokazano w poniższym przykładzie, pola tytułu, wydawcy i roku są obowiązkowe, a należy wypełnić pole autora lub redaktora albo oba te pola. Wszystkie pozostałe pola są opcjonalne. Każdy rodzaj literatury ma swój własny szablon.

```
@Książka{,
  ALTAuthor      = {},
  ALTeditor      = {},
  title          = {},
  wydawca        = {},
  rok            = {},
  OPTkey         = {},
  OPTvolume      = {},
  OPTnumber      = {},
  OPTseries      = {},
  OPTaddress     = {},
  OPTedition     = {},
  OPTmonth       = {},
  OPTnote        = {},
  OPTkeywords    = {},
  OPTfile        = {}
}
```

Przechodź między polami za pomocą klawisza C-j (bibtex-next-field) i uzupełnij wszystkie wymagane pola i jedno z pól ALT. Po zakończeniu naciśnij C-c C-c (bibtex-clean-entry), aby sprawdzić składnię i usunąć puste pola. Funkcja ta przypisuje również unikalny klucz cytowania do wpisu, korzystając z pewnych konfigurowalnych reguł. Możesz nadpisać ten klucz cytowania i ustawić go ręcznie, o ile jest on unikalny. Tryb BibTeX wyświetla ostrzeżenie, gdy wykryje zduplikowane klucze. Aby wyczyścić bufor i wyrównać pola, użyj funkcji bibtex-fill-entry

(C-c C-q). Polecenie to usuwa również zbędne nawiasy klamrowe, aby uzyskać przejrzysty wygląd. Tabela 5.3 zawiera podsumowanie najważniejszych skrótów klawiaturowych i funkcji w trybie BibTeX.

Aby wprowadzić nazwiska autorów lub redaktorów, należy wpisać najpierw nazwisko, a następnie przecinek i imię lub inicjały. Dodatkowych autorów należy oddzielić słowem „i”, np. „Hawking, S. i Penrose, R.”. Jeśli skopiujesz i wkleisz nazwisko autora, imię i nazwisko mogą być odwrócone. Dobrym trikiem w Emacsie jest użycie funkcji `org-transpose-words`, przypisanej do M-t, która zamienia kolejność dwóch słów po lewej i prawej stronie kursora. Na przykład, przekształć „Stephen Hawking” na „Hawking Stephen” za pomocą M-t i dodaj przecinek po nazwisku, aby zakończyć.

Tabela 5.3: Przegląd skrótów klawiaturowych do dodawania i edytowania plików.

Klawisz	Funkcja	Opis
C-c C-b	<code>bibtex-entry</code>	Dodaj wpis dla wybranego typu
C-c d	<code>bibtex-empty-field</code>	Opróżnij bieżące pole
C-j	<code>bibtex-next-field</code>	Przejdź do następnego pola
C-down	<code>bibtex-next-entry</code>	Przejdź do następnego wpisu
C-up	<code>bibtex-previous-entry</code>	Przejdź do poprzedniego wpisu
C-c C-c	<code>bibtex-clean-entry</code>	Wyczyść wpis
C-c C-q	<code>bibtex-fill-entry</code>	Wyrównaj pola

W tej sekcji przedstawiono krótki przegląd możliwości tego pakietu. Dokumentacja tego pakietu jest skąpa. Jonathan Le Roux (jonathanleroux.org) udostępnia na swojej stronie internetowej obszerny podręcznik, który szczegółowo wyjaśnia funkcjonalność tego pakietu. Można również zapoznać się z innymi funkcjami trybu BibTeX, używając polecenia M-x `^bibtex` i przeglądając opcje na liście uzupełnień.

Attacheets

Konfiguracja EWS zapewnia dodatkowe pola do kategoryzowania literatury za pomocą słów kluczowych oraz dołączania jednego lub więcej plików. Załączniki BibTeX powinny również znajdować się w katalogu `ews-bibtex` lub podkatalogu. Wszystkie ścieżki plików dla wpisów BibTeX zaczynają się w tej lokalizacji.

Na przykład, gdy wpis BibTeX zawiera: `file={topology/article.pdf}`, załącznik jest przechowywany w lokalizacji: `~/library/topology/article.pdf`. Każdy wpis może mieć więcej niż jeden załącznik, oddzielone średnikiem. Załączniki BibTeX nie mają formalnej konwencji nazewnictwa plików, więc można używać dowolnej metody. Sekcja y.2 zawiera wskazówki dotyczące tworzenia spójnej konwencji nazewnictwa plików.

EWS udostępnia dwie wygodne funkcje zapewniające integralność powiązań między plikami BibTeX a załącznikami. Idealnym stanem jest sytuacja, w której wszystkie pliki wymienione w wpisach BibTeX faktycznie istnieją i odwrotnie: wszystkie pliki w katalogu bibliografii są wymienione w BibTeX.

Funkcja `ews-bibtex-missing-attachments` wyświetla listę wszystkich brakujących załączników w buforze Messages. Aby naprawić tę rozbieżność, należy usunąć lub edytować pole pliku w odpowiednim wpisie BibTeX lub przywrócić nazwę pliku w swojej kolekcji. Funkcja `ews-bibtex-missing-files` wyświetla listę wszystkich załączników w katalogu bibliografii, które nie są zarejestrowane w plikach BibTeX. Aby rozwiązać ten problem, należy zmienić nazwę odpowiedniego pliku lub dodać go do powiązanego wpisu BibTeX w polu pliku. Te dwie funkcje pomagają zapewnić stały dostęp do literatury elektronicznej za pośrednictwem menu Citar.

Dodaj eetvies fvom oeliee souvees

Tryb BibTeX wymaga ręcznego wpisywania wszystkich wpisów, co jest nieefektywne i może łatwo prowadzić do błędów. Pakiet Biblio autorstwa Clémenta Pit-Claudela umożliwia przeglądanie i importowanie odniesień bibliograficznych ze źródeł internetowych w celu przeprowadzenia systematycznego przeglądu literatury. Obecnie pakiet umożliwia wyszukiwanie w serwisach CrossRef, DBLP, arXiv, doi.org i Dis-semin.

Crossref łączy miliony pozycji z różnych rodzajów treści, w tym czasopism, książek, materiałów konferencyjnych, grantów badawczych, dokumentów roboczych, raportów technicznych i zbiorów danych. Połączone treści obejmują materiały z dziedzin naukowych, technicznych i medycznych (STM), a także nauk społecznych i humanistycznych (SSH). DBLP to komputer

strona internetowa zawierająca bibliografię naukową, obejmująca ponad siedem milionów publikacji. arXiv (wymawiane „archive”) to ogólnodostępne repozytorium przedruków i powtórnych wydruków zatwierdzonych do publikacji po moderacji, ale niepoddanych recenzji. W matematyce i fizyce prawie wszystkie artykuły naukowe są samodzielnie archiwizowane w repozytorium arXiv przed publikacją w recenzowanym czasopiśmie.

Większość publikacji elektronicznych posiada identyfikator obiektu cyfrowego (DOI), trwały kod identyfikacyjny, który odsyła do metadanych dotyczących publikacji. System DOI gwarantuje, że publikacje można znaleźć nawet w przypadku zmiany ich adresów. Identyfikatory DOI są szeroko stosowane do identyfikacji informacji akademickich, zawodowych i rządowych, w tym artykułów naukowych, raportów badawczych, zbiorów danych i publikacji urzędowych.

Aby skorzystać z pakietu Biblio, otwórz odpowiedni plik BibTeX, uruchom biblio-lookup, wybierz odpowiednią bazę danych i wprowadź zapytanie. Gdy wyniki wyszukiwania będą dostępne, otworzy się nowy bufor. Wybierz cel za pomocą klawiszy strzałek lub przeszukaj bufor za pomocą klawisza C-s. Po znalezieniu potrzebnej literatury wstaw jej rekord BibTeX do bufora, w którym wywołałeś funkcję za pomocą klawisza i. Alternatywnie możesz skopiować rekord BibTeX za pomocą klawisza c i wkleić go później w odpowiednim miejscu. Wyniki wyszukiwania można zamknąć za pomocą klawisza q. Aby wyświetlić wszystkie możliwe polecenia w tym buforze, użyj klawisza h.

Funkcja biblio-doi-insert-bibtex wstawia rekord BibTeX oparty na numerze DOI do bieżącego bufora. Można wprowadzić tylko identyfikator w jednym z dwóch wyżej wymienionych formatów. Polecenie to należy aktywować z poziomu bufora BibTeX, umieszczając kursor w miejscu, w którym ma pojawić się nowy wpis.

Niestety, duże wydawnictwa korporacyjne nadal trzymają światową wiedzę akademicką za zamkniętymi drzwiami. Jednak publikacje w otwartym dostępie i przedrukowane powoli stają się normą. Witryna Dissemin wyszukuje ogólnodostępne kopie artykułów w obszernej kolekcji otwartych repozytoriów i witryn internetowych. Aby skorzystać z tej usługi w Emacs, należy uruchomić dissemin-lookup, aby wyświetlić informacje o statusie otwartego dostępu do artykułu przy użyciu numeru DOI. Można również nacisnąć x w menu wyszukiwania, aby skorzystać z funkcji biblio-lookup i sprawdzić, czy dostępna jest wersja otwartego dostępu.

Pakiet Biblio jest przydatny, ale jedną z niewielkich niedogodności jest konieczność przejścia do

odpowiedni plik bibliograficzny przed dodaniem nowego wpisu. Zapewnia również dwie oddzielne funkcje wyszukiwania, jedną dla DOI, a drugą dla baz danych.

Specjalna funkcja `ews-biblio-lookup` (C-c w b b) pozwala wybrać plik Bib-TeX, w którym chcesz zapisać wyniki wyszukiwania, przed wybraniem jednego z dostępnych źródeł. Funkcja ta łączy również wyszukiwanie DOI z innymi źródłami, eliminując niektóre etapy z procesu pracy.

Access bibliographies

Prowadzenie jednego lub kilku plików BibTeX do przechowywania biblioteki to dobry początek, ale zawartość nie jest łatwa do wyszukiwania i uzyskania dostępu, zwłaszcza gdy masz wiele bibliografii. Potrzebujemy interfejsu, który ułatwi lokalizowanie i uzyskiwanie dostępu do literatury na komputerze, w tym zasobów takich jak linki, załączniki i notatki.

Pakiet Citar wykorzystuje uzupełnianie minibuffera, aby uzyskać dostęp do bibliografii, zapewniając dostęp do hiperłączy i załączników. Citar zapewnia również dostęp do notatek dotyczących literatury (rozdział 5) i integruje się z modulem cytowania Org (rozdział 7).

Globalna bibliografia Emacs to zestaw plików BibTeX dostępnych z dowolnego miejsca w Emacsie, znajdujących się w katalogu `ews-bibtex-directory`. Citar przetwarza wszystkie pliki bibliograficzne w folderze globalnym oraz wszystkie pliki, do których odwołuje się aktywny bufor Org. Jeśli dodasz nowy plik do globalnej bibliografii, musisz poinformować o tym program Citar za pomocą funkcji `ews-bibtex-register` (C-c w b r). Funkcja ta rejestruje wszystkie pliki bibliograficzne w wyznaczonym katalogu do wykorzystania przez program Citar i wyświetla je w obszarze echo. Funkcja ta służy wyłącznie do rejestrowania nowych plików; nie jest wymagana w przypadku dodawania nowych pozycji do istniejącego pliku.

Citar aktywuje się za pomocą polecenia `citar-open`, które w EWS jest przypisane do skrótu klawiszowego C-c w b o. W minibufferze pojawia się menu, z którego można przeszukiwać swoją kolekcję. Pierwsze trzy kolumny w menu wskazują, które wpisy zawierają hiperłącze (L), jeden lub więcej załączonych plików (F) oraz powiązaną notatkę (N). Pozostała część pokazuje autora, rok, tytuł, klucz cytowania i słowa kluczowe (rysunek 5.2).

U:--- *scratch* All L4 (Lisp Interaction super-save EIDoc)				
1/111 References: (0/1835): math				
Select Multiple [TAB]				
N	Knuth	1974	Surreal Numbers: How Two Ex-Students Turned on t	knuth_1
F N	Cronbach, Meehl	1955	Construct Validity in Psychological Tests	cronbac
F N	Eglash	2013	An Ethnocomputing Comparison of African and Nati	eglash_
N	Acheson	2010	1089 and All That: A Journey into Mathematics	acheson
N	Aczel	2015	Finding Zero: A Mathematician's Odyssey to Uncov	aczel_2
N	Adams	1994	The Knot Book: An Elementary Introduction to the	adams_1
F	Adams, Franzosa	2008	Introduction to Topology. Pure and Applied	adams_2
L F N	Alivizatos	2012	Debating heritage authenticity: kastom and devel	aliviza
F N		2013	The Mathematics Tarot	anchori

Rysunek 5.2: Przykład menu Citar.

Wyszukiwanie literatury za pomocą programu Citar jest łatwe dzięki możliwościom pakietów Vertico i Orderless, opisanych w rozdziale 4. Po każdym naciśnięciu klawisza program Citar zawęży listę opcji do odpowiednich wyników. Wybierz odpowiednią pozycję za pomocą klawiszy strzałek lub użyj klawisza Tab, aby wybrać więcej niż jedną pozycję. Możesz filtrować menu uzupełniania Citar pod kątem wpisów z załącznikami, używając „:f”, oraz pod kątem linków, używając „:l”. Po naciśnięciu klawisza Enter Citar wyświetla menu podręczne w minibufferze, w którym można otworzyć załączniki lub przejść do hiperłączy wymienionych we wpisie BibTeX.

Usieg tlivd-pavty software

Jeśli posiadasz już narzędzie do zarządzania bibliografią, takie jak Zotero, możesz nadal z niego korzystać, aby osiągnąć ten sam rezultat. Użytkownicy Zotero mogą eksportować biblioteki jako pliki BibTeX lub BibLa-TeX. Pakiet Zotero Better BibTeX autorstwa Emiliano Heynsa umożliwia synchronizację eksportowanego pliku z Zotero, co pozwala na płynne korzystanie z niego w ramach Emacsa. Większość pakietów oprogramowania do zarządzania referencjami posiada podobną funkcjonalność.

5.ç Surfowanie po Internecie

Emacs posiada również wbudowaną przeglądarkę internetową o nazwie Emacs Web Wowser (EWW). Pakiet ten przedkłada czytelność nad funkcjonalność, wyświetlając strony internetowe jako zwykły tekst. Może wyświetlać obrazy, ale nie renderuje CSS ani nie uruchamia JavaScript. Wowser to osoba o silnych poglądach moralnych dotyczących umiarkowania i abstynencji. Podejście oparte na zwykłym tekście

Przeglądanie stron internetowych stanowi etyczne stanowisko w sieci WWW, szczególnie w kwestiach bezpieczeństwa i prywatności. Nazwa ta może również odnosić się do reakcji, jaką można mieć, gdy po raz pierwszy zobaczy się stronę internetową wyświetlaną w postaci zwykłego tekstu. Czytanie stron internetowych w postaci zwykłego tekstu nie jest atrakcyjne wizualnie, ale jest to bezpieczniejszy sposób surfowania po Internecie, ponieważ EWW nie wyświetla żadnego kodu JavaScript.

Możesz otworzyć adres URL lub wyszukać informacje w Internecie za pomocą polecenia eww. Jeśli wprowadzone dane nie wyglądają jak adres URL, EWW wyszuka informacje w Internecie za pomocą DuckDuckGo, wyszukiwarki internetowej skupiającej się na prywatności, która nie śledzi Twojej aktywności online. Po załadowaniu strony użyj klawiszy strzałek i klawiszy Page Up/Page Down, aby poruszać się po stronie. Do poruszania się po stronie internetowej dostępnych jest kilka skrótów klawiaturowych.

- <, >: początek i koniec strony
- R: Format czytelny (wyświetla tylko główny tekst)
- G: Nowe wyszukiwanie lub strona internetowa
- H: Historia przeglądania
- M-I: Przełączanie obrazów
- l / n: Poprzednia i następna strona
- q: Zamknij okno
- w: Skopiuj adres URL pod kursorem lub adres URL strony
- &: Otwórz stronę w zewnętrznej przeglądarce
- ?: Plik pomocy z listą innych skrótów klawiaturowych

Przycisk Enter otwiera linki (eww-follow-link). Jeśli chcesz, aby nowa strona otworzyła się w nowym buforze, użyj M-<Ret> (eww-open-in-new-buffer).

Najbardziej przydatną opcją jest R, która pozwala zignorować elementy nawigacyjne strony i skupić się na treści. Jeśli strona nie wyświetla się w EWW lub pojawia się ostrzeżenie o konieczności użycia JavaScript, użyj znaku ampersand (&), aby przejść do domyślnej przeglądarki internetowej systemu.

Po otwarciu linku do strony internetowej z bufora Emacs innego niż EWW, strona zostanie otwarta w domyślnej przeglądarce systemu operacyjnego. Gdy już przyzwyczaisz się do przeglądania Internetu w postaci zwykłego tekstu, możesz ustawić EWW jako domyślną przeglądarkę do otwierania linków w Emacs. Aby wprowadzić tę zmianę, dostosuj zmienną `browse-url-browser-function`, wybierz „Emacs Web Wowser” z menu wartości i kliknij [Apply and Save] (Zastosuj i zapisz).

Jeśli znajdziesz stronę internetową, która Ci się podoba, dodaj ją do zakładek za pomocą polecenia `eww-add-bookmark`, przypisanego do klawisza b w programie EWW. Polecenie `eww-list-bookmarks` wyświetla listę wszystkich zapisanych zakładek, z której możesz wybrać jedną i odwiedzić stronę. Aby uzyskać więcej informacji, przeczytaj instrukcję obsługi programu EWW, wpisując polecenie C-h R eww.

Prowadzenie kolekcji zakładek jest pomocne, ale trzeba regularnie odwiedzać te strony, aby sprawdzić, czy pojawiły się na nich nowe treści. Wiele blogów i podcastów wykorzystuje kanały RSS do powiadamiania czytelników o nowych treściach, co jest tematem następnej sekcji.

Czytaj kanały RSS i Atom

Znalezienie interesujących treści w Internecie może przypominać przeszukiwanie stosów śmieci w poszukiwaniu czegoś wartościowego. Media społecznościowe mogą być zabawne i angażujące, ale kakofonia nieistotnych i obraźliwych treści generowanych przez mroczne algorytmy i boty jest przygnębiająca. Kanały RSS i Atom umożliwiają subskrypcję stron internetowych i blogów, które lubisz. Kanał to plik XML zawierający najnowsze treści ze strony internetowej, w postaci pełnego tekstu lub tylko fragmentu.

RSS (Really Simple Syndication) to elegancki mechanizm konsumowania treści, ponieważ wyświetla tylko blogi lub podcasty, które subskrybujesz. Kanały Atom są nowszym formatem, który wyjaśnia niektóre niejasności w RSS. Oba kanały są formą XML i do wyświetlenia ich zawartości potrzebny jest agregator.

Kiedy korzystasz z RSS, żaden algorytm nie decyduje o tym, co możesz zobaczyć, a czego nie. Subskrybuj

Subskrypcja kanałów RSS jest anonimowa, więc nie będziesz otrzymywać spamu z ofertami sprzedaży produktów lub usług. Niektóre strony internetowe oferują wiele kanałów, umożliwiając czytelnikom subskrypcję konkretnych tematów. Na przykład kanał RSS dla artykułów dotyczących Emacs'a na stronie internetowej towarzyszącej tej książce to:

<https://lucidmanager.org/tags/emacs/index.xml>

Niestety, kanały RSS straciły na znaczeniu ze względu na dominację mediów społecznościowych i preferencje właścicieli stron internetowych dotyczące gromadzenia adresów e-mail. Znalezienie kanałów RSS było kiedyś łatwe, ale duże firmy internetowe wolą polegać na swoich algorytmach typu „czarna skrzynka”, aby dostarczać treści użytkownikom. W związku z tym kanały RSS są praktycznie niewidoczne. Technologia ta pozostaje jednak żywa i nadal jest wykorzystywana na prawie wszystkich stronach internetowych, w tym na stronach poświęconych podcastom i YouTube. Przeglądarki nie łączą się już automatycznie z kanałami, a strony internetowe rzadko umieszczają do nich widoczne linki, tak jak kiedyś, ale kanały nadal istnieją.

Prawie połowa stron internetowych na świecie korzysta z WordPressa. Feed tych stron można znaleźć, dodając feed na końcu adresu URL. Jeśli wszystko inne zawiedzie, feed można zlokalizować, sprawdzając źródło strony (użyj klawisza v podczas przeglądania strony w EWW). Nie daj się przestraszyć kodowi HTML. Wyszukaj rss-xml i skopiuj adres URL w specyfikacji href.

Aby dodać kanały YouTube, potrzebujesz identyfikatora kanału. Adres URL można znaleźć w kodzie źródłowym strony głównej kanału, wyszukując „channel_id=” i dodając go do tego szablonu:

https://www.youtube.com/feeds/videos.xml?channel_id=<ID>

Pakiet Elfeed autorstwa Christophera Wellonsa agreguje Twoje ulubione kanały RSS. Możesz tworzyć listy i kategorie swoich ulubionych kanałów. Przeglądarka Elfeed pomaga w nawigacji po nieprzeczytanych artykułach, kanałach YouTube lub podcastach. Musisz zainstalować program cURL, co oznacza „Client for URLs” (klient dla adresów URL). Program ten pomaga w pobieraniu plików z Internetu. Jeśli cURL jest niedostępny, Elfeed używa wolniejszej wbudowanej metody Emacs do wyodrębniania danych, która nie działa na komputerach z systemem Windows.

Pakiet tworzy bazę danych do przechowywania kanałów. EWS ustawia lokalizację pobranych treści w folderze konfiguracyjnym Emacs. Skrót klawiaturowy EWS do uruchomienia Elfeed to C-c w e. Jednak przed rozpoczęciem czytania kanałów należy najpierw znaleźć kilka z nich i dodać je do pliku konfiguracyjnego, a następnie pobrać dane.

Pakiet Elfeed-Org konfiguruje listę ulubionych stron internetowych w pliku Org. Pakiet odczytuje wskazane pliki Org i zbiera adresy internetowe lub linki z nagłówków oznaczonych tagiem :elfeed:. Tag dla nagłówka Org mode ustawia się za pomocą C-c C-q. Poniższy przykład ilustruje, jak skonstruować plik Elfeed Org. Należy pamiętać, że tag ma zastosowanie do wszystkich nagłówków na niższym poziomie, więc tag :elfeed: ma również zastosowanie do nagłówków Emacs i news. Można również dodawać komentarze tekstowe, ponieważ Elfeed odczytuje tylko nagłówki.

```
#+title: Konfiguracja Elfeed
```

```
* Kanały                                :elfeed:
** Emacs                                :emacs:
```

```
Informacje dotyczące Emacs.
```

```
*** https://lucidmanager.org/tags/emacs/index.xml
```

```
*** http://www.reddit.com/r/emacs/.rss
```

```
** Aktualności                          :news:
```

```
*** https://www.abc.net.au/news/feed/2942460/rss.xml
```

Możesz użyć zwykłego adresu URL lub hiperłącza Org. Pakiet Org-Webtools wstawia w pełni sformatowane hiperłącza do trybu Org za pomocą kombinacji klawiszy C-c w w. Korzystając z przeglądarki EWW, skopiuj bieżący adres do bufora kill ring za pomocą klawisza w.

Jedyna konfiguracja, jakiej potrzebujesz dla elfeed-org, to dostosowanie nazwy i lokalizacji plików Org, których chcesz używać do przechowywania linków do kanałów. Lokalizacja konfiguracji Elfeed jest przechowywana w zmiennej rmh-elfeed-org-files, której domyślną wartością jest „~/elfeed.org”. Repozytorium GitHub dla tej książki zawiera przykładowy plik, który pomoże Ci rozpocząć pracę.

Musisz dostosować tę zmienną, aby odpowiadała plikowi, którego chcesz używać do przechowywania swoich kanałów. Aby dodać lub usunąć kanał, edytuj ten plik i zaktualizuj bazę danych za pomocą elfeed-update. Teraz możesz już czytać swoje kanały RSS.

Naciśnij C-c w e, aby uruchomić przeglądarkę Elfeed, która wyświetla listę z datą i tytułem każdego wpisu, nazwą kanału i wszelkimi tagami. Po naciśnięciu klawisza Enter Elfeed wyświetla stronę internetową lub podsumowanie z hiperłączem do wersji internetowej w osobnym oknie. Do zarządzania kanałem można używać następujących skrótów klawiszowych:

- G: pobierz aktualizacje kanału z serwerów
- b: otwórz artykuł w przeglądarce systemowej
- c: wyczyść filtr wyszukiwania
- g: Odśwież widok listy kanałów (usuń przeczytane elementy)
- q: Zamknij Elfeed
- r Oznacz wpis jako przeczytany
- s: Aktualizuj filtr wyszukiwania
- u: Oznacz wpis jako nieprzeczytany

Wszystkie nowe wpisy są domyślnie oznaczane jako nieprzeczytane. Pozostałe tagi pochodzą z listy kanałów RSS. Po usunięciu kanału z listy wszystkie wcześniej pobrane artykuły pozostaną w bazie danych i będą wyświetlane na liście do momentu ich przeczytania lub usunięcia. Elfeed posiada również zaawansowany filtr wyszukiwania, który można wykorzystać do filtrowania według tagów, nazw kanałów i dat.

5.4 System multimedialny Emacs

Muzyka jest doskonałym narzędziem zwiększającym produktywność. Odtwarzanie transcendentnych fug J.S. Bacha lub polirytmicznego metalu Sepultury koi duszę podczas pisania kolejnego wielkiego dzieła sztuki. Psycholog Sara Bottiroli i jej współpracownicy zbadali psychologiczny wpływ muzyki w tle i odkryli, że może ona poprawiać pamięć epizodyczną, inteligencję oraz szybkość przetwarzania werbalnego i wizualnego (Bottiroli, Rosi, Russo, Vecchi i Cavallini, 2014).

Emacs może być skromnym edytorem tekstu, ale może również ułatwić odtwarzanie muzyki w tle podczas pisania, słuchania podcastów lub wywiadów terenowych. Być może masz już świetny odtwarzacz muzyki na swoim komputerze, ale brak konieczności przełączania się między aplikacjami w celu odtworzenia muzyki pomaga zachować koncentrację. Emacs Multimedia System (EMMS) to kompleksowy odtwarzacz muzyki i wideo dla Emacs. Posiada intuicyjną przeglądarkę, która wyświetla okładki albumów i metadane, przekształcając system Emacs w osobisty odtwarzacz muzyczny.

Emacs wymaga dostępu do zewnętrznego odtwarzacza muzycznego, aby odtwarzać dźwięki, oraz do oprogramowania graficznego, aby konwertować okładki albumów na miniatury. Aby odtwarzać dźwięki w Emacsie, należy zainstalować jeden z kompatybilnych odtwarzaczy dźwięku (ogg123, mpg321, MPlayer, MPV lub VLC). Po uruchomieniu konfiguracji EWS system sprawdza, czy któryś z tych odtwarzaczy jest dostępny. Wszelkie brakujące oprogramowanie jest wyświetlane w buforze komunikatów po uruchomieniu EWS, do którego można uzyskać dostęp za pomocą C-h e.

Ostatnim krokiem w odtwarzaniu muzyki jest buforowanie plików muzycznych poprzez skanowanie kolekcji plików multimedialnych. Aby przeskanować swoją kolekcję, należy użyć polecenia `emms-add-directory-tree`. EMMS tworzy pamięć podręczną w katalogu konfiguracyjnym Emacs. EMMS odczytuje metadane w plikach muzycznych Ogg Vorbis, Opus, fLAC i MP3, a także w niektórych typach plików wideo. Proces ten buforuje również miniatury i może zająć trochę czasu, w zależności od rozmiaru kolekcji. Odtwarzanie muzyki można rozpocząć, gdy w minibufferze pojawi się komunikat EMMS: All track information loaded (Wszystkie informacje o utworach załadowane). Postęp skanowania można wyświetlić w buforze Messages (Komunikaty), do którego dostęp można uzyskać za pomocą klawisza C-h e. Aby dodać nową muzykę do istniejącej kolekcji, należy ponownie uruchomić to polecenie i wskazać nową kolekcję.

Podstawową zasadą odtwarzania plików muzycznych jest przenoszenie utworów do bufora listy odtwarzania.



Rysunek 3: Okno przeglądarki ZMMS.

Przeglądarka jest najwygodniejszym sposobem wybierania ulubionej muzyki. Przeglądarkę uruchamia się za pomocą emms-browser (C-c w m b), pokazanego na rysunku 5.3. Pierwsze załadowanie przeglądarki w sesji Emacsa może chwilę potrwać, jeśli masz obszerną kolekcję muzyczną.

Podczas korzystania z przeglądarki EMMS naciśnij klawisz b, a następnie cyfrę, aby przeglądać według wykonawcy, albumu, gatunku, roku, kompozytora lub wykonawcy. Przeglądarka jest domyślnie zwinięta i wyświetla tylko okładki albumów, nazwy wykonawców lub wybraną kategorię przeglądania. Przeglądarka ma strukturę hierarchiczną. Na przykład podczas przeglądania według gatunku hierarchia ma cztery poziomy: *Geeve* > *Avtist* > *Album* > *Tvaek*. Użyj klawisza 1, aby zwinąć, a klawiszy 2-4, aby rozwinąć kategorie na poziomach 1 do 3.

Podobnie jak w większości aplikacji Emacsa, naciśnij klawisz ?, aby wyświetlić listę dostępnych skrótów klawiszowych, takich jak:

- r: Przejdź do losowego wpisu
- s: wyszukiwanie według albumu, wykonawcy, tytułu, artysty lub kompozytora

- Ret: Dodaj zaznaczenie do listy odtwarzania
- C-j: Dodaj wybór do listy odtwarzania i odtwórz
- W: Wyszukaj wpis w Wikipedii

Ponieważ jest to bufor Emacsa, standardowe polecenia wyszukiwania będą również działać dla widocznych wpisów. Gdy kursor znajduje się na kategorii, takiej jak nazwa albumu lub kompozytor, doda on wszystkie utwory należące do tej kategorii do listy odtwarzania. Gdy więc kursor znajduje się na albumie, doda on wszystkie utwory.

Gdy kursor znajduje się na pojedynczym utworze, do listy odtwarzania zostanie dodany tylko ten utwór. Przejdź do listy odtwarzania za pomocą emms (C-c w m e), skąd możesz zarządzać odtwarzanymi utworami. Naciśnij

? lista skrótów klawiaturowych, z których niektóre to:

- n / p: następny lub poprzedni utwór
- r: Odtwarzanie losowego utworu.
- D: Usuń wybrany utwór z listy odtwarzania
- C: Wyczyść listę odtwarzania

Konfiguracja EWS konfiguruje również przyciski multimedialne na klawiaturze (odtwarzanie, pauza, następny, poprzedni). Rozszerzenie MPRIS (Media Player Remote Interfacing Specification) zapewnia, że przyciski te działają również poza Emacsem.

Po utworzeniu fajnej playlisty możesz ją zachować na przyszłość w formacie m3u lub pls, aby móc z niej korzystać w Emacs lub innych odtwarzaczach multimedialnych, używając polecenia save-buffer (C-x C-s). Funkcja emms-play-playlist (C-c w m p) pozwala załadować i odtworzyć zapisaną playlistę.

EMMS oferuje wiele zaawansowanych funkcji, które umożliwiają zarządzanie kolekcją plików audio. Inne przydatne funkcje EMMS to:

- emms-play-directory: Dodaj katalog do listy odtwarzania.

- `emms-play-find`: Odtwarza wszystkie pliki w katalogu muzycznym, które spełniają określone kryteria wyszukiwania.

System multimedialny Emacs posiada wiele dodatkowych opcji pozwalających na precyzyjne dostosowanie wrażeń słuchowych. Szczegółowe informacje na temat wielu opcji można znaleźć w instrukcji EMMS dostępnej w przeglądarce Info (C-h R emms).

5.5 Otwieranie plików za pomocą innego oprogramowania

Emacs jest wszechstronnym narzędziem do obsługi zwykłego tekstu, ale nie potrafi wszystkiego. Czasami trzeba otworzyć plik w innym oprogramowaniu, takim jak edytor obrazów lub odtwarzacz wideo.

Pakiet `OpenWith` autorstwa Markusa Triska umożliwia otwieranie plików za pomocą ulubionej przeglądarki plików PDF lub odtwarzacza multimedialnego. Pakiet ten kojarzy typy plików z oprogramowaniem zewnętrznym względem Emacs'a. Aby utworzyć takie powiązanie, należy dostosować zmienną `openwith-associations`. Należy kliknąć przycisk `[Ins]` i dodać następujące trzy parametry:

- Pliki: rozszerzenie pliku (na przykład „pdf” lub „mp4”)
- Program: nazwa programu do otwierania plików (na przykład „evince” lub „mpv”)
- Parametry: domyślnie powinno to być „(file)”, co nakazuje Emacsowi wysłanie nazwy pliku do programu.

Zaletą tego podejścia jest to, że można otwierać pliki w idealnej dla nich aplikacji, korzystając z wygodnego interfejsu Emacs'a, co pozwala uniknąć dodatkowego przełączania się między programami poprzez konieczność otwierania innego oprogramowania.

Ten pakiet oferuje bardziej zaawansowane funkcje kojarzenia plików z zewnętrznym oprogramowaniem, w tym użycie wyrażeń regularnych, które wykraczają poza zakres tej książki.

Teraz, gdy już wiesz, jak czerpać inspirację z Emacs'a, nadszedł czas, aby utrwalić swoje myśli, zapisując je w elektronicznym notatniku. W następnym rozdziale wyjaśniono, jak korzystać z EWS jako osobistego systemu zarządzania wiedzą.

Rozdział 6

Ideacja: rejestrowanie pomysłów i zarządzanie nimi

Czytając literaturę, strony internetowe lub korzystając z innych źródeł informacji, warto robić notatki dotyczące zdobytej wiedzy. Być może pojawi się przypadkowa inspiracja i przyjdzie Ci do głowy jakiś pomysł. Robienie notatek stało się popularnym tematem w ostatnich latach. Rynek pomysłów jest zalewany metodami robienia i przechowywania notatek, z których wszystkie obiecują stworzenie cyfrowego drugiego mózgu, znanego również jako Zettelkasten.

W tym rozdziale wyjaśniono, jak korzystać z pakietów Org i Denote, aby prowadzić elektroniczny notatnik i pielęgnować swój cyfrowy ogród pomysłów. Połączenie tych pakietów umożliwia zbieranie notatek, prowadzenie osobistego pamiętnika lub dziennika, pisanie recenzji literatury, zapisywanie wyników laboratoryjnych w notatniku lub inne możliwe zastosowania. Podejście opisane w tym rozdziale jest niezależne od jakiegokolwiek formalnej metody. Emacs jest na tyle elastyczny, że pozwala na wdrożenie niemal każdego preferowanego sposobu pracy.

W pierwszej części tego rozdziału wyjaśniono, jak używać Org jako narzędzia do płynnego sporządzania notatek, wykorzystując pojedynczy plik do zapisywania swoich przemyśleń. Pozostała część rozdziału opisuje ekosystem pakietu Denote, opracowany przez Protesilaosa (Prota) Stavrou i innych.

Pakiet ten zapewnia elastyczność w zarządzaniu obszerną kolekcją notatek tekstowych i innych typów plików, umożliwiając stworzenie połączonego drugiego mózgu.

6.1 Myśli dotyczące notatek

Przed wynalezieniem pisma nasi przodkowie dzielili się wiedzą poprzez zapamiętywanie. Ludzie z tych prehistorycznych kultur potrafili recytować z pamięci tysiące wersów wielkich dzieł, takich jak Iliada i Mahabharata oraz innych eposów, wspomagając się pieśniami i rymowankami (Kelly, 2015). Rymowane słowa są znacznie łatwiejsze do zapamiętania niż zwykła proza. Potrafimy bezbłędnie zapamiętać godziny tekstów piosenek, ale nie potrafimy zapamiętać listy zakupów. Kiedy pisanie stało się powszechne, grecki filozof Sokrates ubolewał w swoim dialogu z *Platosem*, że osłabia ono naszą pamięć. Najnowsze badania pokazują, że argument Sokrates był zgodny z faktami (Kahn, 1997).

Jednak pisanie uwolniło nasze umysły od bycia magazynem wiedzy faktograficznej i przekształciło je w kreatywne maszyny. Rozwój ludzkości przyspieszył, gdy pisanie pozwoliło ludziom uwolnić umysły od faktów i wykorzystać tę moc umysłową do tworzenia nowych idei. Guru produktywności David Allen ujął to zwięźle, pisząc: „Umysł służy do tworzenia pomysłów, a nie do ich przechowywania” (Allen, 2005). Filozofowie odnoszą się do intuicji Allena jako do *rozszerzonej miedztłęsis*, co sugeruje, że umysł nie znajduje się wyłącznie w ciele, ale rozciąga się na świat fizyczny (Clark & Chalmers, 1998), co prowadzi do koncepcji drugiego mózgu.

Robienie notatek w celu przedłużenia kruchości ludzkiej pamięci jest tak stare jak sama sztuka pisania. Od lat 70. XX wieku ludzkość przeszła od zapisywania słów na papierze do przechowywania ich w formie elektronicznej. W ostatnich latach potrzeba prowadzenia notatek dała początek branży blogów, książek, kanałów YouTube i aplikacji do robienia notatek, które pomagają ludziom uporządkować myśli i życie. Niezależnie od tego, czy chodzi o *Zettelkasten* Luhmanna, *Bullet Journal* Carrola, czy *Seeoed Bvaie* forte'a, wszystkie one twierdzą, że rozwiązują problemy związane z zarządzaniem wiedzą osobistą, stosując określoną metodę (forte, 2022; Kadavy, 2021; Ryder, 2021).

Te sztywne procedury nieuchronnie prowadzą do niepowodzeń, ponieważ każdy ma inne potrzeby.

Podczas kreatywnego sporządzania notatek formalny proces ma niewielkie znaczenie. W miarę powiększania się zbioru notatek Twoja metodologia powstaje w sposób naturalny. Zamiast więc martwić się o metody promowane w Internecie, zacznij pisać i twórz proces w miarę ewolucji swoich potrzeb.

Najlepsza metoda robienia notatek to ta, którą sam wymyślisz i która z czasem staje się coraz bardziej złożona. Elektroniczny drugi mózg nie gwarantuje automatycznie sukcesu. To Twój prawdziwy mózg powinien dyktować warunki drugiemu, a nie na odwrót (Stavrou, 2024).

W ciągu trzech dekad pracy z papierowymi i cyfrowymi dziennikami nauczyłem się, że struktura jest tylko jednym z kluczowych aspektów. To, co naprawdę ma znaczenie, to autentyczność i oryginalność zapisanych myśli. Nawet Luhmann, twórca metody Zettelkasten, postrzegał swój system jako „szambo na pomysły” (ZK II Zettel y/8a2 niklas-luhmann-archiv.de). Korzystanie z programu Emacs do robienia notatek lub jakiegokolwiek innego edytora tekstu ma swoje ograniczenia. Mój osobisty proces tworzenia pomysłów zaczyna się od fizycznego notatnika. Może to być dziwne wyznanie dla kogoś, kto deklaruje podziw dla elektronicznych zalet programu Emacs. Fizyczne notatniki mają pewne zalety w procesie twórczym. Po pierwsze, można z nich korzystać w dowolnym miejscu bez baterii i działają one w pełnym słońcu. Jedynym wyjątkiem może być prysznic, który paradoksalnie jest miejscem, gdzie powstają nasze najlepsze pomysły. Pisanie odręczne w notatniku jest również procesem wolniejszym niż korzystanie z klawiatury, co może wydawać się wadą, ale używanie pióra zmusza do głębszego myślenia. Pisanie na papierze zwiększa kreatywność, ponieważ uruchamia głębsze ścieżki neuronowe niż pisanie elektroniczne (Mueller & Oppenheimer, 2014; Umejima, Ibaraki, Yamazaki, & Sakai, 2021). Pisanie na papierze ułatwia również łączenie grafiki z tekstem. Rysowanie i szkicowanie to artystyczne sposoby konceptualizacji wiedzy, czego nie potrafi edytor tekstu, taki jak Emacs. zapewnić.

Kiedy pomysły z mojego notatnika zaczynają się rozwijać, niektóre z nich trafiają do mojego systemu archiwizacji Emacs. Moja praca obejmuje również skanowanie szkiców z notatnika, zdjęć, filmów i plików elektronicznych z internetu lub przesłanych mi pocztą elektroniczną. Wszystkie pliki są przechowywane w uporządkowanym systemie, co ułatwia wyszukiwanie informacji i rozwijanie nowych pomysłów. Ten sposób pracy łączy elastyczność pracy na papierze z możliwościami, jakie daje informacja elektroniczna.

W praktyce można wyróżnić dwie kategorie notatek: notatki tymczasowe i notatki stałe (Ahrens, 2017). Notatki ulotne zapisuje się na serwetce lub na grzbiecie dłoni. Większość notatek ulotnych ma krótki okres przydatności, ale niektóre z nich uzyskują status notatek trwałych. Jak sama nazwa wskazuje, notatka trwała to informacja, którą chcesz zachować na zawsze. Twoje notatki trwałe tworzą Twoją osobistą wiki, drugi mózg, Zettelkasten, cyfrowy ogród lub jakkolwiek inny neologizm, który preferujesz.

6.2 Notatki ulotne

ulotne notatki pozwalają uchwycić nieoczekiwane pomysły, które pojawiają się w głowie w nieprzewidywalnych momentach. Nagła intuicja, cytat z filmu, przypomnienie o liście rzeczy do zrobienia lub cokolwiek innego. Te ulotne notatki są tymczasowymi miejscami przechowywania myśli. Mogą to być pomysły na przyszły projekt, zadania do wykonania lub coś interesującego, do czego warto wrócić później. System bezproblemowego zapisywania, taki jak sprawdzony papierowy notatnik, aplikacja w telefonie, a nawet tył dłoni, zapobiega zniknięciu tych pomysłów.

Podczas korzystania z Emacsa również pojawia się potrzeba sporządzania szybkich notatek. Wyobraź sobie, że piszesz książkę i nagle przypomina Ci się, że musisz kupić mleko. Tryb Org capture pozwala Ci zapisać tę przelotną myśl za pomocą kilku naciśnieć klawiszy, zachowując ją do późniejszego przejrzania bez rozpraszania uwagi.

Rejestrowanie ulotnych notatek za pomocą funkcji przechwytywania Org jest bardzo proste. Naciśnij C-c c (org-capture), a pojawi się ekran wyboru. Wybierz „fleeting Note” za pomocą klawisza f, wpisz swoje myśli w wyskakującym buforze i naciśnij C-c C-c, aby zapisać notatkę pod nagłówkiem „Notes” w skrzynce odbiorczej jako pozycję listy. System przechwytywania dodaje kolejne ulotne notatki poniżej poprzednich. Jeśli uznasz, że nie warto przechowywać tej myśli, naciśnij C-c C-k, aby anulować wpis.

Po zakończeniu Emacs powraca do miejsca, w którym przerwałeś pracę, i możesz spokojnie kontynuować swoją pracę przy minimalnych zakłóceniach. Menu przechwytywania zawiera również opcję dodania pozycji do listy rzeczy do zrobienia, przechowywanej w tym samym pliku, ale pod innym nagłówkiem. Rozdział y omawia zarządzanie projektami i listami działań.

Zmienna `org-default-notes-file` określa nazwę i lokalizację skrzynki odbiorczej. Domyślnie zmienna ta ma wartość `~/notes`. Kropka oznacza, że plik ten jest ukryty. Można dostosować tę zmienną, aby ustawić preferowaną nazwę pliku skrzynki odbiorczej. Plik jest tworzony automatycznie przy pierwszym użyciu mechanizmu przechwytywania. Oczywiście można bezpośrednio dodawać inne informacje do tego pliku poza systemem przechwytywania.

W miarę tworzenia coraz to nowych, ulotnych notatek, skrzynka odbiorcza stopniowo zapełnia się przypadkowymi przemyśleniami. Dobrym nawykiem pozwalającym utrzymać skrzynkę odbiorczą w stanie jak największej pustki jest przeprowadzanie cotygodniowego przeglądu. Przegląd ten polega na przekształcaniu obiecujących myśli w trwałe notatki lub usuwaniu ich po upływie terminu ważności. Idealnie byłoby, gdyby zawartość skrzynki odbiorczej dążyła do zera, jak omówiono w rozdziale y.

Możesz użyć argumentu uniwersalnego z poleceniem `org-capture` (`C-u C-c c`), aby przejść do pliku wybranego szablonu. Użycie podwójnego argumentu uniwersalnego powoduje przejście do ostatnio przechwyconego elementu (`C-u C-u C-c c`). Dodanie informacji za pomocą systemu przechwytywania powoduje również utworzenie zakładki, dzięki czemu można znaleźć najnowsze notatki za pomocą `C-x r b` (`bookmark-jump`) i wybrać opcję `org-capture-last-stored`.

Funkcja przechwytywania to wszechstronny system, który umożliwia tworzenie szablonów dla różnych notatek. EWS definiuje tylko dwa rodzaje notatek, ale można dostosować ten system, aby dodać własne szablony przechwytywania. Można na przykład utworzyć osobny wpis dla listy zakupów i zsynchronizować ten plik z telefonem. Org otwiera ekran dostosowywania zmiennej `org-capture-templates` za pomocą `C-c c C`, który zawiera obszerny plik pomocy opisujący niezliczone opcje przechwytywania. Podręcznik Org (`C-h R org <Ret> g capture`) szczegółowo omawia tworzenie szablonów przechwytywania.

6.ç Permament notes

Notatki stale tworzą zewnętrzny elektroniczny magazyn informacji, który można uporządkować i przeszukiwać w celu uzyskania nowych spostrzeżeń. Nie przejmuj się zbyt wiele wyborem metody, gdy zaczynasz gromadzić notatki elektroniczne. Możesz zacząć od jednego pliku i po prostu pisać. Kluczem do tworzenia dobrych notatek jest nie martwienie się o drugi mózg, ponieważ pierwszy

mózg jest znacznie ważniejszy w procesie twórczym (Stavrou, 2024). Drugi mózg wyłoni się organicznie z pierwotnej zupy pomysłów, jaką jest system sporządzania notatek.

Istnieją dwie metody tworzenia trwałych notatek. Można albo umieścić wszystkie notatki w jednym pliku Org mode i wykorzystać jego wewnętrzne możliwości do zarządzania informacjami, albo stworzyć zbiór notatek połączonych hiperłączami, zgodnie z popularną metodą Zettelkasten.

W poniższej sekcji wyjaśniono, jak używać pojedynczego pliku do sporządzania notatek. W następnej sekcji przedstawiono pakiet Denote, potężne narzędzie do zarządzania obszerną kolekcją notatek, w tym plikami binarnymi, takimi jak zdjęcia i pliki PDF.

Plik Siegle Ovg

Aby rozpocząć, utwórz plik Org, nadaj mu odpowiedni tytuł i zacznij pisać. Używaj opisowych nagłówków dla każdej notatki. Możesz również grupować notatki, używając nagłówków pierwszego poziomu jako kategorii, a niższych poziomów jako tytułów notatek.

Aby dodać znacznik czasu rejestrujący moment sporządzenia notatki, należy użyć funkcji org-time-stamp. Wywołanie tej funkcji za pomocą C-c . (klawisz Control i kropka) powoduje dodanie daty, a wraz z argumentem uniwersalnym również godziny (C-u C-c .). Notatka z kategorii filozofia mogłaby wyglądać następująco:

```
#+title: Notatki
```

```
* Filozofia
```

```
** Sokrates przeciwko pisaniu
```

```
<2024-04-20 sobota>
```

```
W dialogu „Fajdros” ...
```

Org pomaga zarządzać obszerną kolekcją notatek w jednym pliku. Aby wyświetlić spis treści pliku po jego otwarciu, dodaj #+startup: content do przedniej części dokumentu. Dzięki temu słowu kluczowemu Org wyświetla nagłówki tylko przy pierwszym otwarciu pliku. Możliwość zwijania i rozwijania nagłówków za pomocą S-Tab> pozwala skupić się na tym, co najważniejsze.

Inne opcje startowe to overview, która pokazuje tylko pierwszy poziom, oraz showall, która rozwija cały dokument. Opcją domyślną jest showeverything, która nie ukrywa żadnej części dokumentu.

Inną metodą skupienia uwagi w dużych plikach jest zawężenie bufora, aby wyświetlać tylko sekcję, nad którą aktualnie pracujesz. Prefiks C-x n przenosi Cię do funkcji zawężania. Funkcja org-narrow-to-subtree (C-x n s) zawęży bieżący bufor, aby wyświetlać tylko zawartość poddrzewa, w którym znajduje się kursor. Pozostały tekst nie jest usuwany, a jedynie ukryty. Aby przywrócić pełny bufor, należy użyć polecenia widen, przypisanego do kombinacji klawiszy C-x n w. Funkcja zawężania posiada kilka innych opcji, które można sprawdzić w menu podręcznym po naciśnięciu klawisza prefiksowego.

Kolejną metodą skupienia się na istotnych fragmentach dokumentu jest utworzenie drzewa rzadkiego za pomocą funkcji org-occur, wywoływanej skrótem C-c / /. Drzewa rzadkie oferują filtrowane widoki oparte na kryteriach wyszukiwania, podkreślając istotny tekst i ukrywając nieistotną treść. Po wprowadzeniu kryterium wyszukiwania Org podświetla żądane słowa i pokazuje tylko sekcje, w których występuje wyszukiwane hasło. Dwa skróty pozwalają przeskakiwać między wynikami: M-g n przeskakuje do następnego wyniku, a M-g p do poprzedniego. Użycie dowolnego polecenia edycji lub naciśnięcie C-c C-c powoduje zakończenie wyszukiwania. Główna różnica między drzewem rzadkim a zwykłą funkcją wyszukiwania (sekcja 3.8) polega na tym, że drzewo rzadkie zwija dokument, aby pokazać tylko te części, w których występuje wyszukiwane hasło.

Dodawanie notatek do kategorii poprzez strukturyzowanie nagłówków jest pomocne, ale ma swoje ograniczenia, ponieważ notatka może należeć tylko do jednego podkatalogu. Org może również dodawać tagi do każdego nagłówka, aby powiązać ze sobą różne pomysły. Tag jest etykietą nagłówka służącą do kategoryzowania powiązanych nagłówków. Tagi pojawiają się po tekście nagłówka, umieszczone między dwukropkami. Tagi są właściwościami dziedziczonymi, co oznacza, że każdy tag na poziomie nagłówka pierwszego poziomu należy również do odpowiednich podnagłówków. W poniższym przykładzie wszystkie podnagłówki pod nagłówkiem „Filozofia” dziedziczą tag

:filozofia: tag. Wszystkie podtytuły pod notatką dotyczącą Sokratesa również odziedziczają tagi :filozofia: i :pisarstwo:. Nagłówek może mieć wiele tagów, co pozwala na stworzenie szczegółowej klasyfikacji pomysłów.

```
#+title: Notatki
* Filozofia                               :filozofia:
** Sokrates przeciwko pisaniu            :pisarstwo:sokrates:
    <2024-04-20 sobota>
    W dialogu „Fajdros”...
```

Dodajesz tag do notatki za pomocą C-c C-q (org-set-tags-command). Wpisz nazwę nowego tagu w minibufferze. Wszystkie tagi użyte już w dokumencie są wyświetlane na liście uzupełnień minibuffera. Możesz również ustawić bibliotekę dla każdego pliku, dodając coś takiego na początku pliku Org: #+tags: philosophy(p) writing(w). Litery w nawiasach stają się skrótem w menu minibuffera, umożliwiającym szybki wybór. Aby utworzyć nowy tag, wpisz dowolny tekst w minibufferze. Gdy masz już plik z tagami, możesz używać ich do wyszukiwania notatek według kategorii, korzystając z funkcji drzewa rozproszonego. Aby wybrać jeden lub więcej tagów dla drzewa rozproszonego, użyj org-match-sparse-tree (C-c \). Ta funkcja zwija cały dokument i podświetla segmenty, w których występują wybrane tagi.

Poruszanie się po dużych plikach Org może być uciążliwe. Polecenie org-goto ułatwia to zadanie. Po naciśnięciu C-c C-j Org wyświetla wszystkie nagłówki w menu uzupełniania minibuffera, z którego można wybrać miejsce docelowe. Pakiet Consult autorstwa Daniela Mendlera zawiera wygodną funkcję do poruszania się po dużych plikach Org. Funkcja consult-org-heading (C-c w h) wyświetla listę wszystkich nagłówków w bieżącym pliku Org w minibufferze, z którego można przejść do żądanej lokalizacji. Pakiet Consult oferuje szeroki zakres poleceń wyszukiwania i nawigacji, które usprawniają korzystanie z Emacs'a.

Nagłówki można strukturyzować za pomocą klawiszy Alt i strzałek, jak wyjaśniono w sekcji 4.5. Wygodnym narzędziem do zarządzania dużymi plikami jest funkcja org-refile, przypisana do skrótu C-c C-w. Polecenie to pozwala na łatwe przenoszenie sekcji w obrębie dokumentu. Po wywołaniu tej funkcji w minibufferze pojawia się lista nazw rozdziałów. Poddzewo, w którym znajduje się obecnie kursor, zostanie przeniesione do wybranego rozdziału. Aby przejść do odpowiedniego wpisu

po zmianie lokalizacji, użyj skrótu C-u C-u C-c C-w (dwa argumenty uniwersalne przed poleceniem).

Na koniec możesz chcieć utworzyć linki między notatkami w pliku. Linki do plików omówiliśmy już w sekcji 4.5, ale możemy również utworzyć link do nagłówka w pliku Org. Najłatwiejszym sposobem jest utworzenie linku wewnętrznego za pomocą C-c l, wpisanie nazwy nagłówka bez gwiazdek i dodanie opisu. Link wygląda teraz mniej więcej tak:

```
[[Nazwa nagłówka][Opis]]
```

Problem z tym podejściem polega na tym, że nazwa nagłówka może ulec zmianie lub może ją pan/pani błędnie wpisać. Podążając za linkiem do nieistniejącego celu, tryb Org nie wyświetla błędu, ale pyta, czy chce pan/pani utworzyć nowy nagłówek. Lepszym podejściem do tworzenia linków jest nadanie nagłówkowi unikalnego identyfikatora.

Aby wstawić link między notatkami w dokumencie zawierającym jedną notatkę, przesunij kursor na nagłówek, do którego chcesz utworzyć link, i naciśnij C-c l(org-store-link). Ta funkcja tworzy szufladę pod nagłówkiem. Szuflada składa się ze składanego tekstu, w którym można przechowywać metadane dotyczące nagłówka. Szuflady są pomocne w wielu zadaniach i zostały omówione bardziej szczegółowo w rozdziale 7. Szuflada może wyglądać mniej więcej tak:

```
:WŁAŚCIWOŚCI:
```

```
:ID:          d454979b-2d40-4f95-9f85-f5d9314c28d7
```

```
:KONIEC:
```

Losowy ciąg liter i cyfr to uniwersalny identyfikator (UUI), który tworzy losowy identyfikator. Prawdopodobieństwo wystąpienia zduplikowanego identyfikatora jest tak astronomicznie małe, że możemy uznać go za unikalny. Link do tego identyfikatora jest teraz przechowywany w pamięci i można go wstawić w innym miejscu za pomocą org-store-link. Link do identyfikatora wygląda tak:

```
[[id:d454979b-2d40-4f95-9f85-f5d9314c28d7][Przykład]]
```

Używanie jednego dużego pliku do przechowywania notatek to świetny sposób na rozpoczęcie przenoszenia swoich myśli do Emacsa. Jednak z czasem plik ten może stać się nieporęczny. Jeśli osiągniesz wysoką produktywność, duży plik może spowolnić działanie systemu. W następnej sekcji pokazano, jak używać pakietu Denote do tworzenia zbioru połączonych ze sobą notatek.

6.4 Pisanie notatek za pomocą Denote

Korzystanie z jednego pliku to dobry sposób na rozpoczęcie przygody, ale gdy pliki te osiągną ogromne rozmiary, stają się nieporęczne. Dlatego większość systemów do robienia notatek wykorzystuje oddzielne pliki do tworzenia sieci pomysłów. Użytkownicy Emacsa opracowali szereg pakietów do pisania i zarządzania zbiorami notatek. EWS wykorzystuje pakiet Denote. Pakiet ten nie narzuca żadnej konkretnej metodologii ani sposobu pracy. Może przetwarzać zarówno notatki zapisane w trzech formatach tekstowych, jak i pliki binarne, takie jak zdjęcia, pliki PDF lub inne typy plików, które chcesz przechowywać w swoim archiwum cyfrowym.

Pakiet Denote kategoryzuje pliki za pomocą słów kluczowych. Istnieje również opcja dodania podpisu, który może wyznaczać porządek semantyczny. Notatki mogą być również ze sobą powiązane, tworząc sieć myśli. Dzięki tym trzem mechanizmom można używać Denote do tworzenia organicznego ogrodu cyfrowego lub wdrażania formalnych systemów, takich jak Zettelkasten, Johnny.Decimal lub metoda PARA, albo pracować zgodnie z własnymi preferencjami.

Siłą napędową pakietu Denote jest konwencja nazewnictwa plików. Podejście to polega na osadzaniu metadanych w nazwie pliku, co eliminuje potrzebę korzystania z bazy danych lub innych zewnętrznych zależności do poruszania się po gąszczu notatek. Konwencja nazewnictwa Denote składa się z pięciu części (domyślnie wszystkie pisane małymi literami), z których wymagane są tylko identyfikator i rozszerzenie pliku. Konwencja nazewnictwa plików w Denote ogranicza swobodę w nadawaniu nazw plikom. Jednak ograniczenia te zapewniają niesamowitą moc, wprowadzając przewidywalność i jednolitość, co ułatwia wyszukiwanie notatek. Przykładem w pełni sformatowanego pliku Denote jest.

20210509T082300==9=a=12--duck-rabbit-illusion perception.org

1. Unikalny identyfikator (ID) w formacie czasu ISO 8501.
2. Podpis (małe litery i cyfry), zaczynający się od podwójnego znaku równości.
3. Tytuł oddzielony myślnikami (kebab-case), zaczynający się od podwójnego myślnika.
4. Słowa kluczowe oddzielone znakiem podkreślenia (snake_case), zaczynające się od podwójnego znaku podkreślenia.
5. Rozszerzenie nazwy pliku.

Sygnatura czasowa porządkuje nasze notatki chronologicznie i tworzy unikalny i niezmienny identyfikator, którego Denote używa do łączenia plików. Sygnatura pozwala porządkować notatki tak samo, jak system dziesiętny Deweya porządkuje książki na półkach w fizycznej bibliotece. Słowa kluczowe lub tagi plików grupują notatki, które mają wspólny temat. Sygnatura, tytuł i tagi są elastyczne i mogą się zmieniać w czasie. Sygnatura czasowa powinna zawsze pozostawać taka sama, aby zachować integralność linków.

Aby zachować integralność systemu, podpis Denote może zawierać wyłącznie litery, cyfry i znaki równości. Tytuł zawiera wyłącznie litery, cyfry i myślniki (kebab-case). Słowa kluczowe zaczynają się od znaku podkreślenia i mogą zawierać wyłącznie litery i cyfry (snake_{case}). Denote oczyszcza (sluggifies) nazwy plików, aby zapewnić zgodność z konwencją.

Denote zapisuje nowe notatki w folderze wskazanym przez zmienną `denote-directory`, której domyślną wartością jest `~/Documents/notes`. Możesz dostosować tę zmienną do swoich potrzeb.

Denote może przechowywać notatki w podkatalogach w katalogu `denote`, ale nie ma takiej potrzeby. Podczas używania podkatalogów do kategoryzowania plików część metadanych dla tego pliku zmienia się po przeniesieniu pliku w inne miejsce. Nowoczesne systemy operacyjne mogą bez problemu zarządzać dziesiątkami tysięcy plików w jednym katalogu, więc nie ma potrzeby korzystania z podkatalogów. Zamiast podkatalogów można używać tagów plików, które ułatwiają przeglądanie plików należących logicznie do tej samej grupy. Tagi plików są bardziej elastyczne.

niż podkatalogi, ponieważ każdy plik może mieć wiele tagów, ale może znajdować się tylko w jednym katalogu.

Utwórz notatki

Funkcja Denote jest dostępna pod prefiksem C-c w d EWS. Polecenie Denote, które aktywujesz za pomocą C-c w d n, tworzy nową notatkę jako plik Org. Najpierw prosi o podanie tytułu, a następnie odpowiednich słów kluczowych. Możesz wybrać słowo kluczowe z listy uzupełnień istniejących notatek w minibufferze za pomocą klawisza Tab lub wprowadzić nowe jako dowolny tekst, oddzielając je przecinkami. Sygnatura czasowa jest generowana automatycznie na podstawie daty i godziny utworzenia notatki. Możesz również aktywować to polecenie za pomocą systemu przechwytywania Org i wybrać opcję „Permanent Note” (C-c c p).

Podczas tworzenia nowej notatki otwiera się ona najpierw jako niezapisany bufor. Aby ją zapisać na dysku, należy użyć kombinacji klawiszy C-x C-s. Tworzenie stałej notatki za pomocą mechanizmu przechwytywania Org powoduje zapisanie notatki po zamknięciu wyskakującego okienka przechwytywania za pomocą C-c C-c. Niektóre funkcje mogą nie działać, jeśli notatka nie została zapisana na dysku, więc jeśli pojawi się ostrzeżenie „Buffer not visiting a Denote file” (Bufor nie odwiedza pliku Denote), może być konieczne zapisanie bufora, aby Denote go rozpoznało.

Domyślna konfiguracja EWS nie wymaga podpisu ani podkatalogu dla nowych notatek. Możesz dostosować zmienną denote-prompts, aby zdefiniować domyślny sposób generowania i zmiany nazw plików przez Denote, zaznaczając elementy, które chcesz uwzględnić podczas tworzenia nowej notatki.

Data i identyfikator są również częścią nagłówka pliku. Słowa kluczowe stają się tagami plików, które są podobne do tagów, które widzieliśmy w poprzedniej sekcji, ale mają zastosowanie do całego pliku. Teraz wypełnij bufor odpowiednią treścią i zapisz go na dysku. Czołówka notatki w powyższym przykładzie wyglądałaby następująco:

```
#+tytuł:      Jednoczesny kontrast
#+data:      [2021-05-09 niedziela
08:23]
#+filetags:   :kolor:iluzje:
#+identyfikator: 20210509T082300
```

Ten schemat działania dotyczy ogólnych notatek. Jednak nie wszystkie notatki są tworzone w ten sam sposób. Odpowiedni schemat działania w ramach Denote zależy od celu notatki. Ogólnie rzecz biorąc, można wyróżnić cztery rodzaje:

1. *Jouveal eetvies*: Doświadczenia związane z konkretnym czasem.
2. *Litevatuve eotes*: Notatki dotyczące publikacji.
3. *Attatetmeets*: Notatki tylko do odczytu, takie jak zdjęcia lub pliki PDF.
4. *Meta eotes*: notatki, które odsyłają do wszystkich notatek spełniających kryteria wyszukiwania.

Keepieg a jouveal ov diavy

Możesz używać Denote do osobistych refleksji, tworzenia dziennika lub dziennika laboratoryjnego, dodawania notatek ze spotkań lub zapisywania wszelkich innych notatek związanych z wydarzeniem.

Pisanie dziennika w aplikacji Denote jest łatwe, ponieważ identyfikator każdej notatki wskazuje datę i godzinę jej utworzenia. Dodanie standardowego tagu, takiego jak `_dziennik`, ułatwia odróżnienie wpisów w dzienniku od innych notatek lub innych elementów, które mają znaczenie w Twoim języku ojczystym.

Jeśli tworzysz notatkę dotyczącą wpisu z przeszłości, użyj funkcji `denote-date` (C-c w d d). Datę wpisujesz w formacie rok-miesiąc-dzień (ISO 8501), np. 2023-09-06. Opcjonalnie można dodać konkretną godzinę w formacie 24-godzinnym, na przykład 2023-09-6 20:30. Jeśli nie podano daty i/lub godziny, funkcja Denote używa bieżącej daty lub godziny.

Pakiet Denote-Journal zapewnia dodatkowe specjalistyczne funkcje do prowadzenia dziennika lub pamiętnika. Można ustawić standardowe słowo kluczowe, aby nie trzeba było go wybierać za każdym razem podczas tworzenia nowego wpisu. Pakiet ten umożliwia również dostęp do notatek poprzez widok kalendarza. Instrukcję obsługi można przeczytać za pomocą polecenia C-h R `denote-journal`.

Literature notes

Notatka literacka lub bibliograficzna zawiera streszczenie lub interpretację książki, artykułu z czasopisma lub innego opublikowanego formatu. Notatka literacka jest specjalną kategorią notatek stałych, które odsyłają do jednej lub kilku publikacji.

Pakiet Citar-Denote integruje bibliografię Emacsa i Citar z systemem notatek Denote. Pakiet ten zapewnia rozszerzoną funkcjonalność tworzenia notatek literaturowych i zarządzania nimi. Aby dowiedzieć się, jak tworzyć bibliografię i korzystać z Citar, zapoznaj się z rozdziałem 5.

Citar-Denote umożliwia tworzenie relacji wiele-do-wielu między notatkami a wpisami w plikach BibTeX, zapewniając kompleksowe rozwiązanie do dokumentowania notatek literackich. Oznacza to, że można dodać wiele notatek do jednego wpisu bibliograficznego lub jedną notatkę do więcej niż jednego dzieła literackiego. Można napisać notatkę dotyczącą każdego rozdziału książki lub utworzyć jedną notatkę literacką dla zbioru artykułów z czasopism, w zależności od tego, która metoda bardziej odpowiada danemu procesowi pracy.

Notatki bibliograficzne są zwykłymi plikami Denote, ale zawierają dodatkowe metadane, które łączą plik z jednym lub kilkoma wpisami w bibliografii. Citar-Denote łączy notatkę z wpisem w bibliografii, używając klucza cytowania jako odniesienia w przedniej części dokumentu. Każda notatka bibliograficzna jest również oznaczona tagiem pliku `_bib`, aby zmniejszyć liczbę plików, które system musi śledzić. Część wstępna notatki bibliograficznej może wyglądać mniej więcej tak: (zwróć uwagę na tag `_bib` i linię odniesienia):

```
#+tytuł:      Marcuse: Esej o wyzwoleniu #+data:
              [2022-11-12 sobota 19:23]
#+tagi pliku:  :bib:kultura:marketing:filozofia:
#+identyfikator: 20221112T192310
#+odniesienie: marcuse_1969_essay
```

Otwórz interfejs Citar za pomocą `C-c w b c` (`citar-create-note`), aby utworzyć nową notatkę. Wybierz wpis, do którego chcesz dodać notatkę, naciśnij `Enter` i postępuj zgodnie z instrukcjami. Jeśli dla tego wpisu istnieje już notatka, możesz utworzyć dodatkowe notatki lub otworzyć istniejącą.

Po zebraniu notatek bibliograficznych będziesz chciał uzyskać do nich dostęp i je modyfikować. Możesz uzyskać dostęp do załączników, linków i innych notatek związanych z odniesieniami z poziomu menu Citar za pomocą C-c w b o (citar-open). Wpisy z notatką są oznaczone literą N w trzeciej kolumnie. Z tego menu można również tworzyć dodatkowe notatki.

Aby wyświetlić tylko te wpisy z notatką, rozpocznij wyszukiwanie od znaku :n. Alternatywnie użyj citar-denote-open-note (C-c w b n), aby otworzyć menu Citar zawierające tylko wpisy z co najmniej jedną powiązaną notatką. Ponadto funkcja citar-open-note wyświetla w minibufferze listę nazw plików wszystkich notatek literaturowych.

Funkcja citar-denote-add-citekey (C-c w b k) dodaje klucze cytowań lub konwertuje istniejący plik Denote na notatkę bibliograficzną. Podczas konwersji zwykłego pliku De-note funkcja dodaje słowo kluczowe bib do przedniej części dokumentu i odpowiednio zmienia nazwę pliku. Funkcja ta otwiera menu wyboru Citar i dodaje wybrane klucze cytowań do przedniej części dokumentu. Możesz usunąć odniesienia do cytatów z notatki za pomocą skrótu C-c w b K (citar-denote-remove-citekey). Jeśli odnosisz się do więcej niż jednej publikacji, najpierw wybierz niepotrzebny element w minibufferze. Jeśli notatka zawiera tylko jedno odniesienie, słowo kluczowe bibliography jest usuwane, a plik jest zmieniany, konwertując go na ogólną notatkę stałą.

W notatce bibliograficznej dostępnych jest kilka funkcji służących do zarządzania bieżącym buforem. Funkcja citar-denote-dwim (C-c w b d) zapewnia dostęp do menu Citar dla literatury cytowanej w tej notatce, z którego można otworzyć załączniki, inne notatki i linki. Jeśli notatka zawiera więcej niż jeden element referencyjny, należy najpierw wybrać odpowiedni element.

Jaki jest sens tworzenia bibliografii bez cytowania lub wykorzystania każdego pozycji przynajmniej raz w notatce bibliograficznej? Funkcja citar-denote-nocite (C-c w b x) otwiera menu Citar i wyświetla wszystkie pozycje w bibliografii, które nie zostały ani cytowane, ani przywołane. Stamtąd można utworzyć nową notatkę bibliograficzną, przejść do linku lub przeczytać plik. Funkcja ta może służyć jako lista kontrolna literatury, której jeszcze nie przeczytałeś lub nie przejrzałeś.

Aby dowiedzieć się więcej o funkcjonalności pakietu Citar-Denote, przeczytaj instrukcję za pomocą C-h R citar-denote.

Attache meets

Twój cyfrowy ogród notatek może być czymś więcej niż tylko tekstem. Dzięki Denote możesz zarządzać swoimi zdjęciami i przechowywać archiwum plików PDF, takich jak wyciągi bankowe, certyfikaty kursów lub skany archiwum papierowego. Rozszerzenie Denote o załączniki przekształca listę notatek w kompleksowy system zarządzania wiedzą osobistą, wyposażony w intuicyjne heurystyki służące do wyszukiwania i łączenia dokumentów z notatkami.

Istnieje wiele zastosowań rozszerzenia programu Denote na pliki binarne. Zapisuję swoje zdjęcia i filmy w formacie plików Denote. Przechowuję również pliki PDF, takie jak zeskanowane dokumenty papierowe lub pliki otrzymane pocztą elektroniczną, w tym faktury.

Wykorzystanie przez Denote nazw plików do przechowywania metadanych pozwala zarządzać plikami innymi niż trzy typy plików tekstowych, które Denote może generować (tekst zwykły, Markdown lub Org). Załącznik to plik o zgodnej nazwie, z wyjątkiem plików tworzonych przez Denote. Denote rozpoznaje każdy plik przechowywany w katalogu Denote, który jest zgodny z konwencją nazewnictwa plików.

Pierwszym krokiem podczas rejestrowania załącznika w Denote jest upewnienie się, że ma on zgodną nazwę. Możesz ręcznie zmienić nazwę pliku po otwarciu go za pomocą polecenia `denote-rename-file` (C-c w d r). Ta funkcja używa nazwy pliku jako domyślnego tytułu, który możesz modyfikować i dodawać odpowiednie słowa kluczowe w razie potrzeby. Ostatni znacznik czasu modyfikacji pliku będzie służył jako jego identyfikator. Jednak data utworzenia w systemie plików nie zawsze jest rzeczywistą datą utworzenia. Podczas pracy z załącznikami istnieją trzy opcje prawidłowego znacznika czasu, czyli data i godzina, kiedy:

- Utworzono dokument cyfrowy
- Plik elektroniczny został utworzony (data pierwszego utworzenia)
- Plik elektroniczny został utworzony w systemie plików (oznacz domyślnie)

Pierwszy scenariusz dotyczy głównie dokumentów historycznych. Przez lata stopniowo digitalizowałem swoje archiwa papierowe. Najwcześniejszy znacznik czasu w mojej bibliotece Denote to 13700623T120000, skan średniowiecznej umowy hipotecznej dotyczącej mojego domu rodzinnego w Holandii. Pierwotna data utworzenia dokumentu (kiedy został zeskanowany) to 2021 r., a data w moim systemie plików to 2023 r. Funkcja zmiany nazwy w Denote używałaby daty z systemu plików, co nie jest idealnym rozwiązaniem. Dokument ten wymaga ręcznego wprowadzenia znacznika czasu, który umieszcza dokument w odległej przeszłości.

Drugi scenariusz dotyczy głównie zdjęć. Znacznik czasu w systemie plików może różnić się od czasu wykonania zdjęcia, dlatego musimy znać dokładny czas wykonania zdjęcia. W przypadku nowych zdjęć datę utworzenia można uzyskać z metadanych pliku. Istnieje kilka narzędzi, takich jak ExifTool, które umożliwiają wyodrębnienie metadanych ze zdjęć i plików PDF.

Metadane

Po napisaniu wielu notatek możesz chcieć nadać im pewną strukturę. Jednym ze sposobów jest utworzenie meta notatki. Notatki te stanowią bramę do innych notatek na podobny temat. Meta notatka może zawierać linki do powiązanych notatek lub treści innych notatek na dany temat.

Możemy to osiągnąć dzięki trybowi Org i dodatkowemu pakietowi Denote-Org, który wykorzystuje dynamiczne bloki w Org. Dynamiczne bloki to wszechstronna funkcja Org, która pozwala agregować myśli i łączyć je z odpowiednimi notatkami. Blok dynamiczny to fragment tekstu, który może być dynamicznie aktualizowany w miarę rozwoju kolekcji Denote. Meta notatka może zawierać blok dynamiczny, który wyświetla listę wszystkich notatek w danej kategorii lub uporządkowaną listę notatek pasujących do podpisu, a nawet zawiera tekst innych notatek.

Załóżmy, że pracujesz nad projektem, którego celem jest napisanie artykułu na temat *Daimoeioe* (wewnętrznego głosu), który przemawiał do starożytnego greckiego filozofa Sokratesa. Czytasz literaturę i tworzysz szereg trwałych notatek, w których używasz słowa kluczowego _daimonion. Kiedy zbierasz swoje myśli w spójną całość, możesz utworzyć metanotatkę.

Użyj funkcji `org-dynamic-block-insert-dblock (C-c C-x x)`, aby wyświetlić listę dostępnych bloków dynamicznych i wybierz `denote-links`. Następnie podaj wyrażenie regularne, które pasuje do notatek, które chcesz wyświetlić (w tym przypadku `_daimonion`). Wyrażenie regularne to zaawansowane kryterium wyszukiwania, podobne do użycia symbolu wieloznacznego w nazwie pliku. Denote wstawia do pliku Org blok zawierający listę linków do wszystkich notatek pasujących do tego kryterium wyszukiwania, na przykład:

```
#+BEGIN: denote-links :regexp "_daimonion"
- Platon Apologia
- Sokrates i Platon
- Platon:
  Kriton
#+END:
```

Korzystając z tego podejścia, można zestawić notatki z dziennika za dany miesiąc za pomocą magii wyrażeń regularnych. Użycie `^202309.*_journal` powoduje wyświetlenie wszystkich wpisów z dziennika za wrzesień 2023 r. To wyrażenie regularne wyświetla nazwy plików, które zaczynają się od 202309 i zawierają słowo kluczowe `_journal`. Tilde (^) oznacza, że wyszukiwanie odbywa się na początku nazwy pliku. Symbol `.` w środku wyrażenia regularnego oznacza, że dowolny znak (`.`) może występować wielokrotnie (`*`). Wyrażenia regularne są potężnym narzędziem do wyszukiwania, ale szczegółowe omówienie tego zagadnienia wykracza poza zakres niniejszej książki.

W miarę zmian w zbiorze notatek należy aktualizować blok dynamiczny za pomocą polecenia `C-c C-x C-u`. Polecenie to (`org-dblock-update`) odtwarza listę linków na podstawie najnowszych informacji. Dodanie argumentu uniwersalnego powoduje aktualizację wszystkich bloków dynamicznych w bieżącym buforze (`C-u C-c C-x C-u`).

W Denote dostępne są inne typy bloków dynamicznych, które umożliwiają wyświetlenie listy odnośników do notatki lub dołączenie tekstu innych notatek. Blok dynamiczny `denote-missing-links` wstawia listę odnośników do plików, które pasują do wyrażenia regularnego, ale nie są wymienione w pozostałej części bufora. Innym blokiem dynamicznym z odnośnikami jest `denote-backlinks`, który wyświetla listę wszystkich notatek odnoszących się do bieżącego bufora.

Bloki dynamiczne z linkami mogą zawierać parametry, które określają sposób wyświetlania informacji. Pierwszy parametr jest obowiązkowy, a pozostałe są opcjonalne:

- `regexp`: Wyrażenie regularne plików, które chcesz połączyć.
- `excluded-dirs`: Katalog, który ma zostać wykluczony z listy.
- `sort-by-component`: Sortowanie listy według tytułu, słów kluczowych lub sygnatury. Domyślnie sortowanie odbywa się według identyfikatora. Inne opcje to tytuł, słowo kluczowe lub sygnatura.
- `reverse-sort`: Po ustawieniu na `t` odwraca kolejność listy.
- `id-only`: Po ustawieniu na `t` wyświetla tylko identyfikatory, bez opisów.
- `include-date`: uwzględnia daty na liście.

Ostatni typ bloku dynamicznego Denote umożliwia transkluzję, która obejmuje zawartość innych notatek w meta notatce. Może to być przydatne, gdy masz wiele notatek z krótkimi cytatami lub przemyśleniami i chcesz zobaczyć je wszystkie na jednym ekranie. Blok dynamiczny `denote-files` działa podobnie jak inne wersje, ale ma kilka dodatkowych parametrów.

- `no-front-matter`: Po ustawieniu na `t` wyklucza przednią część plików.
- `file-separator`: po ustawieniu na `t` dodaje separator między kolejnymi plikami.
- `add-links`: Po ustawieniu na `t` dodaje link do każdego pliku na początku.

Podręcznik Denote-Org zawiera szczegółowe informacje (C-h R denote-org).

Liekieg eotes

Podpis i słowo kluczowe Denote oferują unikalny sposób porządkowania i kategoryzowania pomysłów. Dodatkowo Org może stać się osobistą wiki poprzez łączenie notatek. Chociaż termin „osobista wiki” może wydawać się sprzeczny, biorąc pod uwagę, że wiki są narzędziami do wspólnego pisania, łączenie notatek umożliwia tworzenie wzajemnie powiązanej sieci pomysłów.

Org posiada wszechstronny system linków. W poprzednich rozdziałach omówiono dodawanie hiperłączy do źródeł zewnętrznych i wewnętrznych (sekcje 4.5 i 5.3). Linkowanie do innych dokumentów dodaje

dotatkową strukturę do notatek. Jednak ta metoda ma pewien problem, ponieważ link przestaje działać, gdy plik docelowy zmienia nazwę lub lokalizację.

Denote rozszerza funkcjonalność Org, tworząc stabilne linki między notatkami. Link De-note przechowuje tylko identyfikator pliku docelowego, więc podpis, nazwa i słowa kluczowe mogą się swobodnie zmieniać bez ryzyka utworzenia martwych linków.

Możesz łączyć notatki i załączniki z linkami za pomocą funkcji `denote-link-or-create` (C-c w d i). Polecenie to wyświetla listę wszystkich dostępnych notatek przy użyciu systemu uzupełniania minibuffera, z której możesz wybrać cel i nacisnąć Enter. Aby zmodyfikować etykietę linku, naciśnij C-c C-I (`org-insert-link`), gdy kursor znajduje się na linku, i postępuj zgodnie z instrukcjami. Źródło linku Denote wygląda mniej więcej tak:

```
[[denote:20210208T150244][Opis]]
```

Ponieważ linki Denote używają tylko identyfikatora, można dowolnie zmieniać tytuł, podpis i tagi plików bez przerywania linku.

Jeśli wprowadzisz nazwę notatki, która jeszcze nie istnieje, Denote pozwoli Ci utworzyć nową notatkę, a następnie połączyć się z nią. Linki Denote są oznaczone kursywą w EWS, aby odróżnić je od linków do innych zasobów, takich jak strony internetowe.

Możesz również utworzyć link do załączników w notatce Denote. Nie jest jednak możliwe utworzenie linku zwrotnego z załącznika za pomocą Denote, ponieważ pliki te nie są notatkami. Denote może tworzyć linki tylko w plikach Org, Markdown lub zwykłym tekście.

Nie musisz przeszukiwać dokumentu, aby znaleźć odpowiednie linki. Przejdź do dowolnej powiązanej notatki bez przesuwania kursora za pomocą funkcji `denote-link-find-file` (C-c w d l). Funkcja ta wyświetla wszystkie notatki powiązane z otwartą notatką w minibufferze, gdzie można wybrać tę, do której chcesz przejść. Aby dowiedzieć się, które notatki są powiązane z tą, którą aktualnie czytasz, użyj funkcji `denote-find-backlink` (C-c w d b).

Fiedieg eotes aed attaeetmeets

Gromadząc tysiące notatek i załączników, potrzebujesz narzędzi, które pomogą Ci znaleźć potrzebne informacje lub stworzyć nowe powiązania między pomysłami. Najprostsza

metodą wyszukiwania plików jest otwarcie jednego z nich za pomocą standardowej funkcji wyszukiwania plików. System uzupełniania mini-bufora pomaga znaleźć to, czego potrzebujesz, przeszukując nazwy plików.

Jeśli na przykład chcesz filtrować notatki oznaczone tagiem „ekonomia”, wpisz C-x C-f, przejdź do folderu notatek i wpisz _ekonomia. Uzupełnianie minibuffera zawęży dostępne opcje. Jeśli potrzebujesz notatki z ekonomia w tytule, ale nie jako tagiem, użyj -ekonomia. Jeśli wpiszesz ekonomia bez przedrostka, minibuffer wyświetli wszystkie posty zawierające ten termin wyszukiwania w podpisie, tytule lub tagu. Wyrażenia regularne (sekcja 5.4) zwiększają możliwości wyszukiwania. Ponieważ uzupełnianie minibuffera wykorzystuje pakiet Orderless, spacja działa jako operator AND. Wpisanie ^2022 ==9a _art spowoduje wyszukanie wszystkich notatek, których nazwa pliku zaczyna się od „2022” (symbol kapelusza ^ odpowiada początkowi), zawiera podpis zaczynający się od ya (==9a) i ma tag pliku _art.

Pakiet Consult-Notes autorstwa Colina McLeara łączy możliwości pakietów Denote i Consult Daniela Mendlera, pomagając w wyszukiwaniu notatek za pomocą wyrażeń regularnych. Pakiet ten zapewnia również funkcje przeszukiwania treści notatek. Aby znaleźć notatkę na podstawie dowolnej części jej nazwy pliku, należy użyć funkcji consult-notes powiązanej z kombinacją klawiszy C-c w d f. Pakiet Consult zapewnia podgląd na żywo plików pasujących do wyszukiwania. Aby przeszukiwać podkatalog katalogu Denote, należy rozpocząć zapytanie od ukośnika, na przykład /attachments. Pakiet ten akceptuje wyrażenia regularne, jak wyjaśniono w poprzednim akapicie. Bardziej zaawansowanym użytkownikom pakiet ten umożliwia również definiowanie oddzielnych silosów plików Denote, co może być pomocne na przykład w przypadku, gdy chcesz oddzielić pliki służbowe od informacji prywatnych.

Jedna uwaga dotycząca korzystania z podglądu Consult. Pakiet OpenWith (sekcja 5.5) może kolidować z funkcjami podglądu w pakiecie Consult. System wyświetli podgląd typów plików skonfigurowanych w OpenWith za pomocą zewnętrznego oprogramowania, zakłócając przebieg pracy. Korzystanie z tego pakietu wymaga dostosowania consult-preview-excluded-files w celu wykluczenia wszelkich typów plików wymienionych w zmiennej openwith-associations. Wystarczy zarejestrować rozszerzenie pliku i dodać symbol \$, który jest symbolem wyrażenia regularnego oznaczającym koniec ciągu znaków. Dodanie „mp4\$” powoduje, że Consult nie wyświetli podglądu.

Pliki wideo. Kliknij przycisk INS w menu dostosowywania, aby dodać wiele typów plików.

Wyszukiwanie tytułów, tagów i innych metadanych to skuteczny sposób na uzyskanie dostępu do informacji

dzięki konwencji nazewnictwa Denote. Chociaż jest to dobry początek, czasami trzeba przeszukiwać treść notatek, a nie tylko tytuły i metadane. Pakiet Consult zapewnia wartościowy interfejs do osiągnięcia tego celu.

Funkcja `consult-notes-search-in-all-notes` (C-c w d g) uruchamia głębokie przeszukiwanie notatek. Pakiet wykorzystuje Grep, narzędzie do przeszukiwania plików tekstowych w poszukiwaniu wierszy pasujących do wyrażenia regularnego. Aby funkcja ta działała, Grep musi być zainstalowany na komputerze, co zazwyczaj ma miejsce w przypadku komputerów z systemem Linux i Apple.

Wyszukiwanie jest stopniowe, podobnie jak uzupełnianie w minibufferze. Podczas wpisywania kryteriów wyszukiwania pojawia się lista wyników, którą można zawęzić. Wyniki pokazują nazwę pliku i pasujące linie w każdym pliku. Termin wyszukiwania zaczyna się od hashtagu; po wpisaniu kolejnego takiego symbolu, na przykład `#topology#` homotopy, następna fraza zostanie wyszukana w wynikach pasujących do pierwszego wyrażenia regularnego. W tym przykładzie wyszukiwane są wszystkie notatki zawierające słowo „topology” i zawężane do tych plików, które zawierają również słowo „homotopy”.

Denote posiada również wbudowaną funkcję wyszukiwania, która wykorzystuje oprogramowanie Grep, opisane w instrukcji obsługi. Funkcja ta wyświetla wszystkie pasujące pliki w oddzielnym buforze, który można wykorzystać do dalszego przeglądania.

6.5 Wdrażanie metod sporządzania notatek za pomocą Denote

Pakiet Denote jest elastyczny i plastyczny, dzięki czemu można za jego pomocą wdrożyć dowolną opublikowaną metodologię sporządzania notatek. EWS nie promuje żadnych systemów sporządzania notatek, a niniejsza sekcja zawiera jedynie kilka wskazówek dotyczących wdrażania trzech popularnych metod za pomocą Emacs'a i Denote.

Staraj się nie rozpraszać „syndromem błyskotki” i skup się na pisaniu, zamiast ścigać najnowsze pomysły. Idealna metoda to taka, która rozwija się organicznie, dostosowana do

Twoje konkretne potrzeby. Możliwości konwencji nazewnictwa plików Denote i wyrażeń regularnych zapewniają w zasadzie wszystko, czego potrzebujesz.

PARA

Tiago Forte opracował metodę PARA do porządkowania cyfrowego życia (Forte, 2022). W jego systemie wszystkie zasoby cyfrowe znajdują się w jednym z czterech folderów:

1. *Pvojeets*
2. *Aveas*
3. *Resouvees*
4. *AveLives*

forte wykorzystuje przykład gotowania, aby zilustrować metodę PARA. *Pvojeets* to garnki i przybory potrzebne do przygotowania potrawy. Pliki w tej kategorii to materiały, nad którymi musisz pracować, aby zrealizować bieżące zadania. *Aveas* są jak składniki przechowywane w lodówce. Są to notatki, do których musisz regularnie zaglądać. Trzecia kategoria to *Resouvees*, która odnosi się do przedmiotów przechowywanych w zamrażarce. Są to tematy, które Cię interesują lub materiały badawcze. Wreszcie *AveLives*, które możemy nazwać spiżarnią, zawiera ukończone projekty lub te, które są wstrzymane.

Kluczem do tej metody jest to, że każdy plik należy tylko do jednej z tych czterech kategorii. Plik może zacząć jako zasób, stać się projektem, a zakończyć swoje życie w archiwum. W swoim pierwotnym pomysle forte sugeruje użycie czterech katalogów do przechowywania materiałów z każdej kategorii. Możesz wdrożyć tę metodę w Denote, przypisując każdej notatce jeden z czterech tagów. Idealnie byłoby, gdyby każda notatka należała tylko do jednej z tych czterech kategorii. Aby wyświetlić listę wszystkich notatek w kategorii *Pvojeets*, otwórz consult-notes i wyszukaj #projects itd. EWS zawiera specjalną funkcję umożliwiającą wdrożenie metody

PARA forte w Denote.

Funkcja ews-denote-assign-para przenosi notatkę do jednej z czterech kategorii PARA

, przypisując do niej słowo kluczowe. Jeśli słowo kluczowe PARA już istnieje, zostaje zastąpione nową wersją.

Zmienna `ews-para-keywords` zawiera słowa kluczowe używane w tej metodzie. Możesz dostosować tę zmienną, aby przetłumaczyć ją na swój język ojczysty lub użyć innego zestawu ekskluzywnych kategorii. Możesz na przykład skonfigurować tę zmienną tak, aby zaimplementować system ACCESS Nicka Milo, zmieniając opcje na Atlas, Kalendarz, Karty, Dodatki, Źródła, Przestrzenie i Spotkania. Ta funkcja może zastąpić dowolny system zarządzania plikami, który opiera się na folderach ze słowami kluczowymi Denote.

Jeśli chcesz używać folderów zamiast słów kluczowych, Denote też Ci to umożliwi. Dostosuj zmienną `denote-prompts`, żeby podczas tworzenia nowej notatki pytać o podkatalog.

Johnny.Decimal

System Johnny.Decimal wykorzystuje schemat numeracji do porządkowania plików, stworzony przez Johnny'ego Noble'a. Podstawową ideą jest podzielenie cyfrowego życia na mniej niż dziesięć szerokich obszarów. Możesz zacząć na przykład od kategorii „praca” i „osobiste”. Te główne kategorie są wirtualnymi półkami w Twojej cyfrowej bibliotece. Każda półka może pomieścić do dziesięciu pudełek. Na przykład w naszym przykładzie na półce „osobiste” moglibyśmy mieć pudełko „finanse”, „zdrowie” i „podróże”.

Kolejny krok polega na przypisaniu numerów do każdej kategorii. Johnny.Identyfikatory dziesiętne zaczynają się od 10–1y, ponieważ niższe numery są zarezerwowane dla konserwacji systemu. Folder 00 zazwyczaj zawiera indeks ułatwiający poruszanie się po systemie numeracji.

W naszym przykładzie *pevsoeal* to półka 10–1y, a pudełka są ponumerowane od 11 do 1y, na przykład *fieaeae* (11), *wvitiag* (12) i *tvavel* (13). Jest miejsce na siedem dodatkowych pudełek, ale poczekaj z wykorzystaniem tej przestrzeni, aż pojawi się taka potrzeba. W oryginalnym systemie numery tworzą nazwy katalogów początkowych.

System Johnny.Decimal jest podobny do systemu Dewey Decimal stosowanego w bibliotekach, ale ma mniej kategorii. System Johnny.Decimal opisuje Twoje życie, podczas gdy system Dewey

Oczywiście można również kategoryzować notatki przy użyciu podejścia Dewey Decimal; wybór należy do Ciebie.

Możesz zaimplementować system Johnny.Decimal, Dewey Decimal lub dowolny inny system wykorzystujący numerację porządkową za pomocą sygnatur Denote. Denote domyślnie nie używa sygnatur, więc musisz dostosować zmienną denote-prompts i zaznaczyć opcję „signature”.

Na przykład notatka dotycząca EWS może mieć sygnaturę 12=03, co oznacza, że należy do pudełka *wvities* na półce *pevsoeal*. Można użyć trzeciego poziomu w pudełku do numerowania poszczególnych plików, więc plik w pudełku *wvities* może być oznaczony numerem 12=03=01, 12=03=02 itd. Można użyć meta notatek (sekcja 5.4) do wyświetlenia listy wszystkich plików w tym pudełku, zmieniając sortowanie według komponentu na sygnaturę, aby uporządkować linki na liście. Bez tej instrukcji sortowania notatki są uporządkowane według identyfikatora.

```
+BEGIN: oznacz-linki :regexp „=12=03” :sortuj-według-komponentu podpis
- 12=03=01 ews cel
- 12=03=02 zettelkasten
- ... itd.
+KONIEC:
```

Zettelkastee

Wiele osób inspirowane jest koncepcją Zettelkasten Niklasa Luhmanna. Zettelkasten to niemieckie słowo oznaczające pudełko (Kasten), w którym przechowuje się notatki (Zettels). Luhmann był wpływowym socjologiem znanym ze swojej niezwyklej produktywności i obszernej kolekcji notatek, składającej się z ponad dziewięćdziesięciu tysięcy połączonych ze sobą kart indeksowych (Kadavy, 2021). Jego Zettelkasten ułatwił mu prowadzenie szeroko zakrojonych badań naukowych.

Próba naśladowania Luhmanna co do joty nie jest dobrym pomysłem. Głównym powodem jego produktywności było to, że był pracoholikiem, więc stosowanie jego systemu nie sprawi, że w magiczny sposób staniesz się bardziej produktywny. Jego metoda nie jest wyjątkowa. Prawie 30 lat temu podczas studiów artystycznych nauczono mnie, jak używać kart indeksowych do badań. Pamiętam, jak spędzałem wieczory przy stole w jadalni, układając karty indeksowe w celu uporządkowania esejów. Tym, co wyróżnia Luhmanna, jest jego niezachwiana dyscyplina w sporządzaniu notatek, cecha, która nadal stanowi inspirację.

Metoda Luhmanna dotycząca jego Zettelkasten obejmowała sygnaturę, która łączy karty sekwencyjnie w rozgałęzionej hierarchii. Johnny.Decimal i Zettelkasten różnią się tym, że w pierwszym przypadku liczby oznaczają kategorie. Natomiast w Zettelkasten liczby tworzą logiczne powiązania między notatkami. Oryginalny Zettelkasten Luhmanna ma co najmniej sześć poziomów zagnieżdżonych kategorii. Oto niewielki fragment jego oryginalnej pracy, pochodzący ze strony niklas-luhmann-archiv.de:

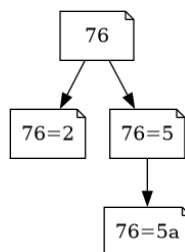
- 75: Przyczynowość
 - 75,2: Przyczynowość — motywacja
 - 75,5: Przyczynowość jako regularny porządek
 - * 75,5a: Przyczynowość: równoważność przyczyny i skutku

Notatki w metodzie Zettelkasten są uporządkowane tak, aby tworzyły spójną całość, dlatego Luhmann był tak płodnym pisarzem. Jego artykuły i książki powstawały w miarę dodawania notatek do systemu. W metodzie Zettelkasten każda notatka ma unikalny identyfikator, który można oznaczyć sygnaturą w Denote. Należy pamiętać, że pakiet Denote nie wymusza stosowania unikalnych sygnatur.

Poszczególne pliki to „Zettels”, a katalog Denote to „Kasten”. Nie można dokładnie skopiować składni Luhmanna, ponieważ używa on znaków niedozwolonych w sygnaturach Denote, których nie można używać w nazwach plików, takich jak symbol ukośnika (/). W powyższym przykładzie ostatnia kategoria miałaby sygnaturę 76=05=a, a poszczególne notatki miałyby sygnatury w postaci 76=05=a01. Należy używać zer na początku numerów, aby zapewnić odpowiednią kolejność notatek podczas korzystania z bloków dynamicznych do wyświetlania lub transkluzji notatek.

Metodę tę można wdrożyć za pomocą pakietu Denote-Sequence, który jest włączony w EWS. Pakiet ten wykorzystuje relacje pokrewieństwa, ponieważ każdy węzeł może być rodzicem, dzieckiem lub rodzeństwem. W powyższym przykładzie nota o sekwencji (podpisie) 75 jest rodzicem. Nota ta ma dwa dzieci (75,2 i 75,5), które są dla siebie rodzeństwem. Nota o podpisie 75,5a jest dzieckiem 75,5.

W pakiecie Denote-Sequence poszczególne elementy sekwencji są oddzielone symbolem równości. Rysunek przedstawia drzewo genealogiczne dla powyższego przykładu przy użyciu składni Denote.



Rysunek 6.1: Sekwencja plików Deeote.

Polecenie `denote-sequence` tworzy nową notatkę, która jest nowym elementem nadrzędnym, równorzędnym lub podrzędnym względem istniejącej notatki. Pakiet automatycznie przypisuje numery do notatek na podstawie wprowadzonych wyborów. Pakiet ten może wykorzystywać dwa rodzaje schematów: albo tylko cyfry, np. $1=13=3$, albo naprzemiennie litery i cyfry, np. $1=m=3$. Aby wybrać pożądane zachowanie, należy dostosować zmienną `denote-sequence-scheme`.

Aby uzyskać bardziej szczegółowy opis dodatkowych funkcji oferowanych przez ten pakiet, przeczytaj instrukcję obsługi Denote-Sequence, wybierając C-h R `denote-seq`.

6.6 Mamaging Twój cyfrowy ogród

Tvoja kolekcja notatek wymaga regularnej konserwacji, ponieważ pomysły i struktury myślowe ewoluują z biegiem czasu. Nazwy, słowa kluczowe i sygnatury notatek mogą się zmieniać wraz z rozwojem i rozkwitem Twojego cyfrowego ogrodu.

Pliki Org w Denote zawierają metadane w nazwie pliku i przedniej części pliku. Najlepiej, aby nazwa pliku i część początkowa były zsynchronizowane. Można również zmienić tytuł i słowa kluczowe, edytując tekst. Dla większej wygody można użyć funkcji `denote-keyword-add` (C-c w d k) i `denote-keyword-remove` (C-c w d K), aby zmienić tagi za pomocą uzupełniania minibuffera. Te dwie ostatnie funkcje zmieniają również nazwę pliku.

Użycie funkcji `denote-rename-file-using-front-matter` (C-c w d R) zmienia nazwę pliku przy użyciu danych zawartych w nagłówku. Funkcja ta pozostawia identyfikator bez zmian, nawet jeśli został on edytowany w nagłówku.

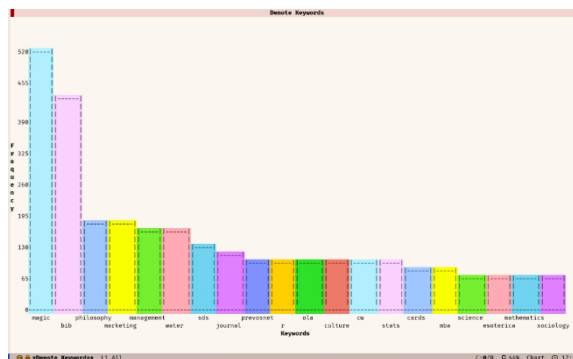
Pakiet Denote-Explore zapewnia wygodne funkcje do zarządzania kolekcją plików Denote. Skrót do pakietu Denote-Explore można znaleźć za pomocą prefiksu C-c w x. Skrót klawiaturowy dla poszczególnych poleceń można znaleźć w menu podręcznym Which-Key. Pakiet ten zapewnia cztery rodzaje poleceń:

1. *Statystyki Summavy*: zliczanie notatek, załączników i słów kluczowych.
2. *Raedom walks*: generowanie nowych pomysłów przy użyciu serendipity.
3. *Jaeitov*: Zarządzanie kolekcją Denote.
4. *Wizualizacje*: wizualizacja plików Denote w formie sieci.

Statystyki Summavy

Po całym dniu ciężkiej pracy w swoim cyfrowym ogrodzie wiedzy możesz chcieć przejrzeć notatki i załączniki w swojej kolekcji. Liczby są świetne, ale wykres jest wart tysiąca liczb. Wbudowany pakiet `chart.el` autorstwa Erica M. Ludlama to ciekawe narzędzie do tworzenia wykresów słupkowych w buforze tekstowym. Dwa polecenia Denote-Explore wizualizują podstawowe statystyki, wykorzystując funkcjonalność `chart.el`:

1. `denote-explore-keywords-barchart`: wizualizacja najczęściej *używanych* słów kluczowych (rysunek 5.2).
2. `denote-explore-extensions-barchart`: wizualizacja używanych rozszerzeń plików. Dzięki uniwersalnemu argumentowi wizualizuje tylko załączniki.



Rysunek 6.2: Przykład wykresu najpopularniejszych słów kluczowych.

Losowe spacery

Kreatywność wynika z mieszanki doświadczeń, emocji, podświadomych rozmyślań i połączenia przypadkowych pomysłów. Wprowadzenie elementów losowych do procesu twórczego generuje ścieżki myślenia, którymi w innym przypadku być może byś nie podążał. Metoda ta może być korzystna, gdy utknieś w rutynie lub wolisz przeglądać swoje pliki losowo. Losowy spacer to dowolna sekwencja zdarzeń bez określonej relacji między kolejnymi krokami. Losowo wędrujesz po swoim drugim mózgu, przeskakując do losowej notatki, powiązanej lub niepowiązanej z bieżącym buforem. Pakiet Denote-Explore udostępnia cztery polecenia, które pozwalają wprowadzić element losowości do Twoich poszukiwań:

1. denote-explore-random-note (C-c w x r): Przejdź do losowej notatki lub załącznika.
2. denote-explore-random-link (C-c w x l): Przejdź do losowej powiązanej notatki (do przodu lub do tyłu) lub załączników (tylko do przodu).
3. denote-explore-random-keyword (C-c w x k): Przejdź do losowej notatki lub załącznika zawierającego wybrane słowa kluczowe.

4. `denote-explore-random-regex` (C-c w x x): Przejdź do losowej notatki pasującej do wyrażenia regularnego.

Domyślnie funkcje te przenoszą użytkownika do dowolnego pliku tekstowego Denote (zwykły tekst, Markdown lub Org). Argument uniwersalny (C-u) uwzględnia załączniki jako potencjalne miejsca docelowe losowego przejścia.

Podczas przechodzenia do losowego pliku z tymi samymi słowami kluczowymi można wybrać jedno lub więcej słów kluczowych z bieżącego bufora. Jeśli bieżący bufor nie jest plikiem Denote, wybierz dowolne dostępne słowa kluczowe z kolekcji Denote. Symbol gwiazdki * wybiera wszystkie słowa kluczowe z listy uzupełnień.

The jaeitov

Tak jak każdy budynek potrzebuje dozorczy, który dba o czystość i wykonuje drobne prace konserwacyjne, tak samo Twoja cyfrowa przestrzeń domowa wymaga pomocy. Po pewnym czasie korzystania z Denote może okazać się, że potrzebujesz dozorczy, który zadba o porządek w Twojej kolekcji.

Pakiet Denote zapobiega powielaniu identyfikatorów podczas tworzenia nowych notatek. Jednak czasami może być konieczne ręczne utworzenie daty i godziny dla starego dokumentu, jeśli data utworzenia różni się od daty w systemie plików, jak wyjaśniono w sekcji 5.4. Ręczne dodawanie identyfikatora Denote wiąże się z ryzykiem powielenia. Duplikaty mogą również powstać podczas eksportowania plików Denote Org, ponieważ eksportowane pliki mają tę samą nazwę, ale inne rozszerzenie.

Polecenie `denote-explore-identify-duplicate-notes` wyświetla listę wszystkich zduplikowanych identyfikatorów w buforze wyskakującym. Należy zachować ostrożność podczas zmiany identyfikatora pliku Denote, ponieważ może to spowodować utratę integralności linków. Należy upewnić się, że plik, którego nazwę zmieniasz, nie ma żadnych linków kierujących do niego. Użyj polecenia `denote-find-backlink` (C-c w d b), aby sprawdzić, czy plik ma jakieś linki kierujące do niego.

Denote-Explore oferuje kilka funkcji do zarządzania słowami kluczowymi i porządkowania kolekcji. Słowo kluczowe oznacza kategorię, więc idealnie byłoby, gdyby wszystkie słowa kluczowe były używane co najmniej

dwukrotnie. Polecenie `denote-explore-single-keywords` udostępnia obszerną listę tagów plików, które są używane tylko raz, co ułatwia identyfikację i rozwiązywanie wszelkich problemów. Lista pojedynczych słów kluczowych jest wyświetlana w minibufferze, skąd można otworzyć odpowiednią notatkę lub załącznik, usprawniając proces zarządzania notatkami.

W zależności od struktury Denote, notatki powinny zawierać co najmniej jedno słowo kluczowe. Polecenie `denote-explore-zero-keywords` wyświetla wszystkie notatki i załączniki bez słów kluczowych w minibufferze, umożliwiając ich otwarcie i rozważenie dodania słowa kluczowego lub pozostawienia ich bez zmian.

Możesz zmienić nazwę lub usunąć słowa kluczowe za pomocą funkcji `denote-explore-rename-keyword`. Wybierz jedno lub więcej istniejących słów kluczowych z listy uzupełnień i wprowadź nową nazwę słowa kluczowego (słów kluczowych). Ta funkcja zmienia nazwy wszystkich wybranych słów kluczowych na ich najnowszą wersję. Usuwa oryginalne słowo kluczowe ze wszystkich istniejących notatek, gdy jako nowe słowo kluczowe wprowadzisz pusty ciąg znaków. Ta funkcja przegląda wszystkie notatki i załączniki zawierające wybrane słowa kluczowe i prosi o potwierdzenie przed wprowadzeniem jakichkolwiek zmian. Nowa lista słów kluczowych jest przechowywana w porządku alfabetycznym, a przednia część jest zsynchronizowana z nazwą pliku.

Denote przechowuje metadane przy użyciu pomysłowej konwencji nazewnictwa plików. Niektóre z tych metadanych są kopiowane do przedniej części notatki, co może powodować rozbieżności między dwoma źródłami metadanych. Funkcja `denote-explore-sync-metadata` sprawdza wszystkie notatki i prosi użytkownika o zmianę nazwy każdego pliku, w którym te dwa zestawy danych są niezgodne. Dane przedniej części są źródłem prawdy. Funkcja ta zapewnia również alfabetyczne uporządkowanie słów kluczowych, co ułatwia wyszukiwanie notatek.

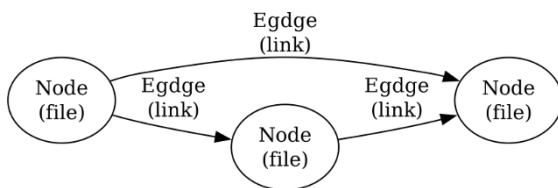
6.7 Wizualizacja notatek jako sieci

Przekładanie pomysłów na tekst wymaga linearnego sposobu myślenia, ponieważ można przetwarzać tylko jedno słowo na raz. Czytamy tekst od góry do dołu, od początku do końca, co oznacza, że myśli muszą być uporządkowane. W moim papierowym notatniku regularnie używam diagramów, takich jak mapy myśli, zamiast narracji, aby przekazać myśli. Myślenie wizualne

jest innym sposobem podejścia do pomysłów, ponieważ może ujawnić wcześniej niewidoczne powiązania. Jedną z najpopularniejszych metod wizualizacji powiązanych dokumentów jest diagram sieciowy.

Łączenie pomysłów w sieci nie jest nowoczesnym narzędziem. Średniowieczni mnisi rysowali diagramy na marginesach czytanych książek, łącząc swoje krótkie notatki liniami. Diagramy te są źródłem nawiasów klamrowych }, które początkowo oznaczały rozgałęzienie pomysłu (Even-Ezra, 2021).

Postrzeganie swoich myśli jako sieci pomaga odkryć dotychczas niewidoczne powiązania między nimi. Wizualizacja cyfrowego ogrodu Denote jako sieci może również pomóc w procesie twórczym. Schemat sieci składa się z węzłów (wierzchołków) i krawędzi. Każdy węzeł reprezentuje plik w systemie Denote, a każda krawędź jest połączeniem między notatkami (rysunek 5.3).



Rysunek 6.3: Przykłady sieci plików Denote.

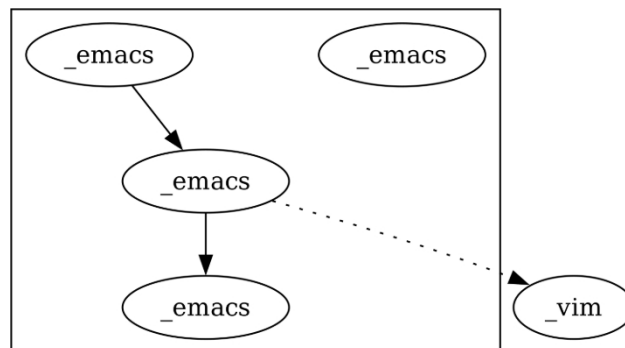
Pakiet Denote-Explore wykorzystuje funkcje zewnętrznego oprogramowania do wizualizacji struktury części sieci Denote. Aby wizualizować sieci, należy zainstalować oprogramowanie Graphviz. Narzędzie to konwertuje opisy sieci w postaci zwykłego tekstu na pliki graficzne. Wszystkie diagramy sieciowe zawarte w niniejszej książce zostały utworzone za pomocą programu GraphViz.

Denote-Explore nie zapewnia środowiska na żywo do przeglądania struktury kolekcji Denote. Funkcja ta służy do analizowania struktury notatek, a nie do pełnienia roli alternatywnego interfejsu użytkownika. Podgląd sieci notatek na żywo jest pułapką dopaminową. Chociaż obserwowanie rozwoju sieci myśli na własne oczy jest satysfakcjonujące, może to również stanowić czynnik rozpraszający uwagę.

Polecenie `denote-explore-network` zapewnia dostęp do trzech typów diagramów sieciowych, które pozwalają zbadać relacje między plikami:

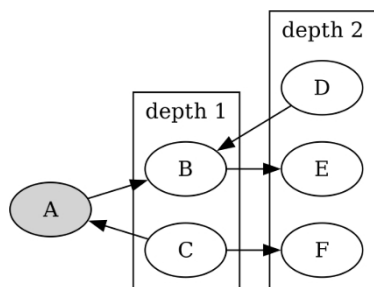
1. Społeczność notatek
2. Sąsiedztwo notatki
3. Struktura słów kluczowych

Społeczność składa się z notatek, które pasują do wyrażenia regularnego. Na przykład wszystkie notatki z Emacs jako słowem kluczowym (`_emacs`) są pokazane na rysunku 5.4. Wykres pokazuje wszystkie notatki pasujące do wyrażenia regularnego i ich powiązania. Wszelkie linki do notatek, które nie pasują, są usuwane i nie są wyświetlane (linia kropkowana do notatki `_vim` w przykładzie). Wykres pokazuje również wszystkie notatki osierocone, takie jak te bez połączeń. Użycie pustego wyrażenia regularnego generuje sieć wszystkich dostępnych plików.



Rysunek 6.4: Społeczność plików Deeote z słowem kluczowym `_Emacs`.

Sąsiedztwo notatki składa się ze wszystkich plików powiązanych z nią na jednym lub kilku poziomach. Algorytm wybiera elementy wykresu spośród powiązanych i odwrotnie powiązanych notatek (takich jak A do B i C do A na rysunku 5.5). Ten typ sieci wizualizuje możliwe ścieżki do śledzenia za pomocą funkcji „oznacz-zbadaj-losowe połączenie” omówionej w sekcji 5.5. Wygeneruj wykres sąsiedztwa za pomocą funkcji „oznacz-zbadaj-sieć”, wybierz „Sąsiedztwo” i wprowadź głębokość wykresu. Podczas tworzenia tego wykresu z bufora, a nie z notatki Denote, system prosi również o wybranie notatki źródłowej. Głębokość większa niż trzy jest



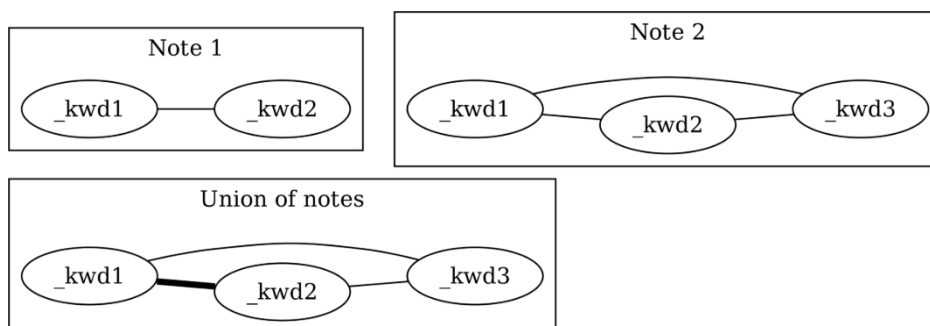
Rysunek 6./: Deepote eigtbouvlood plików (deptł = 2).

Zazwyczaj nie jest to zbyt pouczające, ponieważ sieć staje się zbyt rozbudowana i trudna do odczytania lub natrafia się na granice wyspy połączonych ze sobą notatek wybranego pochodzenia.

Będą pliki bez połączeń, samotne, odizolowane notatki. W zależności od strategii tworzenia notatek, możesz chcieć, aby wszystkie notatki były połączone z inną notatką. Funkcja `denote-explore-isolated-notes` wyświetla listę wszystkich notatek bez połączeń lub odnośników, które możesz przejrzeć. Możesz wybrać dowolną notatkę bez połączeń i dodać do niej kilka odnośników. Wywołanie tej funkcji z uniwersalnym argumentem C-u powoduje wykluczenie załączników z listy samotnych plików.

Ostatnią dostępną metodą wizualizacji kolekcji Denote jest stworzenie sieci słów kluczowych. Dwa słowa kluczowe są połączone, gdy występują w tej samej notatce. Wszystkie słowa kluczowe w notatce tworzą kompletną sieć. Kompletna sieć to taka, w której wszystkie węzły są ze sobą połączone. Połączenie wszystkich kompletnych sieci ze wszystkich plików w kolekcji Denote definiuje sieć słów kluczowych. Relacja między dwoma słowami kluczowymi może występować w wielu notatkach, więc powiązania między słowami kluczowymi są węższe. Grubość linii między dwoma słowami kluczowymi wskazuje częstotliwość (wagę) ich relacji (rysunek 5.5).

Podczas gdy pierwsze dwa typy wykresów są ukierunkowane (strzałki wskazują kierunek powiązań), sieć słów kluczowych jest nieukierunkowana, ponieważ są to dwukierunkowe powiązania między słowami kluczowymi. Poniższy diagram ilustruje scenariusz z dwoma węzłami i trzema możliwymi słowami kluczowymi, pokazując, jak łączą się one, tworząc sieć słów kluczowych.



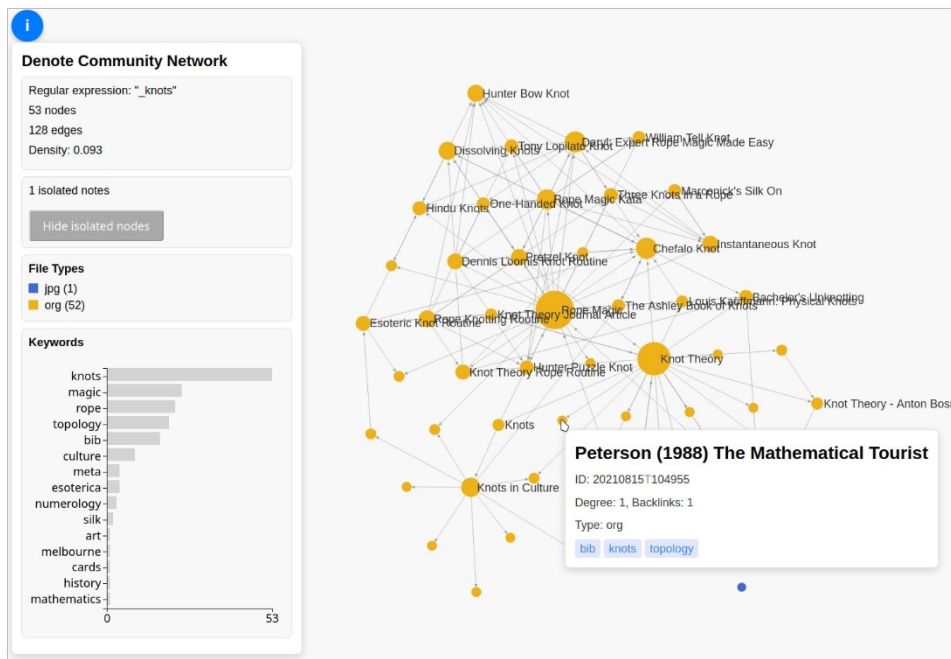
Rysunek 6.6: Sieć słów kluczowych.

Rozmiar każdego węzła jest proporcjonalny do liczby notatek, które są z nim powiązane lub do niego prowadzą. Gdy stopień jest większy niż dwa, wyświetlana jest nazwa węzła. Podczas przeglądania pliku SVG w przeglądarce internetowej, najechnie myszką na węzeł powoduje wyświetlenie metadanych notatki (rysunek 5.7).

Kliknięcie linku spowoduje otwarcie odpowiedniego pliku. Musisz skonfigurować przeglądarkę, aby otwierała pliki Org za pomocą programu Emacs. Najlepiej skonfigurować program Emacs jako serwer, aby nie otwierał nowej wersji dla każdego klikniętego linku.

Po wprowadzeniu zmian w notatkach można ponownie wygenerować tę samą sieć bez konieczności wprowadzania nowych parametrów. Polecenie `denote-explore-network-regenerate` odtwarza bieżący wykres z tymi samymi parametrami, co poprzedni, co jest przydatne, gdy chcesz zobaczyć wynik zmian bez ponownego wprowadzania kryteriów wyszukiwania.

Dodanie większej liczby połączeń między notatkami może poprawić funkcjonowanie Twojego drugiego mózgu, ale nie musi tak być. Skrajnym przypadkiem jest kompletna sieć, w której każdy plik jest połączony z każdym innym plikiem, lub sieć bez żadnych połączeń. W takich sytuacjach brakuje interesującej struktury, która dostarczałaby jakichkolwiek spostrzeżeń. Dlatego należy zwracać uwagę na sposób łączenia notatek i załączników, aby diagramy sieciowe pomagały łączyć pomysły, a nie były tylko przyjemne dla oka.



Rysunek 6.7: Przykład sieci społecznościowej D3.js.

6.8 Dowiedz się więcej o Demote

W niniejszym rozdziale przedstawiono krótkie wprowadzenie do pakietu Denote oraz niektórych jego pakietów pomocniczych. Pakiety te oferują szeroki zakres funkcji umożliwiających dostosowanie programu Emacs do własnych preferencji.

Obszerna instrukcja obsługi Denote opisuje szczegółowo wszystkie jego funkcje, wraz z licznymi opcjami konfiguracyjnymi (C-h R denote). Prot, opiekun pakietu Denote, opublikował również kilka rozszerzeń, z których część została wspomniana powyżej. Każdy z tych pakietów pomocniczych zawiera szczegółową instrukcję obsługi.

Rozszerzenie *Deeote-Silo* sprawia, że praca z silosami (oddzielnymi zbiorami plików Denote) jest nieco wygodniejsza. Silosy mogą być przydatne, gdy na przykład chcesz oddzielić swoje prywatne notatki od służbowych. Pakiet *Deeote-Mavkdowe* rozszerza funkcjonalność

pracy z plikami Markdown. Dostęp do tych instrukcji można uzyskać za pomocą C-h R denote i wybierając odpowiedni pakiet.

Denote w połączeniu z innymi funkcjami Emacsa zapewnia oszałamiającą różnorodność opcji i konfiguracji. Nie daj się skusić, aby od razu zagłębiać się w szczegóły w celu zdefiniowania idealnego systemu. Najlepszą radą jest poświęcenie większości czasu na pisanie notatek, a nie na szczegółowe dostosowywanie systemu. Jak napisał kiedyś pionier informatyki Donald Knuth:

Przedwczesna optymalizacja jest źródłem wszelkiego zła.

Teraz, gdy zebrałeś i uporządkowałeś już wiele notatek, nadszedł czas, aby rozpocząć projekt pisarski. W następnym rozdziale pokazano, jak pracować nad rozbudowanym projektem pisarskim przy użyciu Org.

Produkcja: Przygotowanie manuskryptu

Rozdział 3 położył podwaliny pod pisanie dokumentów w formacie zwykłego tekstu, ustanawiając podstawowe pojęcia, które stanowią kluczową podstawę skutecznego pisania. W rozdziale 4 przedstawiono podstawy składni Org, umożliwiające organizowanie i strukturyzowanie tekstu w sensowny sposób. Opierając się na tych fundamentalnych zasadach, niniejszy rozdział zawiera kompleksowy przewodnik dotyczący przygotowywania manuskryptu przeznaczonego do publikacji w *Zmaes Wvities Studio* (EWS). Omówiono w nim unikalne możliwości trybu Org oraz zintegrowano dodatkowe narzędzia zaprojektowane w celu usprawnienia i ulepszenia procesu pisania.

W tym rozdziale dowiesz się, jak dodawać uwagi redakcyjne, płynnie włączać cytaty i tworzyć odsyłacze, które zwiększają przejrzystość i spójność dokumentu. Aby jeszcze bardziej wzbogacić i urozmaicić swój styl pisania, możesz skorzystać z wbudowanego słownika i tezaursu programu Emacs. Ponadto w tym rozdziale omówiono kluczowe techniki zarządzania dużymi projektami pisarskimi, w tym strategię kontroli wersji i podejścia do pisania zespołowego, które ułatwiają współpracę z innymi autorami.

W ostatniej części tego rozdziału przedstawiono szereg innych trybów tekstowych, które mogą zainteresować autorów. Wśród nich znajduje się Markdown, który szczególnie dobrze nadaje się do tworzenia dokumentacji technicznej, oraz specjalistyczny tryb fountain przeznaczony do pisania scenariuszy filmowych i teatralnych. Dzięki tym informacjom będziesz dobrze przygotowany do odkrywania różnorodnych stylów i formatów pisania, poszerzając swoje możliwości jako pisarz.

7.1 Dodawanie notatek, cytatów i odniesień

Typowy projekt pisarski to coś więcej niż tylko słowa w manuskrypcie. Kompleksowy zestaw funkcji Org, w tym inteligentne konstrukcje tekstu zwykłego dla komentarzy, cytatów i odniesień, zapewnia wzbogacenie manuskryptu. Omówiliśmy już niektóre metadane w poprzedniej części pliku Org, takie jak `#+title:`. Org zapewnia również składnię umożliwiającą włączanie obrazów, tabel i szuflad, jak wyjaśniono w poprzednich rozdziałach.

W tej sekcji dodajemy więcej funkcji Org do zestawu narzędzi, które pomogą Ci przygotować manuskrypt do publikacji. Org umożliwia dodawanie do dokumentu notatek, które są ignorowane w końcowym produkcie. W tej sekcji wyjaśniamy również, jak dodawać cytaty z bibliografii i tworzyć odsyłacze do nagłówków, tabel i innych części dokumentu.

Dodawanie dokumentów

W poprzednim rozdziale szczegółowo opisano wszechstronność tworzenia notatek w Emacs, zarówno w ramach pojedynczego pliku Org, jak i sieci połączonych ze sobą plików. Notatki można również tworzyć w ramach projektów pisarskich, poza formalnym systemem tworzenia notatek. Notatki te służą jako przypomnienia o zadaniach do wykonania lub innych sprawach, które chcesz ukryć przed ostateczną wersją produktu. Możesz na przykład dołączyć listę kontrolną, linki do stron internetowych lub notatki w swoim cyfrowym ogrodzie Denote lub inne ważne informacje dotyczące procesu pisania. Jeśli chcesz zapisać notatki niezwiązane z bieżącym dokumentem, użyj systemu przechwytywania Org, aby zapisać ulotną notatkę w skrzynce odbiorczej (rozdział 5).

Tryb Org oferuje trzy elastyczne mechanizmy dodawania notatek redakcyjnych. Pierwsze podejście jest proste. Każda linia zaczynająca się od znaku hash/funt (`#`) jest komentarzem

, które Emacs pomija podczas eksportowania tekstu do ostatecznego wyniku.

To jest komentarz redakcyjny.

Możesz również dodawać notatki w bloku strukturalnym, konstrukcji trybu Org służącej do dodawania różnego rodzaju treści. Aby dodać notatkę, użyj skrótu C-c C-, który wyświetla menu umożliwiające wstawienie szablonów (org-insert-structure-template). Naciśnij C, aby dodać sekcję komentarza, co spowoduje dodanie dwóch linii do manuskryptu. Wszystko, co napiszesz między tymi dwiema liniami, będzie komentarzem. Aby zachować porządek na ekranie, możesz zwinąć bloki komentarzy za pomocą klawisza Tab, gdy kursor znajduje się na linii #+BEGIN lub #+END. Niektóre inne opcje w menu szablonów strukturalnych omówiono w następnym rozdziale.

Możesz również użyć tej konstrukcji do przechowywania tekstu, który nie jest już częścią głównego dokumentu, ale nie chcesz go usuwać. Po wybraniu fragmentu tekstu i wydaniu tego polecenia, wybrany tekst zostanie przekształcony w komentarz. W blokach komentarzy można umieszczać bloki dynamiczne. Jest to przydatne do dodawania linków do zbioru notatek związanych z manuskrytem.

```
#+BEGIN_COMMENT
- To jest komentarz
- Kolejny komentarz
#+END_COMMENT
```

Trzecia metoda polega na użyciu szuflady pod bieżącym nagłówkiem, która działa jak blok struktury komentarza. Główna różnica polega na tym, że szuflady mają nazwę, na przykład :NOTES:. Szufladę można otworzyć lub zamknąć za pomocą klawisza Tab, gdy kursor znajduje się na niej. Poniższy przykład pokazuje, jak może wyglądać szuflada w tekście.

```
:NOTES:
- To jest wewnątrz szuflady.
- Można ją wypełnić notatkami.
:KONIEC:
```

Szuflady są zazwyczaj umieszczane bezpośrednio pod nagłówkiem, ale można je wstawiać w dowolnym miejscu. Wywołanie `org-insert-drawer`, powiązane z `C-c C-x d`, powoduje interaktywne wstawienie szuflady w bieżącej lokalizacji. W minibufferze wpisuje się nazwę szuflady (zgodnie z konwencją wielkimi literami, na przykład NOTES) i wypełnia jej zawartość. Jeśli zaznaczysz blok tekstu i utworzysz szufladę, tekst ten pojawi się w jej wnętrzu. Możesz również użyć tej metody, aby ukryć w końcowym produkcie fragmenty tekstu, które warto zachować.

Specjalna funkcja (EWS) (`ews-org-insert-notes-drawer`) tworzy szufladę notatek powiązaną ze skrótem klawiaturowym `C-c w n`. Funkcja ta przenosi kursor poniżej nagłówka sekcji, którą piszesz, i generuje nazwy szuflad jako NOTES. Jeśli dla tej sekcji istnieje już szuflada notatek, funkcja tworzy nową linię na końcu istniejących. Po napisaniu notatek skrót klawiszowy `C-u C-<spc>` przenosi użytkownika z powrotem do pierwotnej pozycji w tekście.

Te szuflady na notatki są pomocne przy dodawaniu listy kontrolnej działań, które należy wykonać, aby ukończyć rozdział manuskryptu. Sekcja y.1 wyjaśnia, jak utworzyć listę kontrolną, aby śledzić postępy w realizacji listy rzeczy do zrobienia.

Dodaj citations

Cytaty są istotą pisarstwa naukowego i walutą kariery akademickiej, świadczącą o wpływie i znaczeniu Twojej pracy. Tryb Org może być Twoim sprzymierzeńcem w tej podróży, oferując narzędzie do zarządzania cytatami, które odczytuje pliki BibTeX, BibLaTeX lub CSL. Aby rozpocząć, musisz utworzyć bibliografię. Można to zrobić ręcznie lub połączyć plik z narzędziem do zarządzania bibliografią, takim jak Zotero, omówionym w rozdziale 5.

Polecenie `org-cite-insert` (`C-c C-x @`) jest podstawowym narzędziem do wstawiania cytatów. W *Zmaes Wvities Studio* polecenie to otwiera menu Citar, umożliwiające wybranie jednej lub kilku publikacji. Aby wybrać wiele odniesień, należy użyć klawisza Tab po każdym wyborze. Org wstawia cytat, który wygląda następująco:

```
[cite:@einstein_1905; @newton_1728]
```


Menu Citar wskazuje, czy dany wpis jest cytowany w bieżącym pliku, umieszczając literę C na liście publikacji. Użyj klawiszy :c w Citar, aby wyświetlić tylko cytowane wpisy.

Pakiet Citar udostępnia również kilka wygodnych funkcji do zarządzania cytatami. Aby zmienić kolejność cytatów w bloku, należy użyć klawiszy Shift i strzałek w lewo/w prawo, gdy kursor znajduje się na cytacie (`citar-org-shift-reference-left` / `-right`).

Cytat może mieć przedrostek i przyrostek, na przykład:

```
[cite:see@darwin1859 p. 123]
```

Cytat ten można przedstawić jako „(Darwin, 185y str. 123)”, w zależności od stylu cytowania, omówionego w następnym rozdziale. Prefiks i sufiks można edytować, gdy kursor znajduje się na cytacie, za pomocą M-p (`citar-org-update-prefix-suffix`). W sekcji 8.2 następnego rozdziału omówiono, w jaki sposób system zarządzania cytatami Org je eksportuje i jak je formatować.

Aby uzyskać dostęp do zasobów związanych z cytatem, użyj polecenia `org-open-at-point` (C-c C-o). Spowoduje to otwarcie menu Citar dla cytatu znajdującego się pod kursorem. Jeśli nie ma żadnej istniejącej notatki ani załącznika, a klikniesz link do cytatu, system poprosi Cię o utworzenie nowej notatki. Cytaty, których nie ma w bibliografii, są zaznaczone na czerwono.

Bibliografia może być globalna lub lokalna. Bibliografia globalna jest dostępna z dowolnego miejsca w Emacsie. Jest ona definiowana przez pliki BibTeX znajdujące się w zmiennej `ews-bibtex-directory`. Natomiast lokalny plik bibliografii jest dostępny tylko w bieżącym buforze Org. Bibliografia globalna jest ustawiana w konfiguracji. Jak omówiono w sekcji 5.2, może ona składać się z jednego lub więcej plików. Bibliografia lokalna jest powiązana z bieżącym buforem za pomocą słowa kluczowego w przedniej części, na przykład:

```
#+bibliografia: bibfile.bib
```

Podpisy aed vefeeveees

Podczas czytania obszernego dokumentu, takiego jak ta książka, wykorzystanie wewnętrznych linków poprawia czytelność tekstu. Ta funkcja pozwala kierować czytelnika

do konkretnych sekcji, rysunków lub innych części manuskryptu, ułatwiając poruszanie się po tekście.

Odwoływanie się do rysunków i tabel jest dziecinnie proste. Przypisz nazwę do elementu, używając słowa kluczowego `#+name:` pod podpisem, na przykład: `#+name: fig-example`. Odwołując się do tego elementu, należy umieścić jego nazwę w podwójnych nawiasach kwadratowych (`[[fig-example]]`). Org automatycznie tworzy link do lokalizacji obrazu lub tabeli, jak wyjaśniono w sekcji

4.5. Odnośniki te mogą kierować do innych dokumentów, o ile są one częścią tego samego projektu.

Odnośniki do sekcji lub rozdziałów składają się z linków do nazwy docelowego nagłówka. Tak więc link do tej konkretnej sekcji brzmiałby `[[Podpisy i odnośniki]]`. Takie podejście niesie ze sobą ryzyko powstania nie działających linków w przypadku zmiany nazwy nagłówka i zapomnienia o zmodyfikowaniu linków, które do niego odsyłają.

Możesz również dodać właściwość do nagłówka za pomocą niestandardowego identyfikatora. Użyj `org-set-property` (C-c C-x p), wybierz `CUSTOM_ID` i wprowadź żądany identyfikator. Linki do niestandardowego identyfikatora wymagają znaku octothorpe, znanego również jako hashtag lub symbol funta. Tak więc link do tej sekcji książki może mieć niestandardowy identyfikator `sec-references` z linkami zdefiniowanymi jako `[[#sec-references]]`. Bez hashtagu tryb Org połączy go z sekcją o tej samej nazwie. Co może być mylące, odsyłacze do nazwanych tabel i obrazów nie wymagają hashtagu.

Nazwy linków wewnętrznych mogą być dowolne, o ile są unikalne w manuskrypcie. Dobrą praktyką jest kategoryzowanie linków za pomocą prefiksów, takich jak `fig-`, `tab-`, `chap-` itp., aby zapewnić przejrzystość i zmniejszyć ryzyko powielania. Należy unikać stosowania dwukropka w nazwie, ponieważ może to powodować problemy podczas eksportowania do formatu HTML. Org może dodać unikalny identyfikator do nagłówka, używając ciągu losowych liter i cyfr, aby zapewnić unikalność nazw linków, ale stosowanie takich identyfikatorów utrudnia śledzenie manuskryptu (sekcja 5.3). Org eksportuje linki odsyłające jako hiperłącza w ostatecznej wersji dokumentu, omówionej w rozdziale 8.

72 Narzędzia zwiększające produktywność

Emacs może pomóc Ci zmaksymalizować produktywność i utrzymać wysoką jakość poprzez usprawnienie procesów pisania za pomocą różnych narzędzi, takich jak autouzupełnianie tekstu. Pisanie wymaga niepodzielnej uwagi, dlatego Emacs zapewnia również spokojne środowisko, wolne od rozpraszających elementów oprogramowania graficznego. Ponadto Emacs oferuje funkcje integracji z narzędziami gramatycznymi, słownikiem i tezaurem, które pozwalają poprawić jakość pisania.

Seavetieg tle eoeteeet of youv pvojeet

Polecenie `consult-grep` przeszukuje zawartość wszystkich plików w bieżącym folderze (C-c w g). Polecenie to wymaga zewnętrznego oprogramowania Grep, jak wyjaśniono w sekcji 5.4. Używając tego polecenia z argumentem uniwersalnym, można najpierw wybrać katalog do przeszukania.

Funkcja wyszukiwania zapewnia elastyczność. Możesz wyszukiwać za pomocą wyrażeń regularnych. Termin wyszukiwania pomiędzy hashtagami oferuje listę plików zawierających dane słowo. Każdy kolejny termin wyszukiwania zawęży listę.

- `#alpha omega`: Wyszukaj alpha i omega w dowolnej kolejności.
- `#alpha.*beta`: Wyszukaj alpha przed omega.
- `#alpha#omega`: Wyszukaj alpha, szybki filtr dla omega.

Skróty Zxpaeidieg

Rządy i przedsiębiorstwa zaśmiecają swoje teksty skrótami i akronimami. Skróty nie są problemem współczesnym i są popularne od początku istnienia pisma. Rzymskie inskrypcje na kamieniu są w większości skrócone, co utrudnia czytanie, nawet jeśli rozumie się łacinę. Rzymscy pisarze skracali słowa, ponieważ oszczędzało to wiele wysiłku związanego z wykuwaniem tekstu w marmurze. Jednak w erze pisania elektronicznego nie potrzebujemy już skrótów, ponieważ wysiłek związany z pisaniem pełnych słów lub fraz jest znikomy. Elektroniczne

systemy pisania mogą automatycznie rozszerzać skróty do pełnego kontekstu, dzięki czemu tekst jest łatwiejszy do odczytania.

Skrót „abbrev” w programie Emacs to predefiniowany fragment znaków, który rozszerza się na coś innego. Teksty techniczne często mają charakter formułowy, więc naukowiec może chcieć automatycznie zamienić „stbl” na „pokazano w tabeli” lub podobne typowe zwroty akademickie (fox & Tigchelaar, 2015).

Tryb skrótów Emacs to wbudowana biblioteka, która definiuje bibliotekę osobistych fragmentów, które rozszerzają się do większych fragmentów tekstu. EWS domyślnie włącza tryb skrótów dla wszystkich trybów tekstowych. Podstawowa funkcjonalność polega na tym, że użytkownik definiuje skrót, na przykład „ouat”. Następnie, gdy wpiszesz ten skrót, Emacs rozszerzy go do „once upon a time” (dawno, dawno temu).

Skróty (abbrevs) mogą mieć zastosowanie do wszystkich trybów (globalnie) lub tylko do określonego trybu. W tej sekcji omówiono tylko skróty globalne, ale te same zasady mają zastosowanie do skrótów specyficznych dla trybu.

Funkcja `define-global-abbrev` definiuje globalny skrót. Wystarczy wpisać skrót i jego rozwinięcie w minibufferze i gotowe. Następnie, gdy wpiszesz skrót, zostanie on rozwinięty do wybranego słowa, jeśli po nim pojawi się spacja lub znak interpunkcyjny. Rozszerzenie zostanie również zapisane wielką literą, jeśli skrót zaczyna się wielką literą, więc w naszym przykładzie powyżej „Ouat” zostanie rozwinięte do „Once upon a time”.

Aby zdefiniować globalny skrót w tekście, nad którym aktualnie pracujesz, wpisz `C-x a g` (`add-global-abbrev`). Znaki przed kursorem, aż do początku poprzedniego słowa, staną się rozszerzeniem, jak wskazano w minibufferze. Następnie wpisz skrót i naciśnij `Ret`, aby go zapisać. Funkcja `inverse-add-global-abbrev` definiuje skrót, a następnie rozszerzenie, które aktywujesz za pomocą `C-x a i g`.

Skróty mogą rozszerzać się do wielu słów lub nawet standardowych fragmentów tekstu składających się z wielu akapitów. Zaznacz odpowiedni tekst, użyj `C-x a g`, aby zdefiniować skrót, a następnie naciśnij `C-g`, aby anulować zaznaczenie.

Skróty Emacs są trochę podobne do haseł. Powinny być łatwe do zapamiętania, ale nie identyczne ze słowami występującymi w słowniku. Jednak w przeciwieństwie do haseł nie mogą zawierać znaków interpunkcyjnych.

znaki. Używanie słów ze słownika może prowadzić do frustracji, ponieważ rozszerzają się one do postaci, której nie chcesz. Możesz wykorzystać ten problem na swoją korzyść, definiując często błędnie pisane słowo jako skrót, na przykład rozszerzając „teh” do „the”. Istnieje obejście, jeśli chcesz wyświetlić skrót w tekście. Wpisz C-Q po skrócie i kontynuuj pisanie.

Ujemny argument przedrostka (C-u -) przed dowolnym poleceniem skrótu usuwa je z tabeli. Aby usunąć skrót globalny, użyj C-u - C-x a g, wprowadź wybrany skrót i naciśnij Return.

Aby przejrzeć zbiór skrótów, użyj list-abbrevs. Polecenie to otwiera nowy bufor tylko do odczytu zawierający listę wszystkich skrótów dla każdego odpowiedniego trybu głównego; wyszukaj „global-abbrev-table”. Tabela skrótów może wyglądać mniej więcej tak:

(global-abbrev-table)

„stbl”	3	„wyświetlane w tabeli”
„ouat”	7	„dawno, dawno temu”
„teh”	10	„the”

Ta lista zawiera skróty, numery i odpowiadające im rozszerzenia. Liczba w środku wskazuje, jak często dany skrót był używany, co jest pomocne, jeśli chcesz ograniczyć obszerną kolekcję.

Tabela skrótów jest edytowalna, co pozwala na definiowanie lub modyfikowanie zestawu skrótów. Wpisz nowy skrót między cudzysłowami, a następnie zero i rozszerzenie w cudzysłowie, a po zapisaniu go na dysku będziesz mógł zacząć używać go w swoim manuskrypcie.

Kiedy zapisujesz plik, nad którym pracujesz, a aktualna tabela skrótów nie jest taka sama jak wersja zapisana, Emacs poprosi Cię również o zapisanie zdefiniowanych skrótów w katalogu init. Mechanizm ten gwarantuje, że Twoja kolekcja skrótów oszczędzających czas będzie dostępna również w przyszłych sesjach Emacs.

Podręcznik Emacs zawiera obszerną dokumentację dotyczącą używania i konfigurowania skrótów, w tym zaawansowanych funkcji (zobacz C-h r g abbrev).

Placetoldev text

Lovem Ipsum to pseudolatynski tekst zastępczy używany w projektowaniu stron internetowych, typografii i drukarstwie w celu pokazania, jak strona będzie wyglądać w końcowym produkcie. Tekst pochodzi z traktatu rzymskiego męża stanu i filozofa Cyserona *De Fieibus Boeovum et Malovum* (Skrajności dobra i zła), napisanego w 45 r. p.n.e. Pierwsze dwa słowa (*loven ipsum*) są skrótem od *dolovem ipsum* („sam ból”).

Głównym celem tego tekstu jest skupienie uwagi czytelnika na układzie, typografii i elementach wizualnych, a nie na samej treści. Pakiet ten pozwala zaprojektować układ bez rozpraszania się znaczeniem tekstu. W następnym rozdziale omówiono bardziej szczegółowo typografię i układ dokumentu. Pakiet ten został użyty do zaprojektowania układu wersji papierowej tej książki i jest przydatny podczas testowania funkcji.

EWS zawiera pakiet do generowania akapitów *Lovem Ipsum*, do którego można uzyskać dostęp za pomocą klawiszy C-c w s i. Aby wstawić wiele elementów, należy użyć przedrostka numerycznego. Na przykład M-6 C-c w s wstawia sześć losowych akapitów.

Łatwe tworzenie tekstu

Emacs posiada wbudowane funkcje konwersji słów i regionów na małe i wielkie litery (sekcja 3.8). Podczas pisania prozy czasami wymagamy, aby tytuły książek, rozdziały i inne nagłówki były pisane wielką literą zgodnie z określonymi zasadami.

Pakiet Titlecase autorstwa Case Duckwortha zapewnia wielkie litery w tytułach dla angielskiej prozy. Funkcja titlecase-dwim (C-c w s t) konwertuje zaznaczony fragment lub bieżącą linię/akapit. Nie ma jednej metody wielkich liter w wyrażeniach lub zdaniach. Możesz ustawić konwencję wielkich liter, dostosowując zmienną titlecase-style do preferowanej metody. Menu dostosowywania zawiera rozwijane pole z opcjami ustawiania stylu wielkich liter zgodnie z konwencjami przewodnika stylistycznego APA.

EWS zawiera wygodną funkcję (ews-org-headings-titlecase), która przechodzi przez wszystkie nagłówki w bieżącym buforze Org i zamienia je na wielkie litery. Użyj C-c w s c.

aby uzyskać dostęp do jego funkcji. Dzięki uniwersalnemu argumentowi (C-u) nagłówki stają się wielkimi literami, co oznacza, że tylko pierwsze słowo jest pisane wielką literą. Domyślnie funkcja ta konwertuje wszystkie nagłówki do pożądanego formatu. Konfigurując zmienną `ews-org-heading-level-capitalise`, można dodać liczbę, aby konwertować tylko nagłówki do określonego poziomu, lub zachować domyślne ustawienie konwersji wszystkich nagłówków. Na przykład, gdy ustawisz tę zmienną na jeden, tylko nagłówki najwyższego poziomu zostaną zamienione na wielkie litery, a nagłówki niższego poziomu pozostaną bez zmian. Niestety, ze względu na kaprysy języka angielskiego, niemożliwe jest idealne zamienianie tytułów na wielkie litery za pomocą algorytmu. Na przykład komputer nie rozróżnia słów od akronimów. Dlatego nigdy nie ufaj komputerowi i sprawdzaj swoje tytuły przed publikacją.

A eleae wvitięg ietevfaae

Pisanie kreatywnej prozy wymaga całkowitej koncentracji, a czynniki rozpraszające uwagę są naturalnym wrogiem autora. Komputer jest niezbędnym narzędziem do pisania, ale może również rozpraszać uwagę. Większość programów do pisania jest pełna ikon i opcji umożliwiających zmianę wyglądu dokumentu. Rzadko są one przydatne, a w większości przypadków rozpraszają uwagę. Narzędzia do pisania bez rozpraszających elementów eliminują te czynniki z ekranu, dzięki czemu komputer bardziej przypomina mechaniczną maszynę do pisania, co pozwala autorowi skupić się na treści, a nie na formie.

Olivetti to tryb pomocniczy programu Emacs, który ułatwia pisanie bez rozpraszania uwagi.

Nazwa Olivetti pochodzi od znanej włoskiej marki maszyn do pisania. Ten tryb pomocniczy zmniejsza szerokość tekstu do siedemdziesięciu znaków i wyśrodkowuje tekst w oknie. Szerokość tekstu można zmieniać za pomocą polecenia `olivetti-set-with` (C-c \). EWS ułatwia korzystanie z trybu Olivetti dzięki przyjaznej dla użytkownika funkcji (C-c w o). To polecenie aktywuje tryb Olivetti (`ews-olivetti`) i zapisuje bieżącą konfigurację okna oraz pozycję kursora. Poprawia ono środowisko pisania poprzez zwiększenie rozmiaru tekstu i utworzenie skupionego ekranu. Ponowne aktywowanie funkcji przywraca poprzednie ustawienia okna i kursora, umożliwiając łatwe przełączanie między trybem pisania bez rozpraszania uwagi a trybem normalnym.

Jakość assuuaeee

Chociaż automatyczna kontrola pisowni jest fantastycznym narzędziem zapewniającym poprawność składniową tekstu, czasami trzeba sprawdzić znaczenie słów w słowniku, aby uzyskać więcej kontekstu. Emacs ma wbudowaną funkcję wyszukiwania słów w słowniku, która łączy się ze źródłem internetowym. Domyślnym słownikiem dla EWS jest *Collabovative Ieteveatioeal Dietioeavy of Zeglisł* (CIDE), pochodzący ze słownika 1y13 Webster's Dictionary (dict.org). Treścią tego słownika zarządzają wolontariusze z całego świata, a dostęp do niego można uzyskać za pomocą kilku kliknięć w programie Emacs.

Aby znaleźć definicję słowa, na którym znajduje się obecnie kursor, użyj polecenia dictionary-lookup-definition (C-c w s d). Pojawi się ekran słownika zawierający odpowiednie definicje. Możesz przewijać okno tak jak każdy inny bufor. Bufor słownika zawiera linki do innych zdefiniowanych słów, które można otworzyć za pomocą klawisza Enter. Za pomocą klawiszy n / p można przechodzić między hiperłączami. Aby wyszukać nowe słowo, wpisz s lub kliknij przycisk [Search Definition] u góry okna. Klawisz h wyświetla listę dostępnych opcji w trybie słownika.

Brak słów to częsta bolączka autorów, a lekarstwem na nią jest w większości przypadków słownik synonimów, który pomaga urozmaicić prozę. Polecenie wyszukiwania w słowniku obejmuje również dostęp do słownika synonimów Moby Thesaurus. W trybie słownika wpisz literę D, aby zmienić strategię wyszukiwania i wybrać słownik synonimów.

Podstawową umiejętnością w pisaniu jest dobór odpowiednich słów. Równie ważne jest wiedzieć, których słów nie należy używać. Tryb WriteGood, autorstwa Benjamina Beckwitha, sprawdza tekst pod kątem trzech podstawowych problemów: niejasnych sformułowań, strony biernej i powtórzeń. Tryb Writetood zaznacza problemy w tekście kolorowymi falistymi liniami. Najechnie myszką na zaznaczone słowo powoduje wyświetlenie kontekstu zidentyfikowanego naruszenia.

Politycy i marketerzy często używają niejasnych sformułowań, aby ukryć swoje intencje. Są to niejasne sformułowania, ponieważ pozbawiają język znaczenia, podobnie jak łasica wysysa jaja (Watson, 2004). Zdefiniowaną listę niejasnych sformułowań można znaleźć za pomocą polecenia C-h v writetood-weasel-words. Dostosuj tę zmienną, aby usunąć lub dodać słowa lub utworzyć listę odpowiednią dla swojego języka ojczystego.

Kolejnym obszarem wymagającym poprawy w pisaniu, którego autorzy powinni unikać, jest stosowanie strony biernej. Strona bierna jest jak opowiadanie historii od końca. Zamiast powiedzieć „Pies gonił kota”, co pokazuje, kto co robi, mówisz „Kot był goniony przez psa”. Zdania w stronie biernej mają tendencję do zaciemniania podmiotu wykonującego czynność, przez co zdania są często bardziej rozwlekłe. Taka forma może prowadzić do niejasności, szczególnie w tekstach technicznych i formalnych, gdzie jasność i precyzja mają ogromne znaczenie. Ponadto strona czynna sprawia, że proza jest zazwyczaj bardziej dynamiczna i wciągająca, poprawiając czytelność poprzez podkreślenie podmiotu i jego działań, co dobrze wpisuje się w zasady jasnej komunikacji. Strona bierna ma swoje miejsce w pisaniu, ale w miarę możliwości należy jej unikać (gra słów zamierzona).

Tryb Writegood wykrywa stronę bierną i oznacza ją jako taką. Osiąga to poprzez wykrywanie form czasownika „być” następujących po słowie kończącym się na „ed”. Oprogramowanie zawiera również listę czasowników nieregularnych, którą można przeglądać i modyfikować zgodnie z osobistymi ustawieniami lub własnymi preferencjami.

język poprzez dostosowanie zmiennej writegood-passive-voice-irregulars.

Powtarzające się słowa są częstym efektem kopiowania i wklejania tekstu. Nasz umysł nie jest szczególnie dobry w wykrywaniu powtarzających się słów, ponieważ mózg często pomija słowa i uzupełnia brakujące części (Rayner, Slattery, Drieghe i Liversedge, 2011). Tryb Writegood podkreśla powtarzające się słowa (gra słów zamierzona).

Pakiet WriteGood może również obliczyć wskaźnik łatwości czytania Flescha, aby ocenić, jak łatwy lub trudny do zrozumienia jest tekst w języku angielskim. Wynik mieści się w przedziale od 0 do około 120. Test ten można wykonać za pomocą funkcji writegood-reading-ease (C-c w s r). Dla osób o matematycznym zacięciu, wzór ten oblicza wskaźnik czytelności w następujący sposób:

$$206,835 - 1,015 \frac{\text{słowa}}{\text{zdania}} - 84,6 \frac{\text{sylaby}}{\text{słowa}}$$

Test ten potwierdza to, co intuicyjnie wiemy. Teksty zawierające długie zdania (liczba słów w zdaniu) i długie słowa (liczba sylab w słowie) są trudniejsze do przeczytania. Im niższy wynik czytelności, tym łatwiejszy do zrozumienia jest tekst. Dla porównania, wskaźnik czytelności *Readex's Digest* wynosi około 55, a *Time Magazine* około 52. *Havvard Law Review*

ma ogólny wynik czytelności na poziomie 30 punktów (Lipovetsky, 2023).

Należy pamiętać, że liczenie słów i zdań zależy od założeń dotyczących tego, co stanowi słowo lub zdanie, jak wyjaśniono w sekcji 7.3.

7.4 Zarządzanie projektem pisania

Projekt pisarski to coś więcej niż tylko upchnięcie jak największej liczby słów w dokumencie. Org oferuje pewne funkcje umożliwiające zarządzanie projektem poprzez podzielenie go na kilka plików, śledzenie liczby słów i monitorowanie ogólnego postępu pracy nad manuskrypcem.

Wvitieg lavge pvojeets

Praca z dużymi plikami może być uciążliwa, a w niektórych przypadkach spowalniać działanie Emacs'a, dlatego często dobrym rozwiązaniem jest podzielenie większych projektów na wiele plików. Org może podzielić projekt pisarski na wiele plików z odnośnikami.

Podczas pisania tej książki utworzyłem plik główny oraz plik dla każdego rozdziału. Plik główny zawiera odniesienia do poszczególnych rozdziałów. Na przykład dodanie poniższej linii powoduje włączenie pliku o nazwie `chapter-02.org` do dokumentu w miejscu, w którym ta linia pojawia się podczas eksportu.

```
#+include: „chapter-02.org”
```

Możesz przejść do dokumentu podrzędnego za pomocą `C-c ' (org-edit-special)`, gdy kursor znajduje się na linii włączenia. Org posiada dodatkowe opcje pozwalające określić, która część dokumentu podrzędnego ma zostać włączona. Możesz na przykład wykluczyć linię tytułu włączonego pliku, dodając `:lines "2-"` do słowa kluczowego. Parametr ten instruuje tryb Org, aby włączał tylko tekst od drugiej linii:

```
* Rozdział drugi
```

```
#+include: „chapter-02.org” :lines „2-”
```

Gdy dodasz dokument w ten sposób, poziom nagłówka będzie o jeden niższy od poddrzewa, w którym jest osadzony. Na przykład, gdy dodasz dokument pod nagłówkiem poziomym pierwszego w źródle, nagłówek poziomym pierwszego w połączonym dokumencie stanie się nagłówkiem poziomym drugiego, poziom drugi stanie się poziomem trzecim i tak dalej. Powyższy przykład daje następujący wynik:

* Rozdział drugi
** Nagłówek pierwszego poziomu w „chapter-2.org”
*** Nagłówek drugiego poziomu w „chapter-2.org”

Ta metoda pozwala pracować nad książką lub rozprawą doktorską i przechowywać każdy rozdział w osobnym pliku, tak jak w przypadku tej książki. Podczas eksportowania głównego pliku do ostatecznej publikacji wszystkie zawarte w nim pliki są eksportowane jako jeden plik.

Podczas pracy z wieloma plikami w ramach jednego projektu należy pamiętać o dwóch kluczowych kwestiach. Można dodawać linki do odnośników w innych plikach, ale linki te nie będą działać, dopóki projekt nie zostanie wyeksportowany do jednego pliku lub strony internetowej. Po drugie, korzystając z lokalnej bibliografii, należy upewnić się, że jest ona przywołana we wszystkich plikach, które z niej korzystają, ponieważ pliki, do których się odwołuje, nie dziedziczą właściwości bibliografii.

Counting words

Liczenie słów jest standardową czynnością dla każdego autora. Zamierzam napisać od 5000 do 10 000 słów w każdym rozdziale tej książki. Aby policzyć liczbę słów w zaznaczonej części bufora aktywnego, użyj M-= (count-words-region). Ta funkcja wyświetla liczbę wierszy, zdań, słów i znaków w obszarze echo.

Dodanie argumentu uniwersalnego powoduje zliczenie słów w całym buforze (C-u M-=). Funkcja count-words, która nie ma domyślnego skrótu klawiaturowego, zlicza wszystkie słowa w buforze lub zaznaczonym obszarze. W tym kontekście linia jest linią logiczną, która jest taka sama jak akapit w trybie linii wizualnej.

Liczenie słów nie jest nauką ścisłą, ponieważ wynik zależy od definicji znaku, słowa lub zdania. Podczas liczenia znaków Emacs liczy również

spacje i konstrukcje semantyczne, takie jak metadane pliku Org. Będąc przede wszystkim edytorem kodu komputerowego, Emacs liczy słowa z łącznikami lub słowa oddzielone znakami interpunkcyjnymi jako dwa słowa.

Domyślnie Emacs definiuje zdanie jako sekwencję znaków zakończoną kropką i podwójną spacją. To domyślne ustawienie generuje błędne wyniki podczas liczenia zdań, ponieważ większość autorów używa pojedynczych spacji, więc EWS wyłącza tę funkcję. Dodawanie podwójnych spacji na końcu zdania miało sens w czasach maszyn do pisania. Nowoczesne oprogramowanie do składu tekstu nie wymaga już tej archaicznej praktyki. Większość podręczników stylistycznych, takich jak *The Chicago Manual of Style*, zaleca stosowanie pojedynczych odstępów. Podczas eksportowania tekstu do produktu końcowego oprogramowanie do składu tekstu wstawia odpowiednie odstępy między zdaniami. Podwójne spacje po kropce mają sens, gdy wynik jest w czcionce o stałej szerokości, co jest powszechne w pisaniu kodu oprogramowania. Jedyną wadą tej metody jest to, że skróty takie jak „E. W. S.” są liczone jako wiele słów i zdań.

Aby policzyć liczbę słów w każdym rozdziale lub sekcji tekstu, należy uruchomić funkcję `count-words-region` dla każdej części dokumentu. EWS udostępnia wygodną funkcję, która automatyzuje to zadanie i wyświetla liczbę słów dla każdego nagłówka. Funkcja `ews-org-count-words` (C-c w c) przechodzi przez wszystkie nagłówki i dodaje liczbę słów w szufladzie właściwości, która jest innym rodzajem szuflady działającej podobnie do opisanej powyżej szuflady notatek. Liczba słów dla nagłówków wyższego poziomu obejmuje zawartość nagłówków niższego poziomu. Metoda ta pozwala również na dodanie celów dla każdej sekcji, co umożliwia monitorowanie postępów. Użyj C-c C-x p (`org-set-property`), wpisz TARGET i wprowadź żądaną liczbę słów. Oczywiście można również ręcznie edytować szufladę.

* Nagłówek

:WŁAŚCIWOŚCI:

:LICZBA SŁÓW: 305

:CEL: 300

:KONIEC:

Szuflady właściwości to potężna funkcja, która pozwala przekształcić bufor Org w bazę danych.

Właściwości te można wyświetlić w formie tabeli. Najpierw należy zdefiniować właściwości, które mają być wyświetlane, dodając następujący wiersz na początku bufora Org:

```
#+columns: %40ITEM(Sekcja) %10WORDCOUNT(Słowa) %10TARGET(Cel)
```

Liczby po znaku procentu wskazują rozmiar tej kolumny, a tekst po liczbie odpowiada nazwie właściwości; tutaj ITEM oznacza tekst nagłówka. Tekst w nawiasach jest nazwą wyświetlaną dla kolumny. Teraz możesz wyświetlić liczbę słów i cel dla każdego nagłówka w tabeli za pomocą C-c C-x C-c (org-columns). Upewnij się, że funkcja ta jest wywoływana, gdy kursor znajduje się na najwyższym poziomie hierarchii (na początku dokumentu). Widok ten tworzy nakładkę, której górna linia bufora stanowi nagłówek tabeli.

Wszystkie nagłówki mają szare tło i zawierają wartości zdefiniowanych właściwości. Tabela pojawia się na poziomie przeglądu i zawartości dokumentu poprzez przechodzenie przez dokument za pomocą S-<Tab>.

Nagłówki stają się tylko do odczytu i zawierają właściwości zdefiniowane jako kolumny. Gdy kursor znajduje się na jednym z nagłówków, dostępnych jest kilka opcji. Przycisk c zwiija nagłówki, dzięki czemu widoczna jest tylko tabela, a nie tekst znajdujący się pod spodem. Nadal można edytować tekst, ale tryb wizualny linii jest wyłączony.

Poruszaj się po tabeli za pomocą klawiszy strzałek. Możesz edytować właściwość za pomocą klawisza e. Zmień zawartość w minibufferze i naciśnij Enter. Klawisz g resetuje kolumny. Umieść kursor na nakładce kolumny, aby ją usunąć, i naciśnij q.

Tvaekieg tke status youv wvitiieg

Typowy proces pisania przebiega przez różne etapy, od wczesnych szkiców, przez wersje edytowane, aż po gotowe teksty. Podczas pracy nad różnymi częściami projektu pisarskiego warto znać status każdego rozdziału. Tryb Org zawiera rozbudowany system zarządzania projektami, który można wykorzystać do śledzenia postępów w tworzeniu dokumentu. Niniejsza sekcja stanowi jedynie krótkie wprowadzenie do tej funkcji. Sekcja

y.1 zawiera bardziej szczegółowe wyjaśnienie dotyczące zarządzania projektami.

Każdy nagłówek w trybie Org może mieć token statusu, taki jak TODO, DRAFT lub EDITED, lub dowolny inny preferowany przez Ciebie status. Token statusu dodaje się za pomocą klawiszy Shift i strzałek w lewo/w prawo, gdy kursor znajduje się na nagłówku. Można również użyć skrótu C-c C-t (org-todo). Domyślnie system rozpoznaje tylko statusy TODO i DONE. Tryb Org pozwala jednak na dodanie dodatkowych stanów przepływu pracy, zapewniając elastyczność dostosowania systemu do konkretnych potrzeb. Poniższy przykład instruuje tryb Org, aby przechodził cyklicznie przez te cztery tokeny statusu, ale tylko w tym pliku. Tokeny przed pionową linią (symbolem rury) są w trakcie realizacji i zazwyczaj są zaznaczone na czerwono. Elementy po pionowej linii są ukończone i zaznaczone na zielono.

```
#+todo: TODO PROJEKT EDYCJA | OSTATECZNA
```

Jeśli chcesz dodać status swojego nagłówka do tabeli podsumowującej omówionej w poprzedniej sekcji, dodaj %TODO(Status) lub coś podobnego do definicji kolumny w przedniej części dokumentu.

7.4 Wersje Control i współpraca

Podczas pisania często zdarza się, że zmieniasz kolejność lub strukturę tekstu. Żeby nie zgubić ważnych informacji, warto wiedzieć, jak Emacs zarządza różnymi wersjami bufora lub pliku. Pisanie może wydawać się samotną czynnością, ale często współpracujesz z innymi autorami i redaktorami. W takich sytuacjach kontrola wersji odgrywa kluczową rolę w utrzymaniu ducha współpracy.

Chociaż Emacs nie może pochwalić się efektywnymi systemami współpracy w chmurze, jakie można znaleźć w oprogramowaniu biurowym, oferuje różne metody kontroli wersji. Na najniższym poziomie mamy wersję w bieżącym buforze i wersję zapisaną na dysku. System cofania skrupulatnie śledzi wszystkie zmiany w bieżącym buforze na bardziej szczegółowym poziomie. Inna metoda polega na użyciu wbudowanego systemu kopii zapasowych, który zapisuje starsze wersje plików, zachowując ewolucję manuskryptu. W przypadku bardziej zaawansowanych potrzeb Emacs współpracuje również z oprogramowaniem do kontroli wersji, umożliwiając formalne sprawdzanie plików i wycofywanie ich, co jest przydatne podczas współpracy z innymi autorami lub redaktorem.

oprogramowaniem do kontroli wersji, umożliwiając formalne sprawdzanie plików i wycofywanie ich, co jest przydatne podczas współpracy z innymi autorami lub redaktorem.

Revertig tle euvveet buffev

Zawsze istnieją co najmniej dwie wersje tekstu, nad którym pracujesz. Ostatnio zapisana wersja i bieżący bufor. Możesz odrzucić wszystkie zmiany wprowadzone od ostatniego zapisania bufora za pomocą polecenia revert-buffer, które ponownie ładuje plik z dysku, usuwając wszystkie zmiany wprowadzone od ostatniego zapisania pliku. Używaj tej opcji z rozważą. Przywrócenie bufora ma zastosowanie tylko w przypadku popełnienia poważnych błędów lub aktualizacji pliku poza bieżącą sesją Emacs.

System cofania

W sekcji 3.8 omówiono korygowanie błędów za pomocą potężnego systemu cofania zmian w programie Emacs. Jednak po wielokrotnym wydawaniu poleceń cofania i ponawiania zmian łatwo jest zgubić się w poprzednich stanach dokumentu. Pakiet Vundo autorstwa Yuan fu pomaga śledzić zmiany, wizualizując je w formie drzewa, tworząc w ten sposób maszynę czasu do pisania.

Polecenie vundo (C-M-/) wizualizuje różne zmiany w pliku w postaci poziomego drzewa w minibufferze. Funkcja ta pozwala na poruszanie się po poprzednich wersjach za pomocą klawiszy strzałek. Bieżący bufor zmienia się w miarę przeglądania historii. Naciśnij Enter, aby wybrać wybraną zmianę i kontynuować pisanie. Klawisze C-g pozwalają wyjść z drzewa bez wprowadzania zmian.

W większości przypadków drzewo jest po prostu linią prostą. W niektórych przypadkach dokument ma równoległe wersje wizualizowane jako gałęzie podczas łączenia poleceń cofania i ponawiania. Drzewo Vundo dla prostego przykładu na rysunku 3.4, gdzie zaczęliśmy od Sokratesa, zmieniliśmy na Platona i z powrotem, a następnie dodaliśmy trochę tekstu, wyglądałoby tak:

```
o--o--o--o
  '--o
```

Pakiet Vundo zapewnia intuicyjny interfejs do zarządzania różnymi dokumentami od momentu otwarcia pliku. Podręcznik pakietu Vundo zawiera dodatkowe informacje, które można przeczytać, wpisując C-h P i wybierając vundo.

Automatyczne kopie zapasowe

Przepisanie pliku powoduje utratę jego poprzedniej zawartości, co czasami może oznaczać utratę wielu godzin pracy w ciągu ułamka sekundy. Aby zapobiec takim katastrofom, Emacs może przechowywać kopię zapasową każdego pliku.

Emacs tworzy kopię zapasową pliku przy pierwszym dodaniu treści do bufora. Pierwsza kopia zapasowa każdego nowego pliku jest zatem plikiem pustym. Bez względu na to, jak często zapisujesz plik w bieżącej sesji, jego kopia zapasowa pozostaje niezmieniona, dopóki nie usuniesz bufora i nie powrócisz do pliku. Pliki kopii zapasowych zawierają więc wersje sprzed rozpoczęcia nowej sesji pisania. Ta kopia zapasowa pozostanie taka sama jak bieżący plik, dopóki nie zapiszesz ponownie bufora. Aktualna wersja stanie się plikiem kopii zapasowej, jeśli bufor zostanie zapisany z argumentem prefiksowym (C-u C-x C-s).

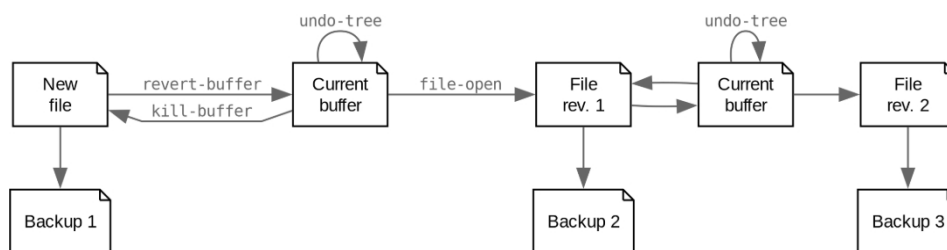
Domyślnie Emacs przechowuje pliki kopii zapasowych w tym samym katalogu co plik oryginalny, co powoduje bałagan. EWS przechowuje kopie zapasowe w katalogu konfiguracyjnym Emacs'a w folderze backups. Emacs dodaje do oryginalnej nazwy pliku tyldę, aby zaznaczyć, że jest to kopia zapasowa, więc kopia zapasowa pliku chapter-02.org będzie miała nazwę chapter-02.org~. EWS jest również skonfigurowany tak, aby zachowywać trzy ostatnie wersje pliku. Emacs dodaje numery wersji na końcu nazwy pliku: ~1, ~2 itd.

Edytor katalogów (dired) umożliwia przeglądanie dostępnych plików kopii zapasowych. Użyj skrótu C-x d i wprowadź lokalizację folderu kopii zapasowych (w katalogu konfiguracyjnym Emacs'a). Za pomocą klawiszy strzałek i Enter wybierz i otwórz plik. Rozdział y zawiera szczegółowe wyjaśnienie sposobu korzystania z edytora katalogów.

Sposób, w jaki Emacs zarządza kopiami zapasowymi, może być mylący, więc spróbujmy to zwizualizować (rysunek 7.1). Kiedy tworzysz nowy plik i zaczynasz go edytować, system tworzy kopię zapasową, która na tym etapie jest pustym plikiem lub zawiera zawartość pliku w stanie, w jakim został on po raz pierwszy otwarty w Emacsie. Podczas gdy

edytowanie i zapisywanie wersji pośrednich, nowe kopie zapasowe są tworzone tylko wtedy, gdy plik zostanie zapisany z argumentem uniwersalnym. Po zamknięciu bufora i ponownym jego otwarciu tworzona jest nowa kopia zapasowa. Pakiet Undo-Tree zarządza wersjami w otwartym buforze. Te wersje pośrednie są usuwane po zamknięciu (lub usunięciu) bufora.

Mechanizmy te zapewniają precyzyjną kontrolę wersji, która minimalizuje ryzyko utraty informacji niemal do zera. Aby dowiedzieć się więcej o systemie kopii zapasowych Emacs, przeczytaj odpowiedni wpis w podręczniku Emacs, używając polecenia `C-h r g backup`.



Rysunek 7.1: Logika systemu kopii zapasowych Zmaes.

Porównanie wersji plików

W miarę postępów w pisaniu projektu możesz otrzymać różne wersje tego samego pliku, stworzone przez Ciebie, edytora lub innego współpracownika. Sytuacja ta może stanowić problem, ponieważ nie masz pewności, który plik jest najnowszą wersją, lub może zająć potrzeba utworzenia nowej wersji zawierającej wszystkie najnowsze zmiany.

Polecenie ediff pomaga rozwiązać ten problem. Zapewnia bogaty interfejs do porównywania dwóch lub trzech plików i zarządzania zmianami. Ediff wizualizuje różnice między plikami i pozwala wybrać części każdego pliku, które chcesz scalić do głównej wersji.

Korzystanie z Ediff jest proste. Uruchom polecenie ediff i wybierz dwa pliki w minibufferze, które zostaną następnie wyświetlone w ramce Emacs w dwóch oknach obok siebie. Ediff odnosi się do lewego okna jako pliku A, a do prawego jako pliku B. Panel sterowania Ediff

znajduje się poniżej dwóch okien plików. Panel sterowania umożliwia wydawanie poleceń do jednego z dwóch otwartych buforów, co sprawia, że proces porównywania i scalania plików jest bardzo prosty.

Po wpisaniu litery n program Ediff przechodzi do następnej różnicy. Akapit, w którym występuje różnica, jest podświetlony, a rzeczywiste różnice są zaznaczone na bardziej intensywnym tle. Wielokrotne wpisywanie litery n powoduje przejście do kolejnych różnic, a litery p — do poprzedniej. Wiersz trybu panelu sterowania wyświetla liczbę różnic i postęp w ich przeglądaniu. Podświetlone sekcje są w rzeczywistości śledzonymi zmianami używanymi w oprogramowaniu do edycji tekstu.

Ediff oferuje elastyczne podejście do obsługi różnic. Możesz zsynchronizować części pliku A z plikiem B lub odwrotnie, wpisując a lub b w panelu sterowania. Użycie a synchronizuje dwa pliki z zawartością pliku A dla zaznaczonej sekcji, natomiast użycie b sprawia, że oba pliki stają się takie same jak zaznaczenie w pliku B. Elastyczność Ediff pozwala wybrać wersję pliku, która najlepiej odpowiada Twoim potrzebom, dając Ci pełną kontrolę nad procesem scalania.

Możesz również przesunąć kursor do jednego z buforów plików, aby edytować je ręcznie, tak jak zwykle. Może to jednak powodować zamieszanie, ponieważ trudno będzie odróżnić wpisany tekst od oryginalnej treści. Żaden tekst dodany podczas sesji Ediff nie jest rozpoznawany jako nowa różnica. Aby zakończyć sesję, wpisz q w panelu sterowania i postępuj zgodnie z instrukcjami, aby zapisać lub usunąć dwa bufor.

Ediff jest przydatny podczas pracy z redaktorem (osobą, która redaguje tekst, a nie programem komputerowym). Możesz wysłać mu plik tekstowy, który redaktor poprawi, a następnie przetworzyć go za pomocą Ediff. Po otrzymaniu zmodyfikowanego pliku możesz użyć Ediff do zarządzania wszelkimi proponowanymi zmianami w tekście. Jeśli Twój redaktor nie czuje się komfortowo podczas pracy z plikami tekstowymi, w następnym rozdziale wyjaśniono, jak eksportować pliki Org do popularnych formatów edytorów tekstu.

Ediff posiada wiele funkcji wykraczających poza zakres niniejszej książki. Aby wyświetlić listę opcji, należy wpisać znak zapytania, gdy kursor znajduje się w panelu sterowania. Kompleksowy opis funkcji Ediff można znaleźć w instrukcji obsługi, dostępnej za pomocą polecenia `ediff-documentation` lub `C-h R ediff`.

Vevsioe eoetvol

Tradycyjna metoda kontroli wersji, często stosowana przez autorów, polega na zmianie nazwy pliku w celu rozróżnienia różnych wersji tekstu. Na przykład, możesz mieć plik o nazwie `article-draft.org` i niemal identyczne kopie tego pliku o nazwach `article-draft-02.org` i tak dalej. Chociaż takie podejście może być praktyczne, gdy wszyscy współpracownicy stosują ten sam proces, może stać się uciążliwe i mylące. Dzięki możliwościom programu Ediff można łatwo śledzić zmiany między plikami, jednak podejście to może być uciążliwe.

Używanie wersji nazw plików powoduje zaśmieszenie katalogu projektu wieloma kopiami plików. System kopii zapasowych Emacs eliminuje potrzebę przechowywania wielu wersji tego samego pliku. Jednak system kopii zapasowych zapewnia ograniczoną aktywną kontrolę nad przechowywanymi wersjami. Programiści często współpracują z innymi programistami. Rozwiązują problemy między wersjami tego samego kodu za pomocą systemu kontroli wersji (VCS). Narzędzia te są nie tylko odpowiednie dla hakerów, ale przynoszą również korzyści autorom piszącym teksty dla ludzi.

System kontroli wersji (VCS) jest jak superwydajna szafka na dokumenty. To Twój osobisty asystent, który śledzi każdą zmianę w dokumencie, pozwala wrócić do poprzedniej wersji, a nawet umożliwia pracę nad wieloma wersjami roboczymi bez ryzyka pomyłki. Narzędzie to może również informować Cię o tym, kto i kiedy wprowadził zmiany, ułatwiając współpracę.

Systemy kontroli wersji mogą również definiować różne gałęzie Twojej pracy. Pomyśl o rozgałęzianiu jako o tworzeniu równoległych wszechświatów. Możesz pracować nad różnymi wątkami fabularnymi bez ich mieszania. Gdy wszystko będzie gotowe, możesz scalić zmiany z głównym dokumentem. Załóżmy, że piszesz powieść. Możesz utworzyć główną gałąź dla bieżącego szkicu. Kiedy piszesz alternatywne zakończenie, możesz utworzyć nową gałąź o nazwie „alternatywne zakończenie” i pracować nad nią bez zakłócania głównego szkicu. Jeśli współpracujesz z redaktorem, może on wprowadzać zmiany i sugerować poprawki w swojej gałęzi. Możesz przejrzeć i scalić te zmiany z głównym szkicem, gdy będą gotowe.

Jeśli jesteś współautorem książki lub otrzymujesz informacje zwrotne od redaktora, system kontroli wersji (VCS) może zapobiec konfliktom między wersjami. Możesz sprawdzić, kto wprowadził dane

zmiany i rozwiązać wszelkie potencjalne konflikty, gdy dwie osoby edytują tę samą część dokumentu.

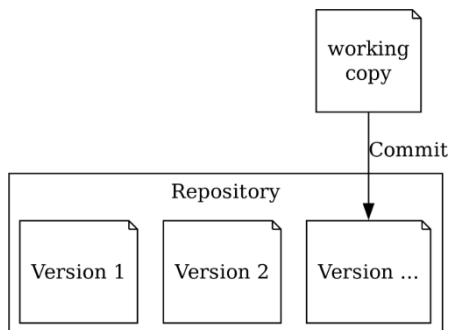
Ponieważ system VCS zapisuje wersje dokumentu, służy również jako niezawodna kopia zapasowa. Możesz odzyskać swoją pracę w przypadku awarii komputera lub przypadkowego usunięcia czegoś. Ponadto narzędzia VCS mogą przechowywać komentarze dotyczące konkretnych zmian. Jest to przydatne, aby przypomnieć sobie, dlaczego wprowadzono daną zmianę, lub w celu komunikacji ze współpracownikami. System VCS porządkuje burzliwy proces pisanie i gwarantuje, że nigdy nie stracisz świetnego pomysłu lub krytycznej poprawki.

Wbudowany pakiet VC programu Emacs może współpracować z najpopularniejszymi systemami kontroli wersji, z których najpopularniejszym jest Git. Aby ta funkcja działała, należy zainstalować Git, zdecentralizowany system kontroli wersji opracowany przez Linusa Torvaldsa w celu wsparcia rozwoju systemu Linux.

Załóżmy, że pracujesz nad projektem zawierającym wiele plików Org i kilka ilustracji w katalogu i podkatalogach. Katalog plików objęty kontrolą wersji nazywany jest w żargonie hakerów repozytorium lub repo. Zacznij od zainicjowania bieżącego katalogu jako repozytorium za pomocą polecenia `vc-next-action` (`C-x v v`).

To polecenie wykrywa następną logiczną czynność, którą w tym przypadku jest utworzenie nowego repozytorium. Należy wybrać backend (Git) oraz folder, w którym ma zostać wdrożona kontrola wersji. VC przechowuje dane kontroli wersji w ukrytym folderze w katalogu projektu. W przypadku Git jest to `.git`. Ręczne zmiany w tym katalogu mogą zakłócić kontrolę wersji, dlatego należy pozostawić go bez zmian. Należy pamiętać, że system kopii zapasowych Emacs ignoruje pliki w katalogach zarządzanych w ramach kontroli wersji, aby zapobiec powielaniu.

Kolejnym logicznym krokiem w tym procesie jest zatwierdzenie pliku do repozytorium. Zatwierdzenie pliku do repozytorium Git oznacza zapisanie migawki aktualnego stanu pliku w historii repozytorium (rysunek 7.2). Kiedy zmieniasz plik w repozytorium i zatwierdzasz te zmiany, Git rejestruje je jako nowe zatwierdzenie w historii repozytorium. Każde zatwierdzenie ma unikalny identyfikator i zawiera informacje o zmianach, w tym nazwisko osoby, która wprowadziła zmianę, a także datę i godzinę jej wprowadzenia. Plik zatwierdza się za pomocą tego samego polecenia (`C-x v v`).



Rysunek 7.2: System Vesioe Coetvol.

VC zatwierdza bieżącą wersję pliku i prosi o opisanie zmian w krótkim oświadczeniu zawierającym nie więcej niż pięćdziesiąt znaków. Pod podsumowaniem (pod poziomą linią) można wpisać bardziej szczegółowy opis zmian, ale nie jest to obowiązkowe. Podsumowanie zmian stanowi bieżący komentarz dotyczący ewolucji manuskryptu. W dolnej części ekranu VC wyświetla listę plików objętych zatwierdzeniem. Zatwierdzenie finalizuje się za pomocą C-c C-c. Jeśli użytkownik uzna, że nie jest jeszcze gotowy do zatwierdzenia zmian, powinien użyć C-k.

Pasek trybu pliku wyświetli teraz wskaźnik informujący, że plik jest objęty kontrolą wersji, wraz z gałęzią, do której należy, zazwyczaj Git: master. Oznacza to, że używasz Git do zarządzania wersjami tego pliku i pracujesz w gałęzi master. Polecenie vc-diff (C-x v =) wyświetla wyskakujące okienko, które wizualizuje różnice między bieżącą wersją a ostatnim commitem.

Możesz wygenerować listę bieżących zmian w plikach za pomocą C-x v l lub vc-print-log. Lista ta zawiera unikalny identyfikator zatwierdzenia, autora, datę zmiany oraz podsumowanie zmian, z najnowszą wersją u góry. Przechodź między różnymi wersjami za pomocą klawiszy n (następna) i p (poprzednia). Zmiany między wersjami można wyświetlić za pomocą klawisza d. Wybranie więcej niż jednego zatwierdzenia za pomocą klawiszy m i strzałek, a następnie klawisza d powoduje wyświetlenie różnic między najstarszą a najnowszą wersją. Aby zamknąć ten widok, należy użyć klawisza q.

Aby wyświetlić zmiany w całym repozytorium, użyj `C-x v L` (`vc-print-root-log`).

Istnieje również polecenie umożliwiające wyświetlenie historii rozwoju wybranego fragmentu tekstu. Wybierz interesujący Cię fragment tekstu i użyj `C-x v h` (`vc-region-history`). Bufor ten działa w taki sam sposób, jak w dwóch poprzednich przykładach. Polecenie `vc-annotate` (`C-x v g`) wyświetla odpowiednie zatwierdzenie dla każdego wiersza tekstu, pokolorowane zgodnie z wiekiem wkładu.

Polecenie `vc-dir` (`C-x v d`) wyświetla status wszystkich plików w bieżącym katalogu i jego podkatalogach. Pierwsza linia pokazuje backend dla tego katalogu kontrolowanego wersją, który w naszym przypadku jest Git. Kolejne cztery linie pokazują katalog pod kontrolą wersji i inne metadane. Ten bufor pozwala na działanie na poszczególnych plikach. Naciśnij klawisz z znakiem zapytania, aby wyświetlić listę dostępnych opcji.

Kontrola wersji ma znacznie więcej funkcji, a ich pełne wyjaśnienie wykracza poza zakres tej książki, np. tworzenie oddzielnych gałęzi pracy i synchronizacja repozytorium z wersją online w celu udostępnienia jej współpracownikom. Rozdział podręcznika Emacs'a poświęcony kontroli wersji można znaleźć za pomocą polecenia `C-h r g version`. Podręcznik został napisany z myślą o tworzeniu oprogramowania, więc proszę użyć wyobraźni, aby zobaczyć, jak można go zastosować do pisania prozy.

Współpraca z innymi autorami i redaktorami

Niestety, pisanie zwykłego tekstu przy użyciu zaawansowanych systemów kontroli wersji nie jest najczęstszym podejściem do współpracy przy projektach pisarskich. Rodzi to pytanie, w jaki sposób współpracować z współautorami lub redaktorami. Istnieją trzy opcje, współpracownicy mogą:

1. Nie używać systemu kontroli wersji ani plików tekstowych
2. Nie używają systemu VCS, ale piszą w zwykłym tekście
3. Korzystają zarówno z VCS

Czwarty możliwy scenariusz, w którym ktoś korzysta z systemu VCS, ale nie używa plików tekstowych, nie ma większego sensu. Omówmy pokrótce każdy z tych trzech scenariuszy.

Nawet jeśli Twój współpracownik nie korzysta z systemu kontroli wersji (VCS) lub nie rozumie zalet zwykłego tekstu, nadal możesz efektywnie współpracować. W następnym rozdziale dowiesz się, jak eksportować swoją pracę do najpopularniejszych formatów plików edytorów tekstu (sekcja 8.3). Dzięki temu będziesz mógł wysłać swój manuskrypt do redaktora lub innego współpracownika. Po otrzymaniu wyniku możesz łatwo zaakceptować wszystkie śledzone zmiany, zapisać plik jako plik tekstowy i użyć programu Ediff do zarządzania różnicami w stosunku do ostatniej wersji. Proces ten gwarantuje, że nadal możesz uczestniczyć w procesie wspólnego pisania, niezależnie od narzędzi używanych przez współpracowników.

Twoi współpracownicy nie muszą koniecznie korzystać z programu Emacs, ponieważ każdy edytor tekstu może odczytywać i zapisywać pliki Org. Jednak zaawansowane funkcje Org nie są dostępne, jeśli nie korzystasz z programu Emacs. Gdy współpracownik wyśle Ci nową wersję pliku, idealnym narzędziem do rozstrzygania różnic między wersjami jest Ediff.

Gdy wszyscy współpracownicy korzystają z systemu kontroli wersji (VCS) i potrafią pisać w trybie Org, zarówno w programie Emacs, jak i innym edytorze tekstu, należy przesłać repozytorium do platformy internetowej, takiej jak GitLab. Przesłanie repozytorium do pamięci online nazywa się „pushowaniem repozytorium”. Użytkownicy mogą następnie „sklonować” repozytorium online i pracować nad manuskrytem. Współpracownicy mogą przysyłać swoje zmiany do centralnego repozytorium. Właściciel repozytorium może następnie scalić zmiany z główną gałęzią i rozwiązać potencjalne problemy.

Inną metodą współpracy jest udostępnienie folderu projektu za pomocą systemu udostępniania, takiego jak NextCloud. Metoda ta wiąże się z ryzykiem, że oboje otworzycie ten sam plik w tym samym czasie. Emacs może zablokować plik przed otwarciem przez innego użytkownika, ale konfiguracja EWS wyłącza możliwość blokowania plików. Aby korzystać z folderu współdzielonego, należy dostosować zmienną `create-lockfiles`. Jeśli jest ona ustawiona na `t`, plik może być otwarty tylko przez jednego użytkownika jednocześnie. Osiąga się to poprzez zapisanie pliku, który uniemożliwia innym użytkownikom wprowadzanie zmian.

7.5 Inne tryby tekstowe

Ta książka stanowi kompleksowy przewodnik po korzystaniu z trybu Org podczas pisania. Org jest tylko jednym z wielu trybów tekstowych dostępnych w programie Emacs. Zrozumienie innych formatów opartych na trybie tekstowym ma kluczowe znaczenie dla podejmowania świadomych decyzji i usprawnienia procesu pisania.

Najbardziej podstawową wersją jest plik tekstowy, zazwyczaj oznaczony rozszerzeniem .txt. Pliki te są proste, ponieważ nie zawierają żadnego formatowania i zazwyczaj składają się wyłącznie z znaków alfanumerycznych, spacji i znaków interpunkcyjnych. Jeśli chcemy opublikować dzieło w formie strony internetowej, książki lub innego rodzaju mediów, plik tekstowy nie będzie wystarczający, ponieważ nie ma możliwości zdefiniowania, jak powinien wyglądać efekt końcowy, np. układ strony, rodzaje czcionek, hiperłącza i inne istotne elementy opublikowanego dzieła.

Inne tryby tekstowe obejmują tekst sformatowany lub tekst sformatowany. Pliki te zawierają zwykły tekst oraz dodatkowe informacje dotyczące projektu dokumentu, takie jak styl czcionki i linki. Org i HTML są przykładami sformatowanego zwykłego tekstu. Instrukcje dotyczące stylizacji są oznaczeniami dokumentu. Tradycyjne oznaczenia wydawnicze to system adnotacji czerwonym lub niebieskim ołówkiem, które instruują drukarza, jak stylizować rękopis. Oznaczanie dokumentu było pracochłonne, a redaktorzy i zecerzy używali symboli (oznaczeń) do wskazania, jak tekst powinien wyglądać na stronie. W publikacjach cyfrowych używamy sekwencji znaków i znaków interpunkcyjnych jako oznaczeń, aby poinstruować komputer, jak wyświetlać dokument.

Graficzne edytory tekstu ukrywają znaczniki i wyświetlają tekst w formie drukowanej. Metoda ta może wydawać się wygodna, ale może również stać się koszmarem, gdy próbujesz zmusić system do uzyskania pożądanego rezultatu przy użyciu tych niewidocznych instrukcji. Zwykły tekst jest łatwiejszy w użyciu, ponieważ znaczniki są bezpośrednio widoczne w dokumencie, dzięki czemu masz bezpośrednią kontrolę nad wyglądem swojego manuskryptu.

Istnieje wiele trybów zwykłego tekstu przeznaczonych do różnych celów. Widzieliście już, jak Org wykorzystuje fragmenty zwykłego tekstu do dodawania instrukcji. Istnieją dwa rodzaje znaczników. Znaczniki prezentacyjne zawierają instrukcje dotyczące sposobu prezentacji tekstu, w tym pogrubienia, kursywy, listy i nagłówki. Znaczniki proceduralne składają się z symboli, które instruują komputer

komputerowi o takich aspektach, jak rozmiar strony, pozycja tekstu, cytaty, metadane i inne bardziej złożone elementy publikacji (Travis & Waldt, 1yy5).

Tryby tekstu stylizowanego są dostępne w dwóch rodzajach: zwykłym znacznikowym i uproszczonym. Zwykły język znacznikowy, taki jak HTML lub LaTeX, zawiera instrukcje przypominające język komputerowy, które definiują wygląd dokumentu. Na przykład, aby napisać nagłówek w HTML i LaTeX, należy użyć:

- HTML: `<h2>To jest nagłówek</h2>`
- LaTeX: `/section{To jest nagłówek}`

Standardowe języki znaczników oferują zaawansowane możliwości definiowania wszystkich szczegółów końcowego wyniku projektu. Wadą jest to, że tekst jest pełen nawiasów kątowych, nawiasów klamrowych i instrukcji. W wersjach uproszczonych liczba znaków wymaganych do zdefiniowania dokumentu jest mniejsza, co upraszcza proces pisania. Org jest przykładem lekkiego języka znaczników. Nie jest on lekki ze względu na ograniczone możliwości, ale raczej ze względu na uproszczony zestaw instrukcji. Aby utworzyć ten sam nagłówek w Org, należy dodać gwiazdkę na początku linii, co pozwala usunąć część bałaganu z ekranu.

W kolejnych dwóch sekcjach wyjaśniono zasadę działania dwóch standardowych trybów tekstowych używanych przez autorów. W następnym rozdziale opisano, jak eksportować manuskrypty Org w celu utworzenia e-booka, pliku PDF lub książki drukowanej albo jak eksportować je do LaTeX lub HTML.

letvodueieg Mavkdowe

Markdown, język znaczników wprowadzony przez pionierów internetu Johna Grubera i Aarona Swartza w 2004 roku, został zaprojektowany tak, aby był łatwy do odczytania i zrozumienia, przy minimalnym użyciu znaków semantycznych. W przeciwieństwie do bardziej złożonego HTML, prostota Markdown jest jego kluczową cechą, stąd nazwa „mark-down”. Jest to wszechstronne narzędzie, szeroko stosowane w komunikatorach internetowych, forach internetowych i dokumentacji oprogramowania. Podstawowe zasady Markdown są podobne do zasad Org, jak pokazano poniżej.

```
# Nagłówek
## Podtytuł
Atrybuty tekstu: kursywa, pogrubienie, „monospace”.
```

Listy punktowane zagnieżdżone w listach numerowanych:

- ```
1. Owoce
 * Jabłko (podlisty wcięte o cztery spacje)
 * Banan
2. Warzywa
 - Marchew
 - Brokuły
```

```
Link [link](http://example.com).
```

```
Obraz [Image](Icon-pictures.png
„icon”)
```

Istnieje wiele odmian Markdown, z których większość zapewnia dodatkową funkcjonalność w stosunku do standardowej składni. Pakiet Markdown Mode autorstwa Jasona R. Blevinsa implementuje oryginalną wersję. Twórca opublikował również szczegółową książkę na temat korzystania z Markdown w Emacsie (Blevins, 2017). Konfiguracja EWS domyślnie aktywuje Markdown, ale pełny opis tego formatu wykracza poza zakres niniejszej książki.

Markdown jest często używany w dokumentacji technicznej i jest bardziej powszechny niż tryb Org do udostępniania informacji.

Pakiet Denote umożliwia tworzenie notatek w Markdown w dwóch odmianach. W przeciwieństwie do Org, Markdown nie posiada natywnych funkcji przechowywania metadanych dotyczących dokumentu. Denote oferuje dwie metody osiągnięcia tego celu: TOML (Tom’s Obvious Minimal Language) lub YAML (YAML Ain’t Markup Language). Aby rozpocząć tworzenie notatek Markdown, należy ustawić zmienną `denote-file-type` na `markdown-toml` lub `markdown-yaml`.

Więcej szczegółów znajdziesz w instrukcji obsługi Denote. Wypróbuj różne opcje samodzielnie. Możesz również rozważyć użycie pakietu Denote-Markdown, który zapewnia dodatkowe funkcje do pisania plików Markdown Denote.

*Seveewitieg witł Fouetaie*

Kto nie chciałby napisać scenariusza do kolejnego hollywoodzkiego lub bollywoodzkiego hitu? Pisanie scenariuszy filmowych lub teatralnych podlega ścisłym zasadom i regułom formatowania. Standardowa czcionka dla scenariuszy ma stały rozstaw znaków, nadając dokumentowi wygląd starej maszyny do pisania.

Możesz jednak uwolnić swoją kreatywność dzięki fountain, prostemu formatowi tekstowemu do pisania scenariuszy w dowolnym edytorze tekstu. Format plików fountain jest dość wyjątkowy, ponieważ nie zawiera prawie żadnych znaczników. Biorąc pod uwagę ścisłe konwencje obowiązujące w scenariuszach, fountain może logicznie określić sposób formatowania dokumentu.

Tryb fountain implementuje ten format tekstu w Emacs i jest włączony w *Zmaes Wvit-ieg Studio*. Aby stać się następnym Szekspirem lub Stanleyem Kubrickiem, przeczytaj obszerny podręcznik za pomocą C-h R fountain.



---

## **Publikacja: Podziel się ze światem**

---

Tryb Emacs Org to nie tylko idealne narzędzie do pisania bez rozpraszania uwagi; to potężne narzędzie do publikowania. Praca z czystym tekstem pozwala skupić się na treści, a nie na wyglądzie. Dzięki możliwościom eksportu Org Twoja praca może istnieć w różnych formatach. Org może eksportować manuskrypt do formatu PDF w celu wydrukowania książki, artykułu do czasopisma, strony internetowej lub dokumentu edytora tekstu. Dodatkowe pakiety umożliwiają eksportowanie manuskryptu do innych formatów, takich jak e-book w formacie ePub.

W tym rozdziale opisano sposób przygotowania plików Org do eksportu poprzez zdefiniowanie układu i typografii. W pierwszej części wyjaśniono zasady procesu eksportu oraz sposób, w jaki Org współpracuje z innym oprogramowaniem w celu uzyskania pożądanego wyniku. W drugiej części przedstawiono ogólne ustawienia mające zastosowanie do wszystkich formatów wyjściowych, wyposażając czytelnika w praktyczną wiedzę niezbędną do przygotowania dokumentu do publikacji. W pozostałej części rozdziału omówiono zawartość najpopularniejszych formatów eksportu, takich jak PDF, HTML i dokumenty edytorów tekstu. Zawiera on wskazówki dotyczące konfiguracji programu Emacs i manuskryptu w celu uzyskania pożądanego wyniku.

## 8.1 Eksportowanie plików Org

Przygotowanie dokumentu do eksportu należało kiedyś do obowiązków zecerów. Proces składu tekstu określa rozmiary czcionek i elementów graficznych, a także ich rozmieszczenie na stronie. W tradycyjnym drukarstwie skład tekstu polega na układaniu fizycznych ruchomych czcionek w celu utworzenia strony. Dzięki publikacjom elektronicznym nie musimy już przenosić fizycznych elementów, ale wydajemy polecenia. Skład tekstu w Org rozpoczyna się od dodania odpowiednich metadanych do manuskryptu, w zależności od wybranego formatu eksportu.

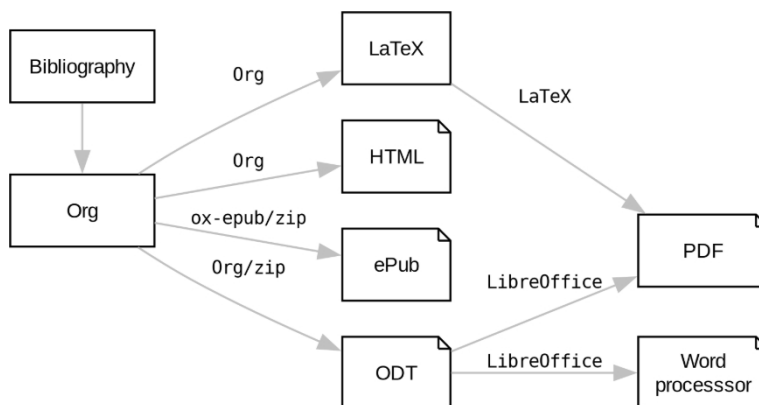
Podstawową zasadą eksportowania plików Org do żadanego formatu jest to, że Emacs konwertuje tekst i kojarzy go z klasą dokumentu, arkuszem stylów CSS lub innym typem szablonu. Szablon określa typografię i układ dokumentu. Nie musisz znać CSS, HTML ani LaTeX, ale znajomość tych języków pomoże Ci dostosować wynik. Twój plik tekstowy może również zawierać linki do lokalnej lub globalnej bibliografii, aby zarządzać cytatami.

Podczas eksportowania do formatu PDF lub edytora tekstu, wyeksportowany plik jest przekazywany do programu LaTeX lub LibreOffice, a następnie kompresowany za pomocą programu Zip w celu uzyskania ostatecznego wyniku. Program Zip jest niezbędny, ponieważ dokumenty ePub i Office są stronami internetowymi spakowanymi w jeden plik. Rysunek 8.1 przedstawia przebieg eksportu Org oraz narzędzia używane do renderowania danych wyjściowych z pliku Org do pożądanego formatu wyjściowego. Ten sam plik można wyeksportować do dowolnego z dostępnych formatów. Na przykład pliki źródłowe tej książki tworzą gotowy do druku plik PDF dla wersji papierowej, e-book w formacie ePub oraz stronę internetową.

Na przykład, jeśli tworzysz stronę internetową lub e-book, każdy tekst Org otoczony ukośnikami (`/example/`) przekształca się w `<i>example</i>`, natomiast w przypadku LaTeX staje się

`\emph{example}`. Domyślnym ustawieniem dla tego przykładu jest tekst pisany kursywą, ale może to być coś innego, w zależności od tego, jak skonfigurujesz eksport.

Konfiguracja Org i Emacs zawiera instrukcje dotyczące układu i typografii dokumentu (tzw. „co masz na myśli” w WYSIWYM). Każdy format eksportu ma własną metodę łączenia składni z typografią. Podczas eksportowania do HTML, kaskadowy arkusz stylów (CSS) określa wygląd, natomiast w LaTeX klasa dokumentu i preambuła decydują o tym, jak ta składnia wygląda w końcowym wyniku. Szablon tekstowy OpenDocument



Rysunek 8.1: Przykłady eksportu w trybie Org.

(OTT) określa ostateczny wygląd podczas eksportowania do formatu edytora tekstu.

Eksportowanie pliku Org rozpoczyna się od funkcji `org-export-dispatch`, której domyślnym skrótem klawiaturowym jest C-c C-e. Pojawia się nowy bufor z dużą liczbą opcji. Pierwsza wpisana litera określa format eksportu, a kolejne litery określają konkretne opcje. Użycie `q` powoduje zamknięcie programu eksportującego.

## 8.2 Ustawienia dokumentu

Twój manuskrypt może przechowywać metadane i konfiguracje, które pomagają w procesie eksportowania. Liczne ustawienia kontrolują sposób eksportowania dokumentu przez Org. Wykorzystanie tych danych zależy od formatu eksportu. Niektóre ogólne metadane mają zastosowanie do wszystkich formatów. Następujące słowa kluczowe mają zastosowanie do wszystkich formatów:

- `#+title`: Tytuł dokumentu.
- `#+author`: Nazwisko autora.
- `#+date`: Data publikacji.

Możesz zdefiniować kolejność roku, miesiąca i dnia zgodnie z własnymi preferencjami za pomocą zmiennej `org-export-date-timestamp-format`. Zmienna ta wykorzystuje kodowanie powszechnie stosowane w językach programowania do wyświetlania daty. Na przykład `%e %B %Y` oznacza, że zaczyna się od numeru dnia (bez zer wiodących), oznaczonego przez `%e`, następnie pełnej nazwy miesiąca (`%B`), a następnie czterocyfrowego roku (`%Y`), oddzielonych spacjami (3 czerwca 2024 r.). Aby użyć amerykańskiego formatu daty, dostosuj tę zmienną do `„%B”`. `%e, %Y` (3 czerwca 2024 r.). Strona pomocy dla zmiennej `format-time-string` zawiera listę dostępnych formatów.

Nazwisko autora pojawia się domyślnie w każdym eksportowanym dokumencie, nawet jeśli nie zostało zdefiniowane w pliku `Org`. Gdy to słowo kluczowe nie jest używane, eksport domyślnie przyjmuje wartość zmiennej `user-full-name`. Dodanie `#+options: author:nil` neguje to zachowanie. Ta opcja może również usunąć tytuł, autora i/lub datę.

```
#+options: author:nil date:nil title:nil
```

`Org` posiada precyzyjnie dostrojony mechanizm kontroli eksportu za pomocą słowa kluczowego `options`. Ustawienia te mają zastosowanie wyłącznie do rozpatrywanego bufora. Wszystkie dostępne opcje mają również odpowiadającą im zmienną, którą można dostosować tak, aby miała zastosowanie do wszystkich eksportów. Informacje na temat wszystkich dostępnych opcji można znaleźć w podręczniku za pomocą polecenia `C-h R org` i wyszukując odpowiedni rozdział za pomocą polecenia `g export settings`.

Tytuł, autor i data to podstawowe metadane. Każdy format eksportu wykorzystuje również określone zmienne w przedniej części dokumentu. Szablon do eksportu do formatu docelowego można uzyskać za pomocą `org-export-insert-default-template`, który wstawia słowa kluczowe związane z wybranym formatem eksportu do bieżącego bufora `Org`. Dostęp do tej funkcji można uzyskać za pomocą dyspozytora eksportu za pomocą `C-c C-e #`. Użycie więcej niż jednego formatu eksportu dla jednego manuskryptu jest całkowicie dopuszczalne. Jednak użycie tej funkcji więcej niż raz prowadzi do powielenia niektórych słów kluczowych.

`Org` zapisuje wynik procesu eksportu w tym samym katalogu, co dokument źródłowy. Eksportowany plik ma taką samą nazwę jak plik źródłowy, ale z innym rozszerzeniem. Eksportowanie plików `Denote Org` może spowodować konflikt nazw plików, ponieważ powoduje to



dwa pliki o tym samym identyfikatorze. Możesz zmienić nazwę eksportowanego pliku, dodając `#+export_file_name: <nazwa pliku>` do nagłówka. Nazwa pliku może być dowolnym ciągiem znaków bez rozszerzenia pliku. Możesz eksportować pliki do innego folderu, ale może to powodować błędy.

### *Typogwapy*

Opublikowany manuskrypt zawiera nie tylko litery i cyfry. Zawiera również specjalne znaki typograficzne, takie jak wielokropki, znaki obce i różne rodzaje myślników. Moduł eksportu Org analizuje znaki LaTeX, takie jak `\alpha`, do odpowiednich wartości czcionek. Więcej informacji na temat symboli specjalnych znajduje się w sekcji 4.5.

Dla typografa zwykły myślnik nie jest wcale taki prosty. Podczas gdy w większości tekstów elektronicznych używa się tylko zwykłego myślnika jako łącznika lub do oddzielania słów i liczb, najlepsze praktyki typograficzne są bardziej subtelne. Istnieją cztery rodzaje myślników, z których każdy ma swoje unikalne cechy i zastosowania. Myślnik jest najkrótszy, myślnik En jest nieco dłuższy, a myślnik M jest najdłuższy. Szerokość myślników En i Em jest mniej więcej taka sama jak szerokość wielkich liter N lub M. Myślniki te zapisuje się w zwykłym tekście jako pojedyncze, podwójne lub potrójne myślniki. Metoda ta jest potrzebna tylko podczas pisania tekstu. W wyrażeniach matematycznych silnik wyświetlania automatycznie używa właściwej wersji.

- -: Myślnik łączy elementy w słowach złożonych (krótkich) lub dzielonych na końcu wyrównanego do obu stron wiersza tekstu.
- --: Myślnik dzieli liczby lub słowa, zastępując słowo „do” (1–3 lub Marathon–Sparta).
- ---: Myślnik długi oddziela myśli i działa jak przecinek, dwukropek lub nawias: (—).
- \$-\$: W trybie matematycznym wpisz łącznik, aby uzyskać znak minus ( $a - b$ ).

*Dodaj tabelę eoeteets*

Org wstawia spis treści bezpośrednio nad pierwszym nagłówkiem pliku. Możesz zmienić to zachowanie, dostosowując zmienną `org-export-with-toc`. Spis treści można też modyfikować dla każdego manuskryptu, dodając słowo kluczowe `#+options:` na początku dokumentu, na przykład:

- `#+options:`      `toc:t`: Dodaj pełną treść.
- `#+options:`      `toc:n`: Uwzględnij tylko *n* poziomów w spisie treści.
- `#+options:`      `toc:nil`: Nie eksportuj spisu treści.

*Ustawienia numeracji sekcji*

Numerowane sekcje są standardem w tekstach biznesowych i akademickich. Po dodaniu `num` do słowa kluczowego `startup`, tryb Org numeruje każdy nagłówek (`#+startup:`      `num`). Numeracja pojawia się w buforze Org jako tekst wirtualny, więc zawartość pliku pozostaje niezmieniona. To, czy numeracja ta pojawi się również w opublikowanym wyniku, zależy od szablonu eksportu dla danego formatu.

Niektóre części manuskryptu, takie jak przednia i tylna część książki lub lista odniesień, zazwyczaj nie są numerowane. Aby wykluczyć nagłówki z numeracji podczas eksportu, wpisz `C-c C-x p` (`org-set-property`) i ustaw właściwość `UNNUMBERED` jako `t` (prawda). Po ustawieniu tej właściwości na notoc nagłówki zostaną również wykluczone ze spisu treści. Właściwości te mają zastosowanie tylko do eksportowanego dokumentu dla tego nagłówka i jego elementów podrzędnych, ale nie do samego bufora Org.

```
:WŁAŚCIWOŚCI:
:UNNUMBERED: t
:KONIEC:
```

### Tabele zetaeieg

Org ułatwia wstawianie tabel, jak omówiono w sekcji 4.5. W dokumencie źródłowym tabele zawierają pionowe linie dla każdej kolumny, a użytkownicy mogą dodawać poziome linie w razie potrzeby. Nie musi to jednak oznaczać, że tabela będzie wyglądać tak samo w ostatecznym wydruku.

Linie poziome zdefiniowane w dokumencie źródłowym pojawiają się w ostatecznym wyniku, ale linie pionowe są pomijane (University of Chicago Press, 2017, sekcja 3.53). Większość publikacji dopuszcza tylko linie poziome, ponieważ dane w tabeli są wyrównane do kolumn, więc oko nie potrzebuje linii pionowej do oddzielenia grup.

Czasami pionowe linie mogą być pomocne w podziale tabeli na grupy kolumn. Wiersz z ukośnikiem (/) w pierwszym polu określa, które kolumny tworzą grupę otoczoną pionowymi liniami. Pozostałe pola zawierają symbol mniejszy niż (<), wskazujący, że dana kolumna rozpoczyna grupę, lub symbol większy niż (>), wskazujący koniec kolumny. Użycie symboli <> w kolumnie tworzy oddzielną grupę otoczoną pionowymi liniami. Symbole te wskazują miejsce, w którym znajduje się pionowa linia. Możliwe jest również wstawienie symboli rozpoczynających grupę kolumn (<) po każdej żądanej pionowej linii. Jednak ta metoda nie dodaje linii po lewej i prawej stronie tabeli. Poniższy przykład ilustruje te zasady, a wersja eksportowana znajduje się w tabeli 8.1 poniżej.

#+caption: Przykład tabeli z pionowymi liniami i wyrównaniem. #+name:

tab-lines

| n | n^2 | n^3 | n^4 | Zwierzę  | a  | b  |
|---|-----|-----|-----|----------|----|----|
| / | <   |     | >   |          | <> |    |
|   | <r> | <c> |     | <r>      |    |    |
| 1 | 1   | 1   | 1   | Aardvark | 3  | 4  |
| 2 | 4   | 8   | 16  | Ptak     | 5  | 12 |
| 3 | 9   | 27  | 81  | Krowa    | 8  | 15 |

Tryb Org może obsługiwać tylko proste tabele bez informacji rozciągających się na wiele kolumn lub wierszy. Aby utworzyć bardziej złożone tabele, Org integruje się z wbudowanym

Tabela 8.1: Przykład tabeli z pionowymi liniami i wyrównaniem.

| n | n <sup>2</sup> | n <sup>3</sup> | n <sup>4</sup> | Zwierzę  | a | b  |
|---|----------------|----------------|----------------|----------|---|----|
| 1 | 1              | 1              | 1              | Aardvark | 3 | 4  |
| 2 | 4              | 8              | 15             | Ptak     | 5 | 12 |
| 3 | y              | 27             | 81             | Krowa    | 8 | 15 |

Pakiet tabel autorstwa Takaaki Ota. Tabele te mają nieco inną składnię niż Org, co ilustruje poniższy przegląd niemieckich artykułów.

Aby przekonwertować standardową tabelę Org mode do bardziej złożonego formatu, użyj C-c ~. Aby edytować zaawansowaną tabelę w Org, użyj org-edit-special (C-c `), co spowoduje otwarcie nowego bufora zawierającego tylko tabelę. Ten sam skrót klawiaturowy pozwala powrócić do dokumentu. Aby dowiedzieć się więcej o składni tego pakietu, przeczytaj instrukcję za pomocą C-h P table.

|                                       |        |                           |                 |        |                 |
|---------------------------------------|--------|---------------------------|-----------------|--------|-----------------|
| +-----+-----+-----+-----+-----+-----+ |        |                           |                 |        |                 |
|                                       |        | Liczba pojedyncza         |                 |        | Liczba mnoga    |
|                                       |        | +-----+-----+-----+-----+ |                 |        |                 |
|                                       |        | Męski                     | Niejednoznaczny | Żeński | Wszystkie płcie |
| +-----+-----+-----+-----+-----+-----+ |        |                           |                 |        |                 |
| Mianownik                             | der    |                           | das             | die    | die             |
| Będący                                | den    |                           | das             | die    | die             |
| Celownik                              | dem    |                           | dem             | der    | denen           |
| Dopełniacz                            | dessen |                           | dessen          | deren  | deren           |
| +-----+-----+-----+-----+-----+-----+ |        |                           |                 |        |                 |

Tabela 8.2: Tabela wyeksportowana za pomocą pakietu Table.

|            | Liczba pojedyncza |               |        | Liczba pojedyncza |
|------------|-------------------|---------------|--------|-------------------|
|            | Męski             | Rodzaj nijaki | Żeński | Wszystkie płcie   |
| Mianownik  | mianownik         | das           | die    | die               |
| Bierny     | den               | das           | die    | die               |
| Dopełniacz | dem               | dem           | der    | denen             |
| Dopełniacz | dessen            | dessen        | deren  | deren             |

### *Quotations and other structures*

Najlepszą radą dla każdego pisarza jest używanie oryginalnych słów podczas wyrażania swoich myśli. Jednak czasami słowa innych autorów są tak mocne, że nie tylko należy je zacytować, ale także inspirować i motywować nas. Najbardziej bezpośrednim sposobem jest oczywiście użycie „cudzysłów”. Większość formatów eksportu definiuje oddzielną typografię dla cytatów, np. wcięcie akapitu lub użycie czcionki pochylej. Poleć Orgowi użycie szablonu struktury cytatu za pomocą C-c C-, q.

```
#+begin_quote
```

```
 Dobre pisanie to w zasadzie przepisywanie (Roald Dahl)
```

```
#+end_quote
```

Szablony struktury organizacyjnej to predefiniowane fragmenty tekstu, które umożliwiają szybkie wstawianie często używanych struktur. Usprawniają one przepływ pracy, ograniczając powtarzalne wpisywanie tekstu. W poprzednim rozdziale zapoznałeś się już z szablonem struktury dla notatek. Szablony struktury najbardziej związane z tematyką tej książki to notatki, cytaty, wersety i tekst wyśrodkowany. Szablony struktury instruują proces eksportu, aby odpowiednio traktował te części tekstu.

### *Macros*

Org posiada również funkcję umożliwiającą stosowanie makr w manuskrypcie. Makro składa się z ciągu znaków umieszczonych pomiędzy potrójnymi nawiasami klamrowymi. Org rozszerza makro do pełnego tekstu podczas procesu eksportu. Na przykład w tej książce makro `{{{ews}}}` zostało zastąpione przez: *Zmaes Wvities Studio*. Podstawowe makro definiuje się w przedniej części dokumentu:

```
#+macro: ews /Emacs Writing Studio/
```

Org zawiera również szereg wbudowanych makr. Jednym z przykładów jest dynamiczne umieszczanie znaczników czasu w polu daty. Znaczniki czasu w Org są statyczne, ale można użyć makra eksportowego

, aby dynamicznie wstawić datę. Użycie `{{{time(format)}}}` powoduje wstawienie bieżącej daty w formacie opisanym powyżej.

Makra Org to wszechstronny zestaw narzędzi, który może pomóc w rozszerzaniu często powtarzających się fragmentów tekstu. Makra mogą również zawierać zmienne i funkcje Emacs Lisp, aby tworzyć dynamiczne rozszerzenia. Podręcznik Org zawiera bardziej zaawansowane przykłady rozszerzeń makr (C-h r org <ret> g macro). Sekcje dotyczące eksportu do HTML i LaTeX zawierają dodatkowe informacje na temat makr.

### *Zxeludieg eoeteet fvom expovt*

Nie wszystko, co piszesz, powinno być udostępniane światu. Org wyklucza wszelkie komentarze z ostatecznego formatu, jak omówiono w sekcji 7.1. Konfiguracja EWS nie eksportuje również szuflad, więc Twoje notatki pozostają prywatne.

Ponadto można wykluczyć sekcję i powiązane podsekcje (poddzweto) tekstu z eksportu, dodając tag `:noexport:` do nagłówka za pomocą C-c C-q (org-set-tags-command). Nazwę tagu wpisuje się w minibufferze i można użyć funkcji uzupełniania, aby znaleźć tagi już użyte w bieżącym buforze. Tag pojawia się po prawej stronie nazwy nagłówka, pomiędzy dwoma dwukropkami.

Tryb Org wyeksportuje nazwę nagłówka wraz z powiązanymi tagami. Możesz unieważnić to zachowanie, dodając tagi:nil do słowa kluczowego opcji.

### *Zxpovtiieg eitatioes*

Org posiada wbudowany menedżer cytatów, który może wykorzystywać pliki bibliograficzne BibTeX / BibLaTeX lub CSL do odwoływania się do pozycji bibliograficznych, takich jak artykuły i książki. Podstawy są proste i działają bez konieczności konfiguracji lub zewnętrznego oprogramowania.

Wiesz już, jak utworzyć globalną i lokalną bibliografię oraz wstawiać cytaty do buforów Org (sekcje 5.2 i 7.1). W tej sekcji wyjaśniono, w jaki sposób Org konwertuje konstrukcje takie jak `[@cite:wittgenstein_1922]` na sformatowany cytat, na przykład „(Wittgenstein, 1y22)”.

Aby wyeksportować cytaty, potrzebujesz tylko globalnej lub lokalnej bibliografii, jednego lub więcej cytatów oraz listy literatury, do której się odwołujesz (bibliografii), tak jak w tym przykładzie:

```
#+bibliography: references.bib
„Świat jest wszystkim, co ma miejsce” [cyt. za: Wittgenstein, 1922].
* Odniesienia
#+print_bibliography:
```

Potrzebny jest procesor cytatów, aby przekonwertować cytaty w pliku Org do wersji eksportowanej. Org domyślnie używa podstawowego procesora eksportu, który jest odpowiedni dla prostych projektów i ma zastosowanie do wszystkich formatów eksportu. Inne procesory zapewniają bardziej elastyczne zarządzanie cytatami, jak omówiono w sekcjach 8.3 i 8.5.

Podstawowy procesor cytatów można skonfigurować za pomocą słowa kluczowego w metadanych dokumentu. Słowo kluczowe określa procesor (podstawowy), a po nim styl bibliografii i styl cytowania:

```
#+cite_export: basic [styl bibliografii] [styl cytowania]
```

Org konwertuje cytaty na sformatowany tekst podczas procesu eksportu i dodaje listę odniesień. Aby dołączyć listę odniesień, dodaj `#+print_bibliography:` w miejscu, w którym ma się ona pojawić. Lista odniesień nie zawiera nagłówka w przypadku korzystania z podstawowego silnika. Druga część słowa kluczowego `#+cite_export:` określa styl bibliografii, który może być jedną z trzech opcji:

- `author-year:` tryb domyślny.
- `numeric:` system Vancouver z numerowanymi wpisami.
- `plain:` Tak samo jak domyślny, ale tylko nazwiska autorów.

Podstawowy procesor eksportu obsługuje różne style cytowania. Jeśli nie podano stylu cytowania, domyślnie stosuje się autora (autorów) i rok (lata) w nawiasach, np. „(Toulmin, Stephen, 2003)”. Istnieją dwa sposoby definiowania stylu cytowania. Trzecia część

#+cite\_export: zapewnia domyślny styl, który można zmienić w poszczególnych cytatach.

Dostępne są następujące style cytowania:

- autor (/a): Tylko autorzy „Toulmin, Stephen”
- noauthor (/na): Brak autorów „(2003)”
- tekst (/t): cytata w tekście „Toulmin, Stephen (2003)”
- nocite (/n): Brak cytatu, ale dodaj wpis do listy referencji.
- note (/ft): Cytat jako przypis.
- numeryczny (/nb): Numerowane odniesienia w nawiasach (system Vancouver).

Możesz zmienić ustawienie domyślne, dodając znacznik stylu przed cytatami (wymienionymi w nawiasach powyżej). Na przykład [cite/t:@rorty\_1979] jest eksportowane jako „Rorty (1y7y)” w systemach cytowania według autora i roku.

Inną opcją konfiguracyjną jest dodanie wariantu cytowania. Ten wskaźnik określa, czy należy usunąć nawiasy (/b), czy też zamienić pierwszą literę na wielką (/c). Wariant dodaje się po stylu. Na przykład [cite/ft/b:@rorty\_1979] powoduje utworzenie przypisu bez nawiasów: „Rorty, 1y7y”.

Niektóre kombinacje stylów bibliografii i cytowania nie mają sensu. Na przykład numeryczna bibliografia i styl cytowania przypisów są niekompatybilne. Org ignoruje niemożliwe kombinacje stylu i wariantu.

Org może również dodawać przedrostki i przyrostki do cytatów. Na przykład [cite: See @rorty\_1979 p.12] daje wynik „(See Rorty 1y7y p.12)”. Istnieje również globalny przedrostek i przyrostek w przypadku używania wielu kluczy cytowania. Cytat z wszystkimi dodatkami może więc wyglądać następująco:

```
[@cite/s/v: global-prefix; prefix @key1 suffix; prefix @key2 suffix; global-suffix]
```



## 8. ę Tworzenie dokumentów biurowych

Istnieje wiele sytuacji, w których musimy udostępniać nasze prace osobom preferującym korzystanie z edytorów tekstu. W ramach mojej osobistej pracy często piszę notatki służbowe i raporty techniczne w programie Emacs, a następnie eksportuję je do formatu edytora tekstu, aby udostępnić je współpracownikom i ułatwić współpracę.

Funkcja eksportu w Org umożliwia tworzenie dokumentów w formatach obsługiwanych przez edytory tekstu, takie jak LibreOffice Writer, Apple Pages lub MS Word. Eksportowanie do formatu edytora tekstu jest przydatne podczas pisania dla klientów korporacyjnych lub współpracy z współautorami lub redaktorem. Proces eksportu Org powoduje utworzenie pliku w formacie OpenDocument Text (ODT). Plik ODT jest skompresowanym zbiorem plików XML i osadzonych obrazów, co wymaga dostępności programu Zip. Rozszerzalny język znaczników (XML) definiuje i przechowuje dane, w tym dokumenty tekstowe, w sposób umożliwiający ich udostępnianie. Po otwarciu pliku

Plik ODT za pomocą programu do archiwizacji, takiego jak Unzip, znajdziesz następujące pliki:

- meta.xml zawiera metadane dla tego pliku.
- styles.xml to arkusz stylów definiujący układ i typografię.
- content.xml zawiera rzeczywisty dokument.

Kiedy Org eksportuje bufor do formatu ODT, generuje dwa pliki: meta.xml i content.xml oraz arkusz stylów. Arkusz stylów jest kopiowany z domyślnej konfiguracji lub z pliku dostarczonego przez użytkownika. Narzędzie eksportu ODT posiada dodatkowe słowa kluczowe front-matter, które pozwalają dostosować wynik. Podtytuł jest dodawany do treści, natomiast opis i słowa kluczowe są dostępne w metadanych pliku.

- `#+subtitle`: Podtytuł dokumentu.
- `#+description`: opis pliku.
- `#+keywords`: Słowa kluczowe eksportowanego pliku (plików).

Jeśli masz zainstalowany pakiet LibreOffice, możesz również utworzyć plik DOCx, aby ułatwić użytkownikom programu MS Word dzielenie się radością z czytania Twoich tekstów. Po włączeniu tej opcji proces eksportu spowoduje utworzenie zarówno pliku ODT, jak i DOCX. Możesz zmienić format wyjściowy, dostosowując zmienną `org-odt-preferred-output-format`. Możesz również dostosować tę zmienną, aby polecić pakietowi LibreOffice wygenerowanie pliku PDF.

### *Obrazy i tabele*

Aby kontrolować sposób eksportowania tabel i obrazów w pliku ODT, użyj linii `#+attr_odt:` tuż nad elementem. Dostępne są różne właściwości dotyczące rozmiaru i umiejscowienia obrazów:

- `:width` i `:height` kontrolują rozmiar obrazu w centymetrach. Można użyć tylko szerokości lub wysokości albo obu tych parametrów. Parametr ten akceptuje tylko wartości liczbowe, nie akceptuje wartości procentowych.
- `:scale` określa względną szerokość obrazu źródłowego.
- `:anchor` kotwiczy obraz jako znak, do akapitu lub do strony.

Na przykład obraz o następujących właściwościach ma szerokość dziesięciu centymetrów i jest zakotwiczony jako znak:

```
#+attr_odt: :width 10 :anchor as-char
```

Podczas kotwiczenia obrazu jako znaku (`as-char`) jest on umieszczany w dokumencie tak jak każdy inny znak. Obraz przesuwa się wraz z tekstem podczas dodawania lub usuwania tekstu przed obrazem. Obraz zakotwiczony do akapitu przesuwa się wraz z akapitem. Podczas kotwiczenia obrazu do strony zachowuje on tę samą pozycję względem marginesów strony i nie przesuwa się. Metoda ta jest szczególnie praktyczna w przypadku publikowania dokumentów o złożonym układzie, takich jak biuletyny.

Na podstawie wyżej wymienionych właściwości silnik eksportu ODT określa rozmiar obrazu w centymetrach. Obraz źródłowy jest osadzany w dokumencie ODT w

rozdzielczość 5 punktów na cal (DPI). Aby użyć innej rozdzielczości, należy dostosować zmienną `org-odt-pixels-per-inch`. Można się zastanawiać, dlaczego rozmiar obrazu jest podawany w centymetrach, a rozdzielczość w jednostkach imperialnych. Niestety tak właśnie działa ten system.

Org może eksportować tabele do formatu ODT. Domyślnie tabele mają górne i dolne ramki oraz poziome i pionowe linie, zgodnie z definicją w źródle. Właściwość `:rel-width` kontroluje szerokość tabeli w procentach szerokości tekstu. Wszelkie rozmiary kolumn określone w tabeli będą odnosić się do całkowitej szerokości (patrz sekcja 4.5). Poniższa linia właściwości nad tabelą spowoduje jej eksportowanie przy względnej szerokości 75% szerokości tekstu:

```
#+attr_odt: :rel-width 75
```

### *Matematyki*

Eksport Org do formatu ODT ignoruje formuły LaTeX, ale istnieje sposób na obejście tego problemu.

Najłatwiejszą metodą jest konwersja formuł matematycznych do pliku graficznego poprzez dodanie następującego słowa kluczowego na początku dokumentu: `#+options: tex:dvipng` dla eksportu do formatu PNG. Ta metoda wymaga programu `dvipng`. Alternatywnie można użyć Image Magic z `tex:imagemagick`. Dokumenty ODT mają natywny format wzorów (MathML), który Org może eksportować.

MathML wyraża formuły matematyczne w notacji opartej na XML. Jednak ta opcja wymaga zaawansowanej konfiguracji i konwertera MathML. Podręcznik Org zawiera bardziej szczegółowe wskazówki dotyczące korzystania z MathML.

### *Szablony stylów*

Kontrola typografii i układu dokumentów biurowych wymaga pliku OpenDocument Text Template (OTT). Pliki te zawierają ustawienia służące do generowania nowych dokumentów, w tym układ typograficzny i inne elementy. Odpowiedni szablon określa się w przedniej części bieżącego bufora za pomocą słowa kluczowego `#+odt_styles_file:`, po którym następuje ścieżka do pliku stylu (w formacie OTT lub ODT). Aby używać tego samego szablonu dla wszystkich eksportów ODT, należy dostosować zmienną `org-odt-styles-file`.

Tworzenie szablonu pliku stylu jest bardzo proste. Utwórz pusty dokument Org, dodaj `#+options: H:4 num:t author:nil` i wyeksportuj do formatu ODT za pomocą `C-c C-e o o`. Słowo kluczowe `options` tworzy cztery poziomy nagłówków z numeracją. Oczywiście możesz modyfikować te ustawienia zgodnie z własnymi preferencjami. Otwórz wyeksportowany dokument w programie LibreOffice i edytuj style (F11).

Tryb Org używa pewnych konkretnych stylów, które zaczynają się od „Org”, więc upewnij się, że je skonfigurujesz. Gdy dokument będzie już stylizowany zgodnie z Twoimi preferencjami, zapisz go jako plik OTT i dołącz do swojego manuskryptu. Przy następnym eksporcie dokumentu Org wynik będzie miał styl zdefiniowany w szablonie. Org wyodrębnia plik `styles.xml` osadzony w pliku szablonu i kopiuje go do eksportowanego pliku.

Jeśli style zawierają obrazy, takie jak obraz tła strony, należy również określić to w pliku stylów za pomocą słowa kluczowego, jak pokazano w przykładzie.

```
#+odt_styles_file: („template.ott” („styles.xml” „background.png”))
```

Należy określić plik `styles.xml`, a plik `background.png` jest plikiem wyznaczonym jako obraz tła w dokumencie szablonu. Należy pamiętać, że LibreOffice zmienia nazwy plików, więc należy otworzyć szablon w programie Emacs i nacisnąć `C-c C-c`, aby wyświetlić strukturę plików i skopiować nazwę pliku obrazu. Ta konfiguracja jest idealna do pisania dokumentów firmowych.

Proces eksportu ODT opiera się na określonych szablonach i nazwach stylów. Szablony innych producentów mogą powodować niezgodności, jeśli używają innych nazw stylów. Należy pamiętać, że szablony można tworzyć wyłącznie w programie LibreOffice. Niestety komercyjne edytory tekstu są niekompatybilne z formatem ODT używanym w oprogramowaniu open source.

Można precyzyjnie dostosować sposób eksportowania Org do OpenOffice. Szczegółowe omówienie eksportu Open Document znajduje się w sekcji *Open Document Text Export* podręcznika Org: `C-h R org <ret> g open`.

### *Styl cytowania Język*

Podstawowy procesor cytowań działa doskonale podczas eksportowania do formatu ODT, ale ma ograniczone możliwości dostosowywania cytatów lub spełniania oczekiwań uczelni lub wydawcy. Korzystanie z języka stylu cytowania (CSL) zapewnia szerokie możliwości stylizacji cytatów. CSL został stworzony przez Bruce'a D'Arcusa, który opracował również pakiet Citar opisany w sekcji 5.2. Procesor cytowań CSL można stosować do wszystkich formatów eksportu. W niniejszej książce do formatów publikacji wykorzystano procesor cytowań CSL.

Silnik CSL opiera się na pliku definiującym wynik. Pliki te można znaleźć na stronie internetowej [citationstyles.org](http://citationstyles.org). W CSL dostępnych jest wiele tysięcy odmian dla konkretnych czasopism lub uniwersytetów. Pobierz pliki stylów odpowiednie dla swoich projektów pisarskich i zapisz je na przyszłość.

Domyślnie procesor CSL renderuje cytaty w formacie Chicago autor-data. Można użyć innego pliku stylu, określając go w dokumencie poprzez dodanie nazwy pliku do słowa kluczowego `#+cite_export:`, na przykład:

```
#+cite_export: csl /ścieżka/do/pliku-stylu.csl
```

Jeśli przechowujesz zbiór plików CSL w tym samym folderze, musisz dostosować zmienną `org-cite-csl-styles-dir`, aby Org mógł je znaleźć. Gdy ta zmienna jest zdefiniowana, wystarczy tylko podać nazwę pliku w przedniej części dokumentu.

Procesor CSL obsługuje następujące style cytowania, z których niektóre są identyczne z tymi obsługiwanymi przez podstawowy procesor omówiony powyżej.

- `autor (/a)`: Tylko autor.
- `noauthor (/na)`: Brak autora (autorów).
- `tekst (/t)`: cytat w tekście.
- `nocite (/n)`: Cytat nie jest podany, ale znajduje się w bibliografii.
- `year (/y)`: Tylko rok

- `title (/ti)`: Tytuł wpisu.
- `bibentry (/b)`: Pełny cytat zgodnie z bibliografią.

CSL zapewnia funkcję dodawania przefiltrowanej listy odniesień według słowa kluczowego lub typu publikacji. Można to połączyć z opcją `nocite`, aby utworzyć bibliografię tematyczną. Użycie znaku `*` jako klucza w cytacie `nocite` powoduje uwzględnienie wszystkich dostępnych pozycji. Poniższy przykład eksportuje wszystkie książki z bibliografii globalnej i lokalnej zawierające słowo kluczowe „Emacs”. Należy pamiętać, że w słowie kluczowym rozróżniana jest wielkość liter.

```
#+title: Książki o Emacsie
#+bibliography: ../library/emacs-writing-studio.bib #+cite_export:
csl
```

Pokaż listę książek oznaczonych słowem kluczowym „Emacs”.

```
* Bibliografia
[cite/n:@*]
#+print_bibliography: :type book :keyword Emacs
```

## 8.4 Generuj strony internetowe

Emacs zawiera wbudowany tryb główny do edycji plików HTML. Jednak Org ma doskonałe możliwości eksportu do tego formatu, więc można korzystać z jego lekkiego znacznika bez martwienia się o składnię techniczną. Org eksportuje się do HTML w celu publikowania stron internetowych i tworzenia ebooków w formacie ePub, co opisano w sekcji 8.5.

Język HyperText Markup Language (HTML) jest silnikiem napędzającym sieć WWW. Org eksportuje bezpośrednio do HTML; wyeksportujmy ten mały dokument Org.

```
#+title: Przykład HTML

Hello world!
```

Przykład konwertuje się na prosty plik z znacznikami HTML otoczonymi nawiasami kątowymi (symbole mniejsze i większe niż). Wcięcia są opcjonalne, ale pomagają zrozumieć strukturę dokumentu.

```
<!DOCTYPE html>
<html>
 <head>
 <title>Przykład HTML</title>
 </head>
 <body>
 <div>
 <p>Witaj świecie</p>
 </div>
 </body>
</html>
```

Silnik eksportu HTML w Org pozwala ustawić szeroki zakres właściwości dokumentu oprócz tych, które zostały już omówione. Konfiguracje te pozwalają precyzyjnie dostosować wyjściowy kod HTML, co wymaga dogłębnej wiedzy na temat kodowania stron internetowych i dlatego jest przeznaczone dla zaawansowanych użytkowników i wykracza poza zakres tej książki. Szczegółowe informacje na ten temat można znaleźć w podręczniku Org mode (C-c R org <ret> g html). Wszystkie dostępne ustawienia HTML można wyświetlić, naciskając C-c C-e # i wybierając html.

Domyślny eksport dodaje do strony postambulę zawierającą metadane. Można to zmienić, dodając html-postamble:nil do słowa kluczowego options w sekcji front matter.

### *Obrazy i tabele*

Org udostępnia szereg atrybutów służących do definiowania sposobu eksportowania obrazów i tabel. Konfiguracje te są bezpośrednio przekształcane na atrybuty HTML. Słowo kluczowe #+attr\_html: definiuje atrybuty zarówno dla obrazów, jak i tabel.

- `:alt` dostarcza alternatywnych informacji o obrazie, jeśli użytkownik nie może go wyświetlić (na przykład z powodu wolnego połączenia, błędu technicznego lub korzystania z czytnika ekranu).
- `:title` dodaje tytuł obrazu. Informacja ta pojawia się jako tekst podpowiedzi po najechaniu myszką na obraz.
- `:align` lewa, środkowa lub prawa.
- `:width` i `:height` do ustawienia rozmiaru obrazu (w pikselach lub procentach)

Obrazy w Org mogą również stanowić linki, gdy opis linku jest sam w sobie obrazem, plikiem lokalnym lub linkiem internetowym. Na przykład, aby wstawić miniaturę z hiperłączem do jej wersji w wysokiej rozdzielczości, użyj czegoś takiego:

```
[[plik:wysoka rozdzielczość.jpg][plik:miniatura.jpg]]
```

Tabele są również modyfikowane przez słowo kluczowe `#+attr_html:`, które udostępnia trzy atrybuty:

- `:border` określa szerokość obramowania wokół tabeli.
- `:rules` ustawione na `all` rysuje wszystkie linie tabeli.
- `:frame` określa widoczność zewnętrznych obramowań. Domyślnie zewnętrzne obramowania nie są wyświetlane. Użyj `border`, aby wyświetlić wszystkie linie obramowania.
- `:cellspacing` i `:cellpadding` dostosowują wypełnienie wewnątrz komórek i odstęp między nimi.



### *Matematyki*

Eksportowanie pliku Org do HTML zawiera odniesienie do MathJax. Ta biblioteka JavaScript wyświetla notację matematyczną w przeglądarkach internetowych. Można zmienić to zachowanie, dodając bezpośrednio obrazy do wyjścia HTML za pomocą `#+options: tex:dvisvgm`, co konwertuje formuły do plików SVG, dzięki czemu nie trzeba polegać na JavaScript.

### *Arkusze stylów*

Plik HTML zawiera treść i strukturę, natomiast kaskadowe arkusze stylów (CSS) określają układ i typografię. Domyślny silnik eksportu zawiera podstawowy arkusz stylów w przedniej części dokumentu, który można zastąpić własnym. Słowo kluczowe `#+html_head`: pozwala dodawać wiersze do preambuły dokumentu.

```
#+html_head: <link rel="stylesheet" type="text/css" href="style1.css" />
#+html_head_extra: <link rel="stylesheet" type="text/css" href="style2.css" />
```

### *Cytaty*

Moduł eksportu HTML może przetwarzać cytaty przy użyciu zarówno podstawowych, jak i procesorów CSL, zgodnie z opisem w sekcjach 8.2 i 8.3.

### *Jeżeli dostosowany HTML*

Jeśli znasz się na pisaniu w HTML, wrzuć go bezpośrednio do pliku Org. Dzięki temu możesz dodawać formularze online i osadzać multimedia lub niestandardową typografię. Wstaw szablon struktury HTML za pomocą `C-c C-, h` i dodaj swój własny kod. Twoja strona internetowa będzie zawierała wszystkie linie między znacznikami początku i końca w niezmienionej formie.

```
#+begin_export html
<jakiś kod html/>
#+end_export
```

Te bloki HTML są eksportowane tylko wtedy, gdy format wyjściowy jest zgodny. Jeśli przygotujesz dokument w wielu formatach, musisz dołączyć alternatywną wersję LaTeX lub ODT.

Niektóre fragmenty kodu HTML mogą pojawiać się wielokrotnie w dokumentach. Większość narzędzi do tworzenia stron internetowych udostępnia skróty, czyli fragmenty tekstu służące do wstawiania kodu HTML. Makra Org idealnie nadają się jako skróty HTML. Można ich używać do osadzania złożonego kodu HTML, takiego jak formularze i multimedia, lub prostych aplikacji, takich jak zmiana koloru tła wybranych słów.

Poniższy przykład tworzy makro, które rozszerza się do osadzonego klipu YouTube przy użyciu jego kodu identyfikacyjnego. Część \$1 makra reprezentuje pierwszy parametr w makrze. Każdy kolejny parametr będzie miał wartość \$2 i tak dalej.

```
#+macro: youtube <iframe src="https://youtube.com/embed/$1"></iframe>
```

W tym przykładzie osadzono film „*Me at the Zoo*”, pierwszy film w historii serwisu YouTube:

```
{{{youtube(jNQXAC9IVRw)}}}
```

Ten krótki kod eksportuje się do:

```
<iframe src="https://youtube.com/embed/jNQXAC9IVRw"></iframe>
```

### *Publikowanie stron internetowych*

Standardowy proces eksportu konwertuje tylko jeden plik i wszystkie pliki dołączone. Może to być uciążliwe podczas pracy nad stroną internetową, która składa się z dużej liczby pojedynczych plików.

Org zawiera również system publikowania służący do generowania stron internetowych, który wymaga eksportowania zbioru plików Org. Można go dostosować, aby wyznaczyć folder źródłowy, a także szereg innych ustawień, które determinują wygląd strony internetowej.

Korzystanie z tej funkcji wymaga wiedzy technicznej na temat tworzenia stron internetowych oraz znajomości języka Emacs Lisp, dlatego nie jest ona szczegółowo opisana w niniejszej książce.

można znaleźć sekcję w podręczniku Org dotyczącą publikowania dokumentów za pomocą C-h R org <ret> g publish.

## 8.5 Tworzenie e-booków

Większość wydawców e-booków używa formatu ePub do dystrybucji. Format ePub to skompresowany plik, którego zawartość jest przechowywana jako strona internetowa zoptymalizowana pod kątem czytników e-booków. Pakiet ox-ePub Marka Meyera dodaje tę funkcjonalność do dyspozytora eksportu Org. Pakiet ten wykorzystuje wbudowaną funkcję eksportu Org do HTML do tworzenia e-booków, umożliwiając wykorzystanie dowolnej z opisanych powyżej funkcji do dostosowania wyniku. Pakiet ten tworzy plik zgodny ze standardem ePub w wersji 2.0.1.

Istnieją pewne specyficzne opcje eksportu dla tego formatu, wykraczające poza te już omówione, które można dodać za pomocą C-c e # epub:

- #+uid: Unikalny identyfikator dokumentu, znany również jako URI, może być adresem strony internetowej lub numerem ISBN. Ta właściwość jest obowiązkowa.
- #+publisher: Nazwa wydawcy książki.
- #+license: Licencja praw autorskich lub copyleft.
- #+epubstyle: Plik CSS używany do eksportu.
- #+epubcover: Obraz okładki książki.

Domyślna konfiguracja eksportu HTML dodaje postambuł na dole ostatniej strony, zawierający sygnaturę czasową, autora i link do usługi walidacji HTML. Dodanie #+options: html-postamble:nil do nagłówka pliku Org usuwa te elementy z e-booka.

Pakiet ox-ePub nie konwertuje znaczników czasu Org, takich jak [2024-07-08 Mon], do formatu daty zgodnego ze standardem ePub. Można to poprawić, usuwając

1  
rok

nawiasów kwadratowych oraz nazwy dnia i godziny ze znacznika czasu. Alternatywnie można użyć makra czasu, aby ustawić format daty na format ISO 8501:

```
{{{time(%Y-%m-%e)}}}
```

Zaleca się używanie wyłącznie otwartych formatów, takich jak PNG, podczas umieszczania obrazów w treści. Niektóre czytniki e-booków nie wyświetlają plików JPG i innych formatów zastrzeżonych. E-book będzie wyświetlał się poprawnie na komputerze, ale może nie przejść kontroli jakości wydawcy. Proces eksportu do formatu ePub zostanie przerwany, jeśli brakuje jakichkolwiek plików graficznych. Chociaż eksport do formatów HTML i PDF przebiega bezbłędnie, plik ePub nie zostanie wyrenderowany, jeśli brakuje jakichkolwiek powiązanych obrazów.

Nie powinniśmy oceniać książki po okładce, ale jest ona niezbędna dla Twojej publikacji. Słowo kluczowe `#+epubcover:` zawiera ścieżkę do pliku okładki. Idealne wymiary okładki e-booka to 2550 x 1500 pikseli lub dowolny inny rozmiar o proporcjach 1:1,5.

Wynik działania tego pakietu jest w pełni czytelny dla czytników ePub. Jednak dwie kwestie powodują, że manuskrypt nie przechodzi walidacji ePub. Gdy manuskrypt odwołuje się do innych plików w celu włączenia ich do dokumentu, zgodnie z opisem w sekcji 7.3, spis treści nie jest generowany poprawnie. Po drugie, Org eksportuje przypisy w sposób niezgodny ze standardami W3C dla plików e-Pub. Obie kwestie można uniknąć lub skorygować ręcznie za pomocą zaawansowanego edytora ePub, takiego jak Calibre.

## 8.6 Eksport do LaTeX i PDF

Kiedy pionier informatyki Donald Knuth otrzymał korektę drugiego wydania swojej książki „*The Art of Computer Programming*”, nie był zadowolony z rezultatu. W latach 70. wydawcy przeszli z tradycyjnego druku stałego na wersje elektroniczne. Swoje rozczarowanie przekształcił w pozytywną energię i postanowił opracować elektroniczny system składu tekstu o nazwie TEX (*tekst*), który odtwarza estetykę tradycyjnych książek (Knuth, 1y84). Oryginalny język jest złożony, więc Leslie Lamport opracował bibliotekę makr TEX o nazwie LaTeX (Lamport, 1yy4).

LaTeX pozwala tworzyć piękne dokumenty gotowe do druku. To narzędzie jest szeroko stosowane w środowisku akademickim i technicznym. LaTeX optymalizuje dokumenty do druku, więc świetnie nadaje się również do pisania książek niebędących publikacjami technicznymi. Książka w wersji papierowej została stworzona przy użyciu Org i wyeksportowana do formatu PDF za pomocą LaTeX.

Nie musisz koniecznie znać języka LaTeX, ale znajomość tego języka pomoże Ci dopracować wygląd publikacji. Podstawowa składnia języka LaTeX jest łatwa do wyjaśnienia. Wszystkie instrukcje LaTeX zaczynają się od odwrotnego ukośnika i parametrów ujętych w nawiasy klamrowe (`\command{}`). Poniższy przykład przedstawia minimalny działający plik LaTeX.

```
\documentclass[11pt, b5paper]{book}
\usepackage{ebgaramond}
\begin{document}
\title{Lorem Ipsum}
\author{Peter Prevos}
\date{lipiec 2024}
\maketitle
\section{Nunc eleifend}
Nunc aliquet, augue nec adipiscing interdum.
\end{document}
```

Obszar pomiędzy `\documentclass{...}` a `\begin{document}` to *pream-ble*, który zawiera polecenia mające wpływ na cały dokument. Treść jest umieszczona pomiędzy poleceniami `\begin{document}` i `\end{document}` po pream-ble. Pierwsza linia definiuje klasę dokumentu. Domyślną klasą używaną przez Org jest artykuł. Klasa dokumentu określa układ i typografię końcowego wydruku. Twórcy LaTeXa i wydawnictwa stworzyli klasy dokumentów dla różnych rodzajów publikacji. Najpopularniejsze klasy dokumentów to:

- `article`: artykuły naukowe
- `report`: mała książka lub praca dyplomowa
- `book`: pisanie długich książek

Każda klasa dokumentu ma konfigurowalne opcje umieszczone w nawiasach kwadratowych, takie jak standardowa czcionka i rozmiar papieru. Powyższy przykład definiuje klasę książki z czcionką 11-punktową i rozmiarem papieru B5 (175 × 250 mm).

Kolejna część dokumentu LaTeX inicjuje pakiety. LaTeX można rozszerzać za pomocą pakietów, podobnie jak Emacs. Pakiety te poprawiają wygląd dokumentu. W powyższym przykładzie polecenie `\usepackage{ebgaramond}` nakazuje LaTeXowi użycie czcionki Garamond, zastosowanej w wersji papierowej tej książki.

Kolejne linie to nagłówek, który określa tytuł, autora i datę. Nagłówek zamyka polecenie `\maketitle`, nakazujące LaTeXowi składanie bloku tytułowego.

Rzeczywisty rękopis zaczyna się po poleceniu `\maketitle`. Przykład definiuje nagłówek sekcji za pomocą polecenia `\section{}`. Standardowe akapity nie wymagają składni LaTeX.

Pisanie dokumentów bezpośrednio w LaTeX może być mylące, ponieważ tekst jest przepelniony ukośnikami odwrotnymi, nawiasami klamrowymi i innymi elementami składniowymi, które rozpraszają uwagę. Pakiet AUCTeX Emacs ułatwia pisanie w LaTeX. Jednak pakiet ten nie jest częścią *Zmaes Wvities Studio*, ponieważ Org ma doskonałe możliwości eksportu do LaTeX, więc można skorzystać z lekkiego znacznika Org. Podczas eksportowania dokumentu można go wyeksportować do pliku LaTeX lub PDF.

Nie potrzebujesz zewnętrznego oprogramowania, aby wyeksportować plik Org do LaTeX. Aby umożliwić eksportowanie plików Org do formatu PDF, musisz mieć zainstalowany LaTeX w swoim systemie. Sposób instalacji LaTeX zależy od systemu operacyjnego. Twoja ulubiona wyszukiwarka wskaże Ci właściwy kierunek.

Klawisze C-c C-e I p tworzą i otwierają wersję PDF bieżącego bufora Org oraz dołączonych plików. Org konwertuje bufor do pliku tex, a następnie oprogramowanie LaTeX konwertuje go do formatu PDF (rysunek 8.1).

EWS dostosowuje proces `ews-latex-pdf-process` do korzystania z programu `pdflatex`, który jest częścią wszystkich nowoczesnych instalacji LaTeX. Konfiguracja jest zoptymalizowana pod kątem tworzenia plików PDF i bibliografii.

*Dostosuj preambułę*

Funkcja eksportu Org dodaje standardowy preambułę do treści dokumentu, która definiuje typografię i układ. Masz precyzyjną kontrolę nad tym, jak Org tworzy preambułę.

Domyślną klasą dokumentu dla eksportu Org jest artykuł. Ta klasa dokumentu ma szerokie marginesy, co może wydawać się nietypowe dla początkujących użytkowników LaTeX. Marginesy nie są zbyt szerokie, ale papier jest zbyt duży. Optymalna długość linii tekstu wynosi od 50 do 75 znaków. LaTeX implementuje ten standard, co skutkuje szerokimi marginesami. Nie jesteś jednak ograniczony do tych domyślnych ustawień LaTeX.

Możesz zmienić klasę dokumentu w przedniej części pliku Org, ustawiając poniższe słowa kluczowe. Ten przykład wywołuje klasę book z papierem A4. Te linie są eksportowane do LaTeX jako `\documentclass[a4paper]{book}`.

```
#+latex_class: book #+latex_class_options:
[a4paper]
```

Aby użyć klasy dokumentu w Org, należy najpierw zdefiniować ją w pliku init. Standardowe klasy dokumentów `article`, `report` i `book` są dostępne domyślnie.

Możesz zmodyfikować preambułę LaTeX w przedniej części pliku Org, dodając dodatkowe pakiety. Poniższy przykład wykorzystuje klasę `article` z dwiema kolumnami. Ta konfiguracja wywołuje również pakiet geometryczny i ustawia rozmiar papieru na A4 z marginesami 25 mm. Ostatnia linia dodaje kolejne elementy preambuły, które w tym przypadku ustawiają czcionkę Times New Roman.

```
#+latex_class: article
#+latex_class_options: [twocolumn]
#+latex_header: \usepackage[a4paper, margin=25mm]{geometry} #+latex_header_extra:
\usepackage{times}
```

W przypadku korzystania z dokumentów powiązanych (sekcja 7.3) wystarczy zdefiniować odpowiednie pakiety w dokumencie źródłowym. Jednak definiowanie konfiguracji LaTeX w dołączonych

plikach może spowodować nadpisanie ustawień w dokumencie macierzystym, dlatego należy użyć opcji pomijania w dołączaniu.

Wykorzystanie przedniej części dokumentu do zdefiniowania preambuły LaTeX wymagałoby powtarzania tych samych wierszy dla każdego eksportowanego dokumentu. Nie byłoby to Emacs, gdyby nie można było skonfigurować tych ustawień w celu utworzenia biblioteki preambuł dokumentów z gotową preambułą. Zmienna `org-latex-classes` zawiera preambułę dla najczęściej używanych typów dokumentów. Klasa LaTeX w Org to kompletny preambuła dodawana do eksportowanych plików. Możesz zdefiniować bibliotekę klas dla tworzonych dokumentów, takich jak rozprawa, artykuł `apa`, książka `kluwer` lub cokolwiek innego, czego potrzebujesz do publikacji. Konfiguracja EWS zawiera klasę dokumentu używaną w wersji papierowej tej książki (`ews`). Szczegółowe omówienie tej zmiennej wykracza poza zakres tej książki. W załączniku omówiono szczegółowo konfigurację EWS. Przeczytaj dokumentację tej zmiennej za pomocą `C-h v org-latex-classes`.

### *Obrazy i tabele*

Org konwertuje obrazy i tabele na obiekty pływające LaTeX. Można dodać określone atrybuty do tych obiektów pływających, używając słowa kluczowego `#+attr_latex:`, jak pokazano na poniższym przykładzie obrazu:

```
#+caption: To jest przykładowy podpis do obrazu. #+attr_latex:
:width 5cm :options angle=90 :placement h
[[katalog/nazwa_pliku.png]]
```

Inne parametry rozmiaru to `:height` i `:scale`. Parametr rozmiaru można podać w centymetrach (cm) lub calach (in), a także w innych formatach rozmiaru akceptowanych przez LaTeX. Parametr ten można również wyrazić w wymiarach względnych, na przykład `0,5\textwidth`, co oznacza połowę szerokości kolumny tekstu.

Parametr `:float` zapewnia kilka opcji umieszczania obrazu lub tabeli:

- `multicolumn`: rozciąga obraz na wiele kolumn.
- `wrap`: Tekst otaczający obraz po prawej stronie.



- sideways: dla nowej strony z obrazem obróconym o dziewięćdziesiąt stopni.

Tabele można modyfikować w ten sam sposób, korzystając z szerokiej gamy opcji. Proszę zapoznać się z odpowiednią sekcją w podręczniku Org, wpisując C-h g org <ret> g table latex.

### *Cytaty*

Ostatni procesor cytatów dostępny w Org łączy się bezpośrednio z LaTeX, który oferuje trzy opcje:

- BibTeX obsługuje tylko polecenia LaTeX `\cite` i `\nocite`.
- NatBib pozwala na więcej wariantów stylistycznych niż standardowe polecenie cytowania LaTeX.
- BibLaTeX jest alternatywnym procesorem bibliograficznym.

Poniższy przykład pokazuje, jak wywołać procesor NatBib przy użyciu stylu cytowania Harvard, zgodnie z wytycznymi wydawnictwa Wolters-Kluwer. Pierwsza część określa procesor NatBib z bibliografią i stylem cytowania Kluwer.

```
#+cite_export: natbib kluwer
```

Podczas gdy podstawowe procesory cytowania i procesory CSL eksportują w pełni sformatowane ciągi tekstowe, te trzy procesory eksportują polecenia LaTeX, takie jak `\cite{nietzsche_1883}`. Oznacza to, że nie można używać tych procesorów podczas eksportowania do innych formatów, ponieważ wynik będzie wyglądał jak kod LaTeX. Jeśli potrzebujesz skonfigurować manuskrypt dla wielu formatów, idealnym wyborem będzie procesor CSL.

### *Fragmenty LaTeX*

Możesz wpisywać proste polecenia LaTeX bezpośrednio do pliku org. Zostaną one wyeksportowane bez zmian do pliku TEX. Na przykład, aby zdefiniować początek przedniej i tylnej części dokumentu

1

rok i

5

nie użyj poleceń LaTeX \frontmatter i \backmatter w pliku Org w odpowiednich miejscach.

iecy

Ta metoda działa idealnie, ale nie jest najlepsza, gdy eksportujesz plik do wielu formatów.

Nie chcesz, żeby te polecenia LaTeX zaśmiecały inne Twoje pliki wyjściowe. Najlepiej byłoby, gdyby te polecenia były w szablonie struktury. Naciśnij C-c C-, l, żeby wstawić blok źródłowy.

Dzięki tej metodzie możesz wstawić blok źródłowy dla każdego formatu wyjściowego.

Jeśli musisz wielokrotnie wstawiać te same złożone fragmenty, rozważ użycie makr opisanych w sekcji 8.2 dla fragmentów LaTeX.

### *Maevos fov wiele formatów fov*

Jeśli eksportujesz ten sam manuskrypt do wielu formatów, możesz zdefiniować jedno makro, które będzie miało zastosowanie do wszystkich tych formatów eksportu. Wielokrotne rozszerzenia makr są otoczone podwójnymi symbolami ampersand (@@), po których następuje format eksportu i dwukropek, na przykład: @@latex:. Poniższy przykład tworzy makro zmieniające kolor tekstu dla eksportów HTML, LaTeX i ODT.

```
#+macro: hl @@html:$2@@
@@latex:\textcolor{$1}{$2}@@
@@odt:<text:span text:style-name="$1">$2</text:span>@@
```

Makro to stosuje się w następujący sposób: {{{hl(red, Ten tekst jest czerwony.)}}}. Podczas eksportowania tego makra do formatu HTML tekst zostanie wyświetlony na czerwono lub w kolorze określonym przez pierwszy parametr. Podczas eksportowania do formatu LaTeX zostanie on wydrukowany na czerwono przy użyciu pakietu xcolor. To samo makro można również zastosować do eksportów ODT, ale należy zdefiniować styl znaków o tej samej nazwie co \$1 w pliku szablonu, w tym przypadku „red”.

## **8.7 Dalsze badania**

W niniejszym rozdziale przedstawiono jedynie ogólny zarys możliwości eksportu programu Org dla poszczególnych formatów. Dostępne są dodatkowe opcje konfiguracyjne. Można skonfigurować proces eksportu

procesu eksportu manuskryptu na dwóch poziomach:

1. Zmienny (wszystkie pliki Org)
2. część wstępna (eksportowany plik)

Na przykład zmienna `org-export-with-tables` określa, czy eksport obejmuje tabele (domyślnie jest to `true`). Można nadpisać globalne ustawienie dla eksportowanego pliku za pomocą `#+options |:nil`.

Org oferuje również szereg formatów eksportu, które nie zostały opisane w tym rozdziale. Można eksportować do formatu Markdown lub pliku tekstowego. Można nawet eksportować plik do trybu Org. Ta opcja analizuje wszystkie cytaty i łączy wszystkie dołączone pliki w jeden dokument. Istnieją dodatkowe pakiety dla innych formatów eksportu. Pakiety te zazwyczaj zaczynają się od `ox`, np. `ox-ePub` omówiony w sekcji 8.5.

Podręcznik Org omawia wszystkie funkcje dostępne do eksportowania, które można znaleźć za pomocą `C-h R org <ret> g exporting`.



---

# **Administracja: zarządzanie projektami**

---

Podstawową działalnością autora jest prowadzenie badań, pisanie, redagowanie i publikowanie swoich prac. Ale to nie wszystko. Pisanie wiąże się również z zadaniami administracyjnymi, takimi jak dotrzymywanie terminów, zarządzanie projektami i utrzymywanie uporządkowanego systemu archiwizacji. Emacs zapewnia rozbudowaną funkcjonalność do wykonywania tych zadań.

W tym rozdziale wyjaśniono, jak używać trybu Org do zarządzania projektami i listami zadań, luźno wdrażając metodę Getting Things Done (GTD). GTD to tylko jedna z wielu metod zarządzania intensywnym życiem. Piękno Emacsa polega na tym, że masz swobodę wdrażania dowolnej preferowanej metody. W kolejnych dwóch sekcjach pokazano, jak zarządzać plikami za pomocą edytora katalogów (Dired) i obrazami za pomocą pakietu Image-Dired.

## **y.1 Getting Things Dome**

W dzisiejszym świecie, w którym brakuje czasu, każdy chce upchnąć jak najwięcej rzeczy w ciągu dnia. Sądząc po obfitej literaturze, można stać się tak produktywnym, że „załatwianie spraw”

zajmuje tylko „cztery godziny pracy tygodniowo” (Allen, 2005; Ferriss, 2011). Osiągamy te niemal magiczne moce, ucząc się od „nawyków wysoce skutecznych ludzi”, którzy wydają się „jeść żaby” na śniadanie (Covey, 2000; Tracy, 2015).

W Internecie można znaleźć wiele metod zwiększających osobistą produktywność, a książki zawierające dobre rady są łatwo dostępne. Rynek jest również pełen oprogramowania umożliwiającego wdrożenie tych systemów, od sprawdzonych gigantów, takich jak OneNote firmy Microsoft, po zwinne aplikacje mobilne, takie jak Todoist.

Wszystkie te metody sprowadzają się do trzech podstawowych etapów: wyznaczenia celu, określenia działań niezbędnych do jego osiągnięcia oraz podjęcia tych działań. Zasady te mogą wydawać się uproszczone, ale stanowią fundamentalną prawdę. Wiele opublikowanych metod omawia różne sposoby skutecznego i wydajnego zarządzania tymi trzema etapami.

Pakiet Org nie tylko umożliwia pisanie i publikowanie tekstów, ale może również pomóc w zarządzaniu działaniami i projektami. Podobnie jak każdy inny pakiet Emacs, oferuje nieograniczoną swobodę wdrażania preferowanej metody zwiększania produktywności. Ta swoboda pozwala przejąć kontrolę nad zadaniami i projektami, wdrażając metodę, która najlepiej pasuje do danej sytuacji.

W tej sekcji pokazujemy, jak wykorzystać Org do zarządzania projektami i zadaniami, czerpiąc inspirację z metody Davida Allena *Getting Things Done* (GTD). David Allen opisuje pięć iteracyjnych kroków, które pozwalają zwiększyć produktywność (Allen, 2005):

1. *Capture*: Oczyść umysł i zapisz wszystko.
2. *Klasyfikuj*: Zdecyduj, co zrobić z zadaniami.
3. *Organizuj*: Umieść zadania w odpowiednich miejscach.
4. *Reflect*: Zastanów się nad swoimi postępami.
5. *Engage*: Podejmij działanie.

### *Captuve: Opróżnij swój umysł*

Jednym z powodów, dla których często jesteśmy mniej produktywni, niż byśmy chcieli, jest to, że nasze umysły są przepełnione nieistotnymi sprawami. Kreatywne i produktywne myśli są tłumione, gdy mózg jest przepełniony myślami o tym, co należy zrobić. Kolejnym problemem związanym z przechowywaniem pomysłów w głowie jest ryzyko, że ostatecznie popadną one w zapomnienie. Jestem pewien, że wszyscy znacie sytuację, w której najpiękniejszy pomysł przychodzi wam do głowy podczas porannej kąpieli. Jednak pół godziny później nie potraficie sobie przypomnieć tej perełki.

Pierwszy krok do osiągnięcia celu jest zaskakująco prosty: opróżnij swój umysł. Nie jest to buddyjska droga do oświecenia, ale prosta technika pomagająca się skupić. Otwórz nowy plik Org i nazwij go np. `todo.org` lub jakkolwiek chcesz. Możesz również użyć skrzynki odbiorczej, jak wyjaśniono w sekcji 5.2.

Przez następne piętnaście minut zapisuj wszystko, co przychodzi Ci do głowy. Wrzuć całą zawartość swojego umysłu do tego nowego pliku Org. Każdy nowy pomysł oznacz gwiazdką, aby stał się nagłówkiem. Twoja lista będzie zawierała mieszankę przypadkowych rzeczy, od drobnych obowiązków domowych po duże projekty, które chciałbyś kiedyś zrealizować. Nie filtruj swoich myśli, po prostu je zapisuj.

Nie wykonuj wielu zadań jednocześnie! Zamiast tego poświęć tej czynności całą swoją uwagę. Pamiętaj, że wielozadaniowość jest wrogiem produktywności, ponieważ nasz mózg może skupić się tylko na jednej czynności intelektualnej naraz. Fakt, że magicy z łatwością potrafią oszukać ludzi, ilustruje, dlaczego wielozadaniowość jest bezsensownym zajęciem (Prevos, 2013). Być może warto posłuchać muzyki za pomocą pakietu Emacs EMMS, aby zachować koncentrację.

Nie martw się o to, kiedy będziesz musiał to zrobić ani w jakiej kolejności; tym zajmiesz się później. Upewnij się, że po ukończeniu listy Twój umysł jest pusty. Dla większości ludzi pięćdziesiąt zadań do wykonania, projektów i niejasnych celów nie jest niczym niezwykłym. Jeśli czujesz się przytłoczony, nie obwiniaj posłańca, ponieważ lista odzwierciedla Twoje życie. Masz teraz długą listę wszystkiego, co musisz zrobić.

\* Skosić trawnik

\* Posprzątaj podwórko

- \* Poprawić swoje umiejętności zawodowe
- \* Nauczyć się obsługi programu Emacs
- \* Napisać ebooka o... (wpisz swoją specjalizację)
- \* Opróżnij skrzynkę odbiorczą poczty elektronicznej
- \* Przygotować prezentację na kwartalne spotkanie w przyszłym tygodniu
- \* I tak dalej, i tak dalej...

*Clavify: Desevibe wlat it meaes*

Czy zauważyłeś, że większość pozycji z powyższej listy wymaga więcej wysiłku niż tylko wykonanie jednej czynności? W języku GTD są to projekty. Projekt ma określony wynik i wymaga więcej niż jednej czynności, aby go zrealizować. Inne pozycje na liście mogą być celami. Cel, taki jak nauka nowego języka, jest mniej określony niż projekt i ma charakter bardziej aspiracyjny niż projekt. Twoja lista będzie zawierała zadania, projekty, cele i niejasne pomysły.

Twoim następnym zadaniem jest wyjaśnienie tego, co zapisałeś. Najpierw uporządkuj listę w odpowiedniej hierarchii, używając klawiszy Alt i strzałek. M-<w górę> i M-<w dół> przesuwały nagłówek w górę lub w dół, a M-<w prawo> i M-<w lewo> przenoszą wpis wyżej lub niżej. Za pomocą tych czterech klawiszy możesz uporządkować listę, wprowadzając porządek w chaosie, który panuje w Twojej głowie. W naszym przykładzie ogród staje się projektem składającym się z dwóch zadań. Pod dowolnym nagłówkiem można dodawać notatki lub linki do obrazów i innych dokumentów, aby zapewnić kontekst zadania.

- \* Ogrodnictwo
- \*\* Posprzątaj ogród
  - Opróżnij szopę
  - Wysypisko śmieci
- \*\* Skoszenie trawnika

Niektóre pozycje z listy mogą stać się częścią listy kontrolnej, która przypomni Ci o wymaganych czynnościach. Org może utworzyć pozycje z polem wyboru w dowolnym miejscu dokumentu, dodając [ ] po wskaźniku listy. Użycie M-S <Ret> po pozycji z polem wyboru tworzy nową listę.



element z polem wyboru. Zaznaczenie lub odznaczenie dowolnego elementu jest tak proste, jak naciśnięcie C-c C-c

(org-toggle-checkbox).

Możesz zamienić zwykłą pozycję listy na pozycję z polem wyboru lub odwrotnie, używając C-u C-c C-c. Dwa argumenty uniwersalne dodają poziomą linię przez pole, żeby pokazać, że ta pozycja nie jest już aktualna (C-u C-u C-c C-c).

Jeśli masz długą listę, rozważ dodanie wskaźnika pokazującego postępy lub ich brak. Możesz dodać tzw. plik cookie. Dodaj znak [/] lub [0%] w linii nad polami wyboru, a Org zarejestruje Twoje postępy przy następnej zmianie pozycji, jak pokazano w poniższym przykładzie.

```
* Ogrodnictwo
** Posprzątaj podwórko [50%]
 - [X] Opróżnić szepę
 - [] Wyrzucić śmieci
** Skosić trawnik
```

Org pozwala łatwo konwertować elementy listy na nagłówki i odwrotnie. Skrót klawiaturowy C-c C-\* (org-ctrl-c-star) konwertuje akapit na nagłówek. Jeśli wiersz zawiera pole wyboru, staje się on zadaniem do wykonania. Użycie C-c C-- (org-ctrl-c-minus) konwertuje akapit lub nagłówek na element listy.

Problem z większością list rzeczy do zrobienia polega na tym, że szybko mogą one stać się przytłaczające. Większość działań nie wymaga natychmiastowej realizacji lub nie może być realizowana od razu. Podstawową zasadą podejścia GTD jest posiadanie listy działań, którą można łatwo zarządzać. W ramach tej metodologii zadanie jest albo wpisywane do kalendarza, oznaczane jako kolejne zadanie do wykonania w najbliższym czasie, albo delegowane. Lista kolejnych zadań stanowi zbiór spraw, które musisz wykonać. Metoda ta gwarantuje, że w Twojej świadomości pozostaje tylko część zadań, dzięki czemu nie czujesz się przytłoczony brakiem aktywności, ponieważ listy zadań są zazwyczaj obietnicą złożoną przyszłej wersji siebie. Przyjrzyjmy się dwóm ostatnim opcjom: kolejnym zadaniom i delegowaniu.

Org może powiązać każdy nagłówek ze statusem przepływu pracy, wskazanym na początku nagłówka, zazwyczaj wielkimi literami, jak pokazano poniżej.

## \*\* TODO Skosić trawnik

Domyślnie istnieją tylko dwa stany: TODO (do zrobienia) lub DONE (zrobione). Jednak aby zwiększyć swoją koncentrację, potrzebujesz więcej niż tylko tych dwóch stanów, ponieważ nie ma możliwości rozróżnienia ważności zadań. Niektóre zadania zależą od wykonania innych lub mogą zostać wykonane w przyszłości. Możesz ustawić różne słowa kluczowe dla każdego pliku Org w sekcji front matter. Poniższa linia definiuje przepływ pracy z pięcioma oddzielnymi fazami.

```
#+todo: TODO(t) NEXT(n) WAIT(w) | DONE(d) CANCELLED(c)
```

Pierwszy stan oznacza, że w przyszłości należy podjąć jakieś działanie (TODO). Gdy nagłówek jest działaniem NEXT, należy je wykonać jak najszybciej. Czasami nie można podjąć działania, ponieważ czekasz na kogoś innego lub zadanie zostało delegowane (WAIT). Statusy po pionowej kresce oznaczają stan zakończenia, co w tym przykładzie oznacza DONE lub CANCELLED. Oczywiście możesz dostosować przepływ zadań do swoich potrzeb. Słowa kluczowe są zazwyczaj zapisywane wielkimi literami, ale nie jest to obowiązkowe.

Org przełącza się między dostępnymi stanami za pomocą klawiszy Shift i strzałek w lewo lub w prawo. Stan można również zmienić za pomocą org-todo (C-c C-t). Gdy dostępnych jest więcej niż dwie opcje, polecenie to wyświetla okno podręczne. Litera w nawiasie oznacza skrót klawiaturowy dla menu podręcznego.

Nie wszystkie zadania muszą być realizowane jak najszybciej. Emacs posiada kalendarz, a Org może planować zadania i wyznaczać terminy. Zaplanowanie zadania na konkretną datę zobowiązuje przyszłą wersję siebie do podjęcia działania w tym terminie. Termin wskazuje, kiedy zadanie musi zostać wykonane ze względu na oczekiwania zewnętrzne.

Aby dodać zaplanowaną datę, użyj C-c C-s, gdy kursor znajduje się na odpowiednim nagłówku (org-schedule). Emacs wyświetli kalendarz, za pomocą którego można wybrać datę. Klawisze Shift i strzałki przesuwają znacznik czasu o dzień lub tydzień. Klawisze < i > pozwalają przesuwać się o miesiąc w czasie. Po zakończeniu naciśnij Enter, a data pojawi się pod nagłówkiem. W ten sam sposób można dodać termin, używając skrótu klawiszowego C-c C-d.

(org-deadline). Zadanie zaplanowane musi zostać wykonane w określonym dniu, natomiast termin to zadanie, które musi zostać wykonane do tego dnia.

Znaczniki czasu mają format ISO 8501: rok, miesiąc, dzień. Format ten eliminuje niejasności między formatami amerykańskimi a formatami stosowanymi w pozostałych częściach świata, ułatwiając sortowanie dat. Edytowanie znacznika czasu jest łatwe. Umieść kursor na roku, miesiącu lub dacie i użyj klawiszy strzałek, aby przesunąć go w górę lub w dół i zmodyfikować znacznik czasu.

\* TODO Ukończ artykuł Org-Mode  
PLANOWANE: <2021-05-08 sobota>

Zadania mogą mieć również powtarzające się harmonogramy lub terminy. Na przykład cotygodniowy przegląd. Po dodaniu +7d na końcu daty Org przelicza datę za każdym razem, gdy zadanie zostanie wykonane, i resetuje status do TODO. Można również użyć liter w, m i y, aby zaplanować zadanie cotygodniowe, comiesięczne lub coroczne.

W poniższym przykładzie siedem dni jest przeliczane za każdym razem, gdy wykonasz zadanie. Tak więc za każdym razem, gdy zakończysz ten przegląd, nową datą będzie 7 lipca, siedem dni po pierwotnie zaplanowanej dacie.

\* TODO Cotygodniowy przegląd  
skrzynki odbiorczej ZAPLANOWANE:  
<2024-06-30 niedziela +7d>

Powyższa metoda jest dopuszczalna, jeśli zadania zostaną wykonane w pobliżu zaplanowanej daty. Na przykład, jeśli zadanie zostanie wykonane po 7 lipca, nowy termin będzie już minął. Możliwe jest również przełożenie zadania na określony okres po jego wykonaniu za pomocą podwójnego znaku plus. W poniższym przykładzie do zaplanowanej daty dodaje się siedem dni, gdy status działania zmienia się na WYKONANE. Załóżmy na przykład, że zadanie zostanie wykonane 20 lipca. W takim przypadku nowa zaplanowana data to niedziela, 21 lipca, z zachowaniem cotygodniowej częstotliwości. W tym przypadku nowa zaplanowana data będzie zawsze przypadać w niedzielę.

\* TODO Cotygodniowy przegląd  
PLANOWANE: <2024-06-30 niedziela ++7d>

Dodanie kropki i znaku plus (+) przed częstotliwością powtarzania powoduje przesunięcie nowej daty zaplanowanej czynności o siedem dni od ostatniego wykonania. Ten rodzaj powtarzania jest idealny, gdy chcesz wykonywać czynność co kilka dni, miesięcy lub lat, niezależnie od tego, kiedy została ona ostatnio wykonana.

\* TODO Cotygodniowy przegląd

PLANOWANE: <2024-06-30 niedziela .+7d>

Dodaj datę zaplanowaną tylko wtedy, gdy jest to czas, w którym planujesz podjąć działanie. Staraj się unikać przeładowywania swojego harmonogramu samodzielnie wyznaczonymi zadaniami. Lepiej jest ustawić zadanie jako następne działanie i określić, kiedy je wykonać, podczas regularnych przeglądów. Termin jest pomocny tylko wtedy, gdy istnieje zewnętrzne oczekiwanie, że musisz coś wykonać w określonym terminie, na przykład wykupić ubezpieczenie podróże przed odlotem.

Nie dotrzymanie terminu może mieć konsekwencje, dlatego warto otrzymać ostrzeżenie z wyprzedzeniem, gdy termin ten się zbliża. Poniższy przykład określa termin nadchodzącego całkowitego zaćmienia Słońca w Australii. Chcę wziąć udział w tym wydarzeniu, więc plik cookie -12m zapewnia, że termin ten zostanie dodany do mojego kalendarza na rok przed jego wystąpieniem, co pozwoli mi zorganizować wycieczkę na Pustynię Centralną. Więcej informacji na temat kalendarza znajduje się w sekcji y.1.

\* Zaćmienie Słońca w środkowej

Australii PLANOWANE: <2028-07-22

sobota -12m>

Dostępne są bardziej zaawansowane harmonogramy, ale wymagają one podstawowej znajomości języka Lisp i użycia polecenia diary-float. Poniższy przykład planuje zadanie na każdy pierwszy wtorek (dzień 2 tygodnia) każdego miesiąca. Pierwszym parametrem może być numer miesiąca lub t dla wszystkich miesięcy. Drugi parametr (2) wskazuje dzień tygodnia, a ostatni parametr (1) to liczba dni w miesiącu. Tydzień w Emacs rozpoczyna się w niedzielę (dzień 0) i kończy w sobotę (dzień 5).

\* Przegląd miesięczny

PLANOWANE: <%(diary-float t 2 1)>

Kluczem do skutecznego wdrożenia każdej metody zwiększającej produktywność jest niebycie dla siebie nadmiernie wymagającym rodzicem. O ile nie ma zewnętrznych terminów, wszystko na mojej liście zadań jest opcjonalne. Wystawienie odpowiedniego kosza na śmieci na krawężnik nie jest opcjonalne, ponieważ polegam na tym, że śmieciarka go odbierze. Jednak napisanie nowego rozdziału tej książki jest opcjonalne, ponieważ nie ma zewnętrznego terminu. Mądrze wykorzystuj możliwości zarządzania swoim życiem za pomocą Org. Nie stań się niewolnikiem listy i nie trać poczucia własnej wartości, ponieważ nie jesteś w stanie sprostać własnym oczekiwaniom.

*Ovgaeise: Umieść to tam, gdzie należy*

W poprzednich dwóch sekcjach omówiono ustalanie punktu wyjścia dla listy działań. Aby skupić się na istotnych działaniach, należy regularnie przeglądać swój system. Dobrym nawykiem jest cotygodniowy przegląd, zwłaszcza w dynamicznych środowiskach.

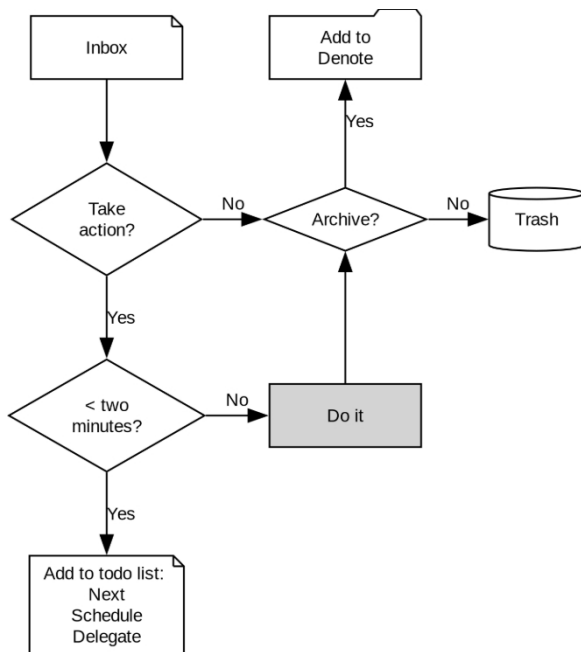
Schemat na rysunku y.1 przedstawia typowy przebieg pracy związanej z zarządzaniem cyfrowym życiem. Wszystko, co przyciąga Twoją uwagę, jest wysyłane do skrzynki odbiorczej jako krótka notatka lub nowe działanie. Skrzynka odbiorcza to nie tylko plik przechwytyjący Org, ale także skrzynka odbiorcza poczty elektronicznej, fizyczna skrzynka odbiorcza lub cokolwiek innego. Przeglądając każdą pozycję, zadaj sobie pytanie, czy można ją wykonać szybko; jeśli tak, to ją wykonaj. Jeśli jest to działanie, które zajmuje nieco więcej czasu, dodaj je do listy rzeczy do zrobienia (kolejne działanie, zaplanowane lub delegowane), jak opisano w poprzedniej sekcji.

Wszystko, co nie wymaga podjęcia działań lub zostało już zrealizowane, można zarchiwizować w systemie Denote, jak opisano w rozdziale 5, lub od razu przenieść do cyfrowego kosza.

*Refleet: Moeitov pvogvess*

Kluczem do wydajnego przepływu pracy jest regularne przeglądanie swoich działań, priorytetów i celów oraz podejmowanie odpowiednich działań. Twoja lista zadań lub wiele list jest uporządkowanych w logiczny sposób. Możesz sortować różne działania według daty, typu lub tagu. Agenda jest głównym narzędziem w Org służącym do porządkowania listy zarejestrowanych działań.

Agenda to podsumowanie działań w pliku (plikach) zadań do wykonania, uporządkowane według czasu. Najpierw należy dodać odpowiednie pliki do listy agendy. Dodaj plik powiązany z bieżącym buforem



Rysunek 9.1: Przykład przepływu pracy w programie Zmaes.

do agendy za pomocą `org-agenda-file-to-front` (C-c []). Możesz dodać wiele plików zadań do listy agendy, na przykład skrzynkę odbiorczą, plik zadań dotyczących życia osobistego i plik zadań dotyczących pracy. Plik można usunąć z listy agendy za pomocą `org-remove-file` (C-c []). Gdy pliki agendy znajdują się na liście, można szybko do nich przejść za pomocą C-' (`org-cycle-agenda-files`). Polecenie to przechodzi cyklicznie przez zarejestrowane pliki agendy, zapewniając do nich szybki dostęp.

Funkcja kalendarza w Org jest bardzo wszechstronna i konfigurowalna. Po wywołaniu funkcji `org-agenda` za pomocą C-c a, Org wyświetla menu z różnymi punktami dostępu do list działań.

Agenda na bieżący tydzień (C-c a a) pokazuje wszystkie działania zaplanowane na ten okres, które mają termin wykonania. Lista ta zawiera również wszelkie działania przeterminowane i terminy w okresie ostrzegawczym. Po agendzie można poruszać się za pomocą klawiszy strzałek i klawisza Tab.

lub Ret przenosi Cię do działania, umożliwiając jego edycję. Przycisk v pozwala wygenerować dziennik dla bieżącego dnia, tygodnia, dwóch tygodni, miesiąca lub roku.

Menu zapewnia również dostęp do listy wszystkich zadań do wykonania. Listę tę można filtrować według statusu, aby wyświetlić na przykład tylko kolejne zadania. Wpisz numer wyświetlony w górnej części okna i wpisz r.

Agenda Org posiada rozbudowane możliwości dostosowywania sposobu podsumowywania informacji. Można na przykład zdefiniować własne pozycje menu, aby wyświetlać agendę zadań prywatnych i służbowych. Konfiguracja EWS zawiera przykład, który pokazuje agendę na najbliższe trzy dni, wszelkie kolejne działania oraz listę spraw, na które czekasz. Dostęp do tej konstrukcji uzyskuje się za pomocą C-c a e i może ona wyglądać mniej więcej tak:

Agenda na 3 dni (W32):

Środa            7 sierpnia 2024

Do zrobienia: 159 dni temu: TODO Kontrola termitów

TODO:            Zaplanowane: TODO Skrzynka odbiorcza

Zero

Do zrobienia:            Zaplanowano 17x: TODO

Sprzątanie biblioteki w czwartek            8 sierpnia  
2024

Piątek            9 sierpnia 2024

Do zrobienia: Zaplanowane: TODO Kosz na śmieci

Jednak tworzenie agendy wymaga pewnych umiejętności kodowania w języku Emacs Lisp, więc wykracza to poza zakres tej książki. W załączniku wyjaśniono, jak tworzona jest agenda EWS. Podręcznik Org zawiera obszerną dokumentację (C-h R org <Ret> g agenda views).

### *Zegage: Weż aetioe*

Emacs to potężne, wielofunkcyjne narzędzie, które nie pomoże Ci skosić trawnika, zrobić zakupów ani wykonać żadnych innych zadań z Twojej listy. Należy jednak pamiętać, że żaden system zwiększający produktywność nie wykona zadań za Ciebie, niezależnie od tego, co obiecują guru. Jednak uporządkowanie swojego życia ma kluczowe znaczenie dla skupienia się na celach.

Zaznaczenie pola lub oznaczenie działania jako wykonanego powoduje satysfakcjonujący wyrzut dopaminy. Po okresie ciężkiej pracy Twoja lista rzeczy do zrobienia będzie wypełniona wykonanymi działaniami i projektami. Org posiada pewne funkcje umożliwiające archiwizowanie tych informacji i porządkowanie starych danych.

Polecenie `org-archive-subtree (C-c C-x C-s)` przenosi zawartość poddrzewa kursora do innego pliku. Domyślną nazwą pliku archiwum jest bieżący plik z dodanym rozszerzeniem `_archive`. Polecenie archiwizacji dodaje takie właściwości, jak data archiwizacji i oryginalna nazwa pliku. Nazwę archiwum można dostosować, dodając `#+archive:` i odpowiednią nazwę pliku jako słowo kluczowe.

### *Leaveieg move*

Podręcznik Org (`C-h R org`) zawiera dodatkowe informacje na temat możliwości zarządzania projektami w tym głównym trybie, takich jak ustalanie priorytetów i rejestrowanie czasu pracy, które nie zostały omówione w niniejszej książce.

Bawarski guru Org, Rainer König, opracował obszerną serię filmów na YouTube, które szczegółowo wyjaśniają wykorzystanie trybu Org do zarządzania działaniami i projektami. Ranier opublikował również obszerny kurs na platformie Udemy, który zawiera więcej szczegółów niż bezpłatne filmy (König, 2020).

## **y.2 Pliki Mamage**

Praca z Emacsem oznacza, że będziesz musiał uzyskiwać dostęp do plików na swoich dyskach, tworzyć je i zarządzać nimi. Emacs jest dostarczany wraz z Dired (wymawiane *div-ed*, od directory editor), potężnym menedżerem plików. Dired ma bogatą historię. Pierwsza wersja dired była samodzielnym programem napisanym około 1974 roku, więc jego początki sięgają jeszcze dalej w przeszłość niż Emacs.

W tej sekcji wyjaśniono podstawowe zasady korzystania z programu Dired i pokazano, jak zarządzać plikami w celu uporządkowania dysku twardego. Przedstawiono również pakiet Image-Dired, który pomaga przeglądać kolekcje obrazów i fotografii.



### *Otwórz pliki diveetovies aed*

Menedżera plików aktywuje się za pomocą funkcji `dired` lub skrótu `C-x d`. Po wybraniu odpowiedniego katalogu w minibufferze Emacs tworzy nowy bufor z zawartością określonego katalogu. Inną przydatną funkcją jest `dired-jump` (`C-x C-j`), która otwiera `Dired` i przenosi kursor bezpośrednio do pliku powiązanego z bieżącym buforem. Górna linia pokazuje ścieżkę bieżącego katalogu i dostępną przestrzeń dyskową. Poniżej tej linii znajduje się lista wszystkich plików i folderów. EWS umieszcza wszystkie podkatalogi na górze, a następnie lista plików uporządkowana alfabetycznie.

Pierwsza kolumna w buforze `Dired` pokazuje typ pliku i uprawnienia. Dwie kolejne kolumny pokazują rozmiar pliku i datę ostatniego zapisania. Ostatnia kolumna pokazuje nazwę pliku. Poniższy przykład pokazuje pierwsze linie tej książki w buforze `Dired`. Użyj lewego nawiasu ( , aby usunąć informacje techniczne i pokazać tylko katalogi i nazwy plików.

```
/home/peter/documents/projects/ews/: (286 GiB dostępne) drwxrwxrwx 6 4,0K
2024-07-21 14:05 .
drwxrwxrwx 7 4,0K 2024-07-19 08:23 ..
drwxrwxrwx 2 4,0K 2024-07-17 06:33 images
-rw-r--r-- 1 5,3K 2024-07-20 19:51 00-emacs-writing-studio.org
-rw-r--r-- 1 3,7K 2024-07-20 08:51 00-i-foreword.org
-rw-r--r-- 1 2,5K 2024-05-04 16:47 00-ii-preface.org
-rw-r--r-- 1 24K 2024-07-20 20:04 01-why-emacs.org
```

Możesz poruszać się po zawartości za pomocą klawiszy strzałek lub nacisnąć klawisz `j`, aby przejść do określonego pliku, wpisując część jego nazwy w minibufferze i wybierając ten, który chcesz otworzyć. Plik lub podkatalog otwierasz klawiszem `Enter`. Aby otworzyć plik lub katalog w innym oknie, naciśnij klawisz `o`. Klawisz `C-o` otwiera plik w innym oknie, ale kursor pozostaje w oknie `Dired`. Klawisz `q` zamyka okno `Dired`, ale nie usuwa go.

Należy pamiętać, że za każdym razem, gdy otwierasz katalog, Emacs otwiera nowy bufor `Dired`. Po pewnym czasie sesja Emacs'a zostaje zaśmiecona nieużywanymi buforami `Dired`. Naciśnięcie klawisza `a` zamiast `Enter` otwiera katalog w tym samym buforze.

Domyślne ustawienie Emacs'a pokazuje ukryte pliki, nawet jeśli są one ukryte z jakiegoś powodu. EWS zapewnia pewną wygodę, usuwając ukryte pliki z widoku. Użyj kropki (.) aby przełączyć to zachowanie i wyświetlić lub ukryć ukryte pliki.

Możesz otwierać pliki za pomocą zewnętrznego oprogramowania, naciskając klawisz &, po czym Dired poprosi o podanie nazwy odpowiedniego programu. Musisz wpisać nazwę pliku wykonywalnego oprogramowania, którego chcesz użyć, np. gimp.

### *Modyfikowanie plików diveetovies aed*

Aby skopiować plik, naciśnij przycisk C. Dired poprosi o podanie nowego katalogu i nazwy w minibufferze. Aby przenieść plik, naciśnij przycisk R, ponieważ przeniesienie pliku jest równoznaczne z zmianą jego nazwy w nowym katalogu. Nie musisz zamykać bufora przed zmianą nazwy otwartego pliku. Emacs połączy otwarty bufor z nową nazwą pliku. Jeśli masz dwa otwarte bufory Dired w ramce, EWS kopiuje i przenosi z folderu w aktywnym oknie do drugiego bufora Dired.

Czasami warto skopiować nazwę pliku do bufora kill ring za pomocą klawisza w, żeby móc go użyć do zmiany nazwy pliku. Żeby zmienić nazwę pliku, skopiuj nazwę za pomocą klawisza w, zmień nazwę pliku za pomocą klawisza R, wklej istniejącą nazwę za pomocą klawiszy C-y i zmień nazwę na nową wersję.

Wybierz i odznacz pliki do usunięcia za pomocą przycisków d i u. Po wybraniu plików, które chcesz usunąć, naciśnij x, aby wykonać usunięcie. Naciśnij wielką literę D, jeśli chcesz usunąć pojedynczy plik. Emacs poprosi Cię również o zamknięcie odpowiedniego bufora podczas usuwania lub przenoszenia do kosza aktualnie otwartego pliku.

Możesz wybrać wiele plików do jednoczesnej pracy, zaznaczając je. Przycisk m zaznacza plik, a u usuwa zaznaczenie. Wielka litera U usuwa wszystkie zaznaczenia w buforze. Użycie klawisza t odwraca istniejące zaznaczenia, co jest pomocne, gdy chcesz wybrać wszystko oprócz jednego lub dwóch plików.

Ta metoda wymaga ręcznego zaznaczenia każdego pliku. Można również użyć wyrażeń regularnych do zaznaczenia plików. Naciśnij % m, aby otworzyć okno wyboru wyrażeń regularnych. Na przykład ^2023.\*\_journal\* zaznacza wszystkie pliki Denote, które zaczynają się od 2023 i mają

tagiem pliku dziennika. Teraz naciśnij **klawisz t**, aby odwrócić wybór, a następnie **k**, aby usunąć wybrane pliki z widoku. Ta sekwencja jest przydatną metodą wyszukiwania powiązanych plików.

Funkcja `dired-narrow-regexp` z pakietu o tej samej nazwie autorstwa Matúša Gol-jera zapewnia wygodniejszą metodę zawężania listy plików. Funkcja ta wyświetla monit o podanie wyrażenia regularnego i wyświetla tylko te pliki, które pasują do niego. Naciśnięcie klawisza **g** powoduje wyczyszczenie filtra.

Kolejną przydatną funkcją programu `Dired` jest możliwość wyszukiwania i zamiany wyrażeń regularnych w zestawie zaznaczonych plików. Pierwszym krokiem jest zaznaczenie plików, w których chcesz wyszukiwać, pojedynczo za pomocą klawisza **m** lub wyszukując wyrażenie regularne za pomocą klawisza **%m**, jak opisano powyżej. Użyj klawisza **Q**, aby rozpocząć sesję wyszukiwania i zamiany wyrażeń regularnych w zaznaczonych plikach. Teraz możesz wyszukać wyrażenie regularne i zamienić je na nowy ciąg znaków. Bufor zostanie podzielony na trzy części. Użyj znaku zapytania, aby wyświetlić wszystkie opcje.

### *Plik-eamieg eoeveetioes*

Kiedy biura korzystały z archiwów papierowych, przestrzegały surowych zasad archiwizacji dokumentów. Zagubienie kartki papieru wśród metrów półek wypełnionych teczkami oznaczało, że nigdy więcej nie uda się znaleźć tego dokumentu. Kiedy w latach 80. pracownicy biurowi zaczęli korzystać z komputerów, zrezygnowano z tych rygorystycznych zasad i procesów. Nieograniczona swoboda komputerów spowodowała chaos, ponieważ nie było instrukcji dotyczących przechowywania plików.

Pracownicy biurowi opracowali własne procedury pracy i konwencje nazewnictwa do zarządzania plikami elektronicznymi, co doprowadziło do niespójności i braku przejrzystości. Typowy katalog mógł wyglądać mniej więcej tak:

```
-rw-r--r-- 1 5,3K 1994-07-20 19:51 Pierwszy szkic EWS.org
-rw-r--r-- 1 3,7K 1994-07-20 08:51 Notatki EWS 17 stycznia 2023.org
-rw-r--r-- 1 2,5K 1994-05-04 16:47 Lista działań.org
-rw-r--r-- 1 24K 1994-07-20 20:04 FINAL paperback version.pdf
```

Taka konwencja nazewnictwa może być zrozumiała dla osoby, która ją opracowała, ale pamięć zanika, a te losowe nazwy plików mogą z czasem stać się mylące. Przekazywanie

Przesłanie komuś folderu takiego jak ten sprawi mu wiele kłopotu przy próbie ustalenia, który plik jest który. Dlatego każdy dobry projekt wymaga ustalenia konwencji nazewnictwa plików. Oto kilka praktycznych zasad dotyczących doskonałej konwencji nazewnictwa plików:

- Dodaj metadane do nazwy pliku, takie jak numer rozdziału, identyfikator eksperymentu lub data.
- Elementy metadanych należy oddzielać dozwolonymi znakami interpunkcyjnymi.
- Nazwy plików należy rozpoczynać od elementu, według którego chcesz posortować folder.
- Używaj tylko małych liter, aby uniknąć nieporozumień.

Niektóre konwencje nazewnictwa plików zawierają identyfikatory wersji, takie jak „wersja robocza”, „edytowana” i „ostateczna”. Jednak praktyka ta nie jest konieczna w przypadku korzystania z systemu kontroli wersji (sekcja 7.4), opisanego w rozdziale 7. Konwencja nazewnictwa plików Denote jest doskonałym przykładem najlepszej praktyki w zakresie nazewnictwa, opisaney w rozdziale 5.

Denote zawiera tryb pomocniczy, który formatuje zgodne nazwy plików w edytorze katalogów. Dzięki temu łatwo jest rozpoznać poszczególne elementy metadanych notatki. Nazwy plików nie tylko dostarczają metadanych dla samej notatki, ale służą również jako heurystyka ułatwiająca wyszukiwanie notatek na podstawie daty, podpisów, tytułu lub słowa kluczowego (rysunek y.2).

```
-rw-r--r-- 1 15K 2023-05-20 10:09 20060103T195904--the-bystander-effect-in-helping-behaviour__ola_prevosnet_psy12_psychology.org|
-rw-r--r-- 1 906 2022-08-27 13:08 20060305T173057--death-of-a-colleague__philosophy_prevosnet.org
-rw-r--r-- 1 8.8K 2022-09-05 14:54 20060403T10804--goffman-and-marcuse-on-identity__ola_prevosnet_sgy230_sociology.org
-rw-r--r-- 1 3.1K 2022-08-26 21:46 20060417T123727--existential-hot-air-balloon__philosophy_prevosnet.org
-rw-r--r-- 1 14K 2023-05-15 19:46 20060427T092727--postmodernism-and-language-games__philosophy_prevosnet.org
-rw-r--r-- 1 2.2K 2022-08-27 13:08 20060510T113929--reflections-on-life-after-death__prevosnet_religion.org
-rw-r--r-- 1 2.6K 2023-12-14 18:45 20060518T054215--immanuel-kant-the-mystic__esoterica_philosophy_prevosnet.org
```

Rysunek 9.2: Przykład plików Deeote, tj. Dived.

Ponieważ bufor Dired jest po prostu kolejnym buforem tekstowym, można edytować nazwy plików bezpośrednio za pomocą funkcji dired-toggle-read-only, przypisanej do skrótu C-x C-q. Funkcja ta pozwala na bezpośrednią edycję nazw plików w buforze, co jest wygodne, gdy trzeba edytować listę plików. Aby zapisać nowe nazwy plików, należy wpisać C-c C-c.

### Skróty klawiaturowe

Jeśli głowa pęka Ci od wszystkich skrótów klawiszowych, tabela y.1 zawiera listę funkcji opisanych w tym rozdziale. Możesz nacisnąć klawisz `h` w buforze `Dired`, aby wyświetlić wszystkie funkcje i powiązane skróty klawiszowe.

Tabela 9.1: Skróty klawiszowe.

Klawisz	Funkcja	Działanie
<code>a</code>	<code>dired-find-alternate-file</code>	Otwórz folder w tym samym buforze
<code>C</code>	<code>dired-do-copy</code>	Skopiuj plik
<code>j</code>	<code>dired-goto-file</code>	Przejdź do pliku powiązanego z aktywnym buforem
<code>g</code>	<code>revert-buffer</code>	Odśwież bufor <code>dired</code>
<code>m</code>	<code>dired-mark</code>	Zaznacz plik pod kursorem
<code>% m</code>	<code>dired-mark-files-regexp</code>	Zaznacz według wyrażenia regularnego
<code>o</code>	<code>dired-find-file-other-window</code>	Otwórz plik w innym oknie
<code>C-o</code>	<code>dired-display-file</code>	Wyświetl plik w innym oknie
<code>q</code>	<code>quit-window</code>	Zamknij bufor
<code>R</code>	<code>dired-do-rename</code>	Zmień nazwę (przenieś) plik
<code>t</code>	<code>dired-toggle-marks</code>	Odwróć zaznaczone pliki
<code>u</code>	<code>dired-unmark</code>	Usuń oznaczenie pliku pod kursorem
<code>U</code>	<code>dired-unmark-all-marks</code>	Usuń zaznaczenie wszystkich plików
<code>&amp;</code>	<code>dired-do-async-shell-command</code>	Otwórz plik w innym programie
<code>enter</code>	<code>dired-find-file</code>	Otwórz plik

### y.ç Wyświetlanie i manipulowanie obrazami

Pisanie zwykłym tekstem jest wspinałe, ale jak sugeruje oklepiane powiedzenie, „obraz jest wart tysiąca słów”. Widzieliście już, że `Org` może osadzać obrazy i eksportować je do żadanego formatu. Emacs posiada również wbudowane funkcje, które pomagają przeglądać i zarządzać zbiorem zdjęć.

Tryb obrazu jest dołączony do Emacs, ale do dodatkowych funkcji może być potrzebne oprogramowanie zewnętrzne. Emacs może wyświetlać obrazy bez oprogramowania zewnętrznego, ale nie może nimi manipulować. Pakiet oprogramowania `ImageMagick` zapewnia funkcje edycji i manipulacji obrazami.

Tryb obrazu może wyświetlać standardowe formaty obrazów bez dodatkowych ustawień. Plik obrazu można otworzyć bezpośrednio za pomocą funkcji `find-file` (C-x C-f) lub poprzez `Dired`. Można również otworzyć obraz powiązany z plikiem `Org` za pomocą C-c C-o (`org-open-at-point`), gdy kursor znajduje się na obrazie.

Emacs automatycznie skaluje obraz, aby idealnie pasował do okna. Do przeglądania zdjęć dostępnych jest szereg skrótów klawiaturowych. Klawisze `n` i `p` (następny i poprzedni) lub klawisze strzałek w lewo i w prawo umożliwiają przechodzenie między zdjęciami w bieżącym katalogu, tworząc pokaz slajdów. Tryb obrazu udostępnia również polecenia umożliwiające zmianę rozmiaru wyświetlania obrazów, między innymi:

- `s o`: Wyświetl obraz w oryginalnym rozmiarze.
- `s w`: dopasowuje bieżący obraz do wysokości i szerokości okna.
- `i +`: zwiększ rozmiar obrazu o 20%
- `i -`: zmniejsz rozmiar obrazu o 20%. Ponadto tryb

obrazu umożliwia manipulowanie obrazami:

- `i r`: Obróć obraz o 90 stopni w prawo.
- `i h`: Odbij obraz w poziomie.
- `i v`: Odbij obraz w pionie.
- `i c`: Przytnij obraz.
- `i x`: Cenzuruj prostokąt z obrazu, w stylu CIA.

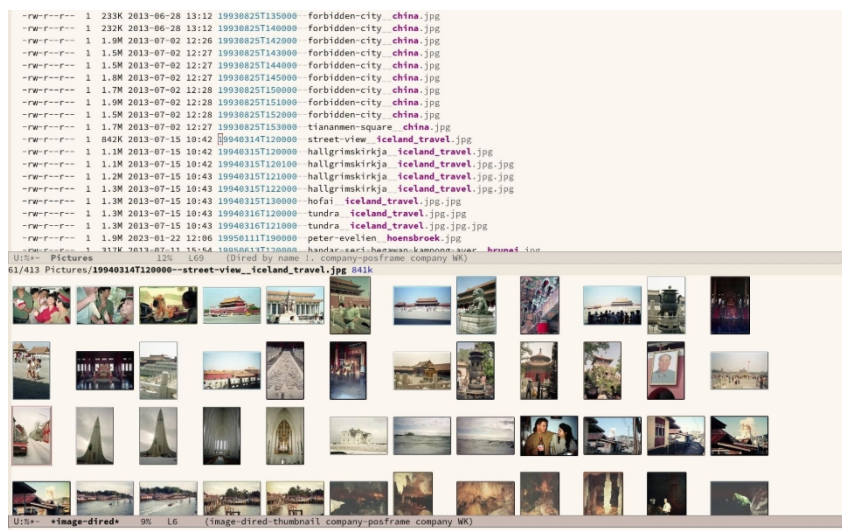
Polecenia przycinania i cenzurowania nakładają prostokąt na obraz. Użyj myszki, aby przesunąć i zmieniać rozmiar ramki. Wpisz `m`, aby przesunąć ramkę zamiast zmieniać jej rozmiar, a `s`, aby przekształcić ją w kwadrat. Gdy wynik będzie zadowalający, użyj `Ret`, aby przyciąć lub ocenzurować obraz.

obraz. Możesz wyjść z menu przycinania i wycinania za pomocą klawisza q bez zmiany pliku źródłowego. Pamiętaj, że te polecenia są dostępne tylko wtedy, gdy zainstalowany jest program ImageMagick.

Po zakończeniu przeglądania obrazów użyj klawisza q, aby zamknąć bufor obrazów, lub k, aby całkowicie go wyłączyć.

### *Pakiet Image-Dired*

Oglądanie pojedynczych obrazów jest pomocne, ale czy nie byłoby fajnie zobaczyć miniaturki? Pakiet Image-Dired udostępnia bufor miniatur, który umożliwia przeglądanie i zarządzanie obrazami z poziomu bufora Dired przy użyciu miniatur. Użyj polecenia image-dired i wybierz katalog, którego chcesz użyć (C-c w I). Emacs podzieli ekran i wyświetli do 1000 miniatur, które pozwolą Ci przeglądać swoją kolekcję (rysunek y.3). Emacs zapisuje miniatury w katalogu konfiguracyjnym, aby umożliwić późniejsze odzyskanie i szybsze ładowanie.



Rysunek 9.3: Wyświetlanie niektórych map podróży, np. Zmaes, z podglądem obrazu.

Aktywny obraz jest oznaczony migającą ramką wokół miniatury, a jego nazwa pliku jest wyświetlana w górnej części okna miniatur. Można przeglądać miniatury.

za pomocą klawiszy strzałek. Klawisze < i > przenoszą do początku lub końca kolekcji. Można usunąć miniaturę, ale nie sam plik, z bufora podglądu za pomocą C-d.

Aby wyświetlić obraz w innym oknie, naciśnij klawisz Ret na wybranej miniaturze. Za pomocą klawiszy spacji i Backspace (lub klawiszy strzałek w lewo i w prawo) można przechodzić między zaznaczonymi obrazami w kolekcji.

Ale po co ręcznie przeglądać album, skoro można to zlecić Emacsowi? Klawisz S uruchamia pokaz slajdów, w którym każdy obraz jest wyświetlany przez pięć sekund. Opóźnienie można dostosować za pomocą zmiennej image-dired-slideshow-delay lub sterować pokazem slajdów ręcznie. Główny obraz jest wyświetlany w trybie obrazu, więc mają zastosowanie wszystkie czynności opisane w poprzedniej sekcji. Jak zwykle, klawisz q powoduje zamknięcie okna obrazu lub miniatury, a k całkowicie je usuwa. Naciśnięcie C-<Ret> otwiera plik w zewnętrznej przeglądarce lub edytorze, takim jak GIMP. Możesz dostosować zmienną image-dired-external-viewer do nazwy swojego im-  
edytor wieku. Edytor zewnętrzny otwiera się z poziomu przeglądarki Image-Dired za pomocą kombinacji klawiszy C-<Ret>. Co może być mylące, w standardowym buforze Dired funkcja ta jest przypisana do kombinacji klawiszy C-t x. EWS używa kombinacji klawiszy C-<Ret> zarówno w Dired, jak i Image-Dired, aby otworzyć obraz w ulubionym edytorze zewnętrznym.

W buforze Dired zawierającym obrazy można wyświetlić miniatury, zaznaczając je za pomocą klawisza m, a następnie wygenerować miniatury za pomocą kombinacji klawiszy C-t C-t. Jeśli nie zaznaczysz żadnych plików, program wyświetli miniaturę obrazu znajdującego się pod kursorem. Ta sama kombinacja klawiszy służy również do usuwania miniatur.

#### y.4 Więcej informacji

Pakiet Dired, w tym Image-Dired, jest szczegółowo opisany w podręczniku Emacs. Wpisz C-h r g dired. Podręcznik Emacs zawiera również rozdziały dotyczące przeglądania obrazów, dostępne po wpisaniu C-h r g image. Niestety, Image-Dired nie posiada podręcznika.



## Rozdział 10

---

# Zostań mistrzem Emacsa

---

Teraz, gdy dotarłeś do końca tej książki, wiesz już wszystko, co potrzebne do wyszukiwania informacji, pisania i publikowania książek, prac dyplomowych lub czegokolwiek innego, czym chcesz się podzielić ze światem za pomocą Emacsa. Przechodząc przez rozdziały *Zmaes Wvities Studio* (EWS), opanowałeś najtrudniejszą część krzywej uczenia się. Jednak po pewnym czasie korzystania z Emacsa z pewnością zechcesz dostosować go do swoich potrzeb. Istnieje pięć etapów, które prowadzą do zostania mistrzem Emacsa, a pierwszy z nich właśnie teraz przechodzisz:

1. Zrozum podstawy
2. Zmodyfikuj konfigurację
3. Twórz funkcje
4. Zbuduj pakiet
5. Pomóż innym

Ten ostatni rozdział wprowadza Cię na ścieżkę do opanowania programu Emacs, dostarczając sugestii dotyczących pogłębienia wiedzy.

## 10.1 Zrozumienie podstaw

EWS obejmuje większość aspektów korzystania z Emacsa w zakresie związanym z pisanem prozy. Jednak ta książka nie jest kompleksowym podręcznikiem Emacsa, a raczej subiektywnym opisem realizacji szeregu zadań. Tekst tej książki jest subiektywny, ponieważ opisuje tylko jedną metodę osiągnięcia każdego celu.

Elastyczność programu Emacs oznacza, że zawsze istnieje więcej niż jeden sposób osiągnięcia tego samego rezultatu. Ta swoboda jest zarówno zaletą, jak i wadą, ponieważ może powodować niepotrzebne dyskusje i zamieszanie dotyczące sposobu wykonania zadania, zamiast po prostu je wykonać. Donald Knuth, geniusz komputerowy i twórca programu LaTeX, napisał kiedyś, że „przedwczesna optymalizacja jest źródłem wszelkiego zła”. Program EWS odzwierciedla tę zasadę, zapewniając rozwiązanie, które po prostu działa.

Jeśli po przeczytaniu tej książki chcesz dowiedzieć się więcej lub poznać różne sposoby wykonywania zadań, przeczytaj inne książki, takie jak *Leaving GNU Emacs* autorstwa Debry Cameron, *Mastery of Emacs* autorstwa Mickeya Petersena lub wbudowaną instrukcję Emacsa (Cameron, 2005; Petersen, 2022; Stallman, 2023).

Czytanie o Emacs to świetny sposób na naukę nowych rzeczy, ale czasem fajnie jest zobaczyć, jak ktoś używa tego systemu. YouTube to świetne miejsce, żeby znaleźć kanały z informacjami o Emacs, jak na przykład polecane filmy Prot Stavrou i kanał Davida Wilsona *System Crafters*. Opublikowałem też serię filmów pokazujących możliwości EWS.

## 10.2 Modyfikacja konfiguracji

*Emacs Writings Studio* oferuje rozwiązanie bez kodowania, zapewniając dokładnie przetestowaną konfigurację i wykorzystując interfejs użytkownika do konfiguracji Emacsa. Nadejdzie moment, kiedy zechcesz zmienić niektóre domyślne ustawienia bezpośrednio w pliku init lub dodać dodatkowe pakiety, co oznacza, że będziesz musiał nauczyć się trochę języka Emacs Lisp.

Dlatego więc miałbyś chcieć konfigurować Emacsa poprzez pisanie kodu zamiast korzystać z nowoczesnych

funkcji graficznych oferowanych przez inne programy? Komercyjne oprogramowanie udostępnia graficzne menu, które pozwalają zdefiniować sposób jego działania. Na przykład na rysunku 10.1 można zaznaczyć pole, wybrać pozycję z listy lub wprowadzić wartość w polu tekstowym, aby skonfigurować program zgodnie z własnymi preferencjami.

The image shows a graphical user interface for Emacs preferences. It consists of three panels. The first panel, titled 'Inhibit startup message', contains a checked checkbox. The second panel, titled 'Initial scratch message', contains a text input field with the text 'Hello world'. The third panel, titled 'Cursor Type', contains a dropdown menu with 'Bar' selected and a blue arrow indicating it can be expanded.

*Rysunek 10.1: Typowy graficzny interfejs użytkownika.*

Poniższy kod odpowiada rysunkowi 10.1. Ten przykład określa, czy Emacs wyświetla komunikat startowy przy pierwszym uruchomieniu. Drugi wiersz ustawia początkowy komunikat tymczasowy. W tym przypadku parametrem jest ciąg liter umieszczony w cudzysłowie. Ostatnia linia ustawia typ kursora na pasek. Ta zmienna ma inne predefiniowane opcje, takie jak „bar” lub „hollow”. Używa pojedynczego cudzysłowu przed tekstem, aby zapobiec pomyleniu tej opcji z zmienną przez Emacs. Porównaj te linie z obrazkiem, aby przeprowadzić inżynierię odwrotną kodu Elisp.

```
(setq inhibit-startup-message t
 initial-scratch-message "Hello world"
 cursor-type 'bar)
```

Zwróć uwagę, że całe wyrażenie jest ujęte w nawiasy, co jest typowe dla programowania w języku Lisp. Nawiasy te mogą dezorientować początkujących użytkowników podczas zagrzeżdżania kodu na wielu poziomach. Gdy przyzwyczaisz się do składni, nawiasy staną się pomocnymi znacznikami, które pomogą Ci zrozumieć logikę kodu.

Chociaż na pierwszy rzut oka metoda oparta na kodzie wydaje się bardziej skomplikowana niż zaznaczanie i wpisywanie w polach oraz wybieranie z listy rozwijanej, jest ona znacznie bardziej wydajna niż interfejs graficzny. Gdy nauczysz się pisać proste skrypty w języku Emacs Lisp, zdasz sobie sprawę, że Emacs jest w rzeczywistości najbardziej przyjaznym dla użytkownika systemem, ponieważ daje Ci ogromne możliwości.

Twój komputer. Korzystanie z Emacs Lisp jest kwintesencją łatwości obsługi. To Ty decydujesz, jak zachowuje się komputer, a nie firma produkująca oprogramowanie kontroluje Twoje zachowanie. Jednak ta ogromna moc wiąże się z wielką odpowiedzialnością i koniecznością nauki.

Najłatwiejszym sposobem nauki języka Lisp jest zmiana pliku inicjującego poprzez skopiowanie wzorców z konfiguracji EWS. Dodatek wyjaśnia konfigurację i zawiera wskazówki dotyczące jej modyfikacji. Wielu użytkowników Emacsa chętnie dzieli się swoimi konfiguracjami, więc nie wahaj się skopiować dowolnych fragmentów ich kodu. Emacs jest wolnym oprogramowaniem w pełnym tego słowa znaczeniu, więc skorzystaj z tej możliwości i ucz się od innych użytkowników.

Ostatnia wskazówka: kiedy kopiujesz kod od innych użytkowników Emacsa, nie dodawaj niczego do swojej konfiguracji, dopóki nie zrozumiesz dokładnie, co on robi. Przeczytaj dokumentację i wypróbuj kilka alternatywnych wersji, aby w pełni zrozumieć kod.

### *Modyfikowanie sekwencji klawiszy*

Emacs posiada szereg predefiniowanych skrótów klawiaturowych dla swoich podstawowych funkcji i wbudowanych pakietów. Większość pakietów zewnętrznych nie definiuje skrótów klawiaturowych, aby zapobiec konfliktom z konfiguracją użytkownika. Konfiguracja EWS przypisuje skróty do najczęściej używanych funkcji za pomocą prefiksu C-c w. Jak wyjaśniono w załączniku, można łatwo modyfikować skróty klawiaturowe wymienione w pliku init.

Zachowanie klawiatury można zmieniać na trzech poziomach: klawiatury programowalne, system operacyjny/menedżer okien i Emacs.

Niektóre klawiatury z wyższej półki są programowalne, co pozwala na zdefiniowanie funkcji każdego klawisza. Na przykład można przypisać prawy klawisz Control jako klawisz Hyperkey. Na drugim poziomie system operacyjny interpretuje dane wejściowe z klawiatury. Na przykład w systemie Windows kombinacja klawiszy S-E (Windows i E) otwiera eksplorator plików. Można usunąć to powiązanie, aby udostępnić je w Emacs. Każdy system operacyjny ma własne metody zmiany map klawiatury (keymaps). Niektórzy doświadczeni użytkownicy Emacs zmieniają przypisanie klawisza Caps Lock, aby działał on jako klawisz Control, co ułatwia korzystanie z klawiatury i zapobiega powstawaniu urazów spowodowanych powtarzającym się obciążeniem, znanych jako „Emacs Pinky”.

Na koniec, w samym Emacsie można definiować sekwencje klawiszy. Poniższy przykład przypisuje klawisz f5 do przełączania trybu spacji. Ten tryb pomocniczy wskazuje spacje w bieżącym buforze za pomocą znaków. Znaki #’ przed nazwą funkcji są wymagane technicznie, aby poinformować Emacs, aby nie oceniał tej funkcji, a jedynie zapisał jej wartość. Jeśli chcesz usunąć naciśnięcie klawisza, użyj nil jako nazwy funkcji.

```
(keymap-global-set "<f5>" #'whitespace-mode)
```

W poprzednim przykładzie użyto globalnej mapy klawiszy, co oznacza, że skrót jest dostępny we wszystkich trybach. Można również zdefiniować skrót dla konkretnego trybu, który będzie dostępny tylko wtedy, gdy ten tryb jest aktywny. W poniższym przykładzie ustawiono ten sam skrót, ale ma on zastosowanie tylko wtedy, gdy aktywny jest tryb Org, więc klawisz f5 może wykonywać inne czynności w innych trybach.

```
(keymap-set org-mode-map "<f5>" #'whitespace-mode)
```

Sekretem dobrych skrótów klawiaturowych jest kodowanie informacji semantycznej, dlatego prawie wszystkie skróty specyficzne dla EWS zaczynają się od C-c w, gdzie w oznacza „pisanie”.

Takie podejście nie zawsze się sprawdza, ponieważ szybko zabraknie najbardziej oczywistych liter.

Niektórzy nie lubią domyślnych ustawień klawiatury Emacs, bo trzeba często używać klawiszy modyfikujących. Twierdzą, że ciągłe używanie tych klawiszy powoduje kontuzje, czyli straszne „różowe palce Emacs”. W ekosystemie Emacs jest kilka pakietów, jak Evil Mode i God Mode, które zmieniają domyślne skróty klawiszowe na inny model.

### *Dodatkowe pakiety alternatywne*

EWS zawiera wiele pakietów, które rozszerzają możliwości Emacs. Istnieją jednak alternatywne rozwiązania i rozszerzenia, które nie znalazły się w EWS. Jednym z celów tej konfiguracji było pozostawianie „tak blisko, jak to tylko możliwe” standardowego środowiska Emacs, aby nowi użytkownicy mogli nauczyć się obsługi Emacs, a nie konkretnej konfiguracji. W tej sekcji wymieniono kilka dodatkowych lub alternatywnych pakietów, na wypadek gdybyś chciał wypróbować coś innego.

EWS rozszerza mechanizm uzupełniania minibuffera o Vertico i powiązane pakiety. Popularnymi alternatywami dla tego ekosystemu są frameworki uzupełniania Helm lub Ivy. Oba pakiety oferują zróżnicowane i kompleksowe środowisko.

Embark to kolejny pakiet, który może wzbogacić doświadczenia związane z korzystaniem z Emacsa. Embark zapewnia funkcjonalność podobną do menu kontekstowego wyświetlanego po kliknięciu prawym przyciskiem myszy w oprogramowaniu graficznym, dostępną za pomocą skrótu klawiaturowego.

Wbudowana przeglądarka plików PDF może działać wolno i nie posiada zaawansowanych funkcji. Pakiet Pdf-Tools autorstwa Vedanga Manerikara zapewnia rozszerzoną funkcjonalność, ale wymaga kompilacji kodu źródłowego, aby w pełni zintegrować go z Emacsem.

Pakiet eBib autorstwa Joosta Kremersa jest narzędziem do zarządzania plikami BibTeX, które rozszerza funkcjonalność Citar. Pakiet ten nie wykorzystuje menu uzupełniania. Zamiast tego posiada bardziej tradycyjny interfejs aplikacji, idealny do tworzenia lub edytowania plików bibliograficznych.

Pakiet Org-Ref autorstwa Johna Kitchina jest częścią jego zestawu startowego Scimax Emacs dla naukowców i inżynierów. Pakiet ten zawiera zaawansowane narzędzia do zarządzania bibliografiami i odsyłaczami w plikach Org i warto się nim zainteresować.

Oto krótki przegląd niektórych alternatywnych pakietów, z których możesz korzystać. Emacs oferuje szeroki wachlarz pakietów do wykonywania różnorodnych zadań. Polecenie `list-packages` zapewnia przydatny interfejs do listy wszystkich dostępnych pakietów. Możesz przeglądać ten interfejs, aby znaleźć interesujące oprogramowanie lub zaktualizować swój obecny system.

### *Czy Zmaes jest programem zwiększającym produktywność?*

Konfiguracja Emacsa może być trudnym zadaniem, które zajmuje dużo czasu i zmniejsza produktywność ze względu na swoją złożoność. Swoboda, jaką daje Emacs, oznacza, że każdy ma swoje osobiste preferencje. Twórcy Emacsa nie są w stanie zaspokoić wszystkich osobistych preferencji, dlatego udostępniają szkielet systemu, który należy dostosować do własnego stylu pracy. Emacs Lisp jest jak pudełko klocków Lego, z których można tworzyć zabawki. Praca nad konfiguracją Emacsa sprawia wiele radości, ale jeszcze większą przyjemność daje korzystanie z niego.

Ciągle dostosowywanie konfiguracji może być kuszące, ale może to negatywnie wpłynąć na produktywność. Możliwość stworzenia spersonalizowanego systemu Emacs jest wielką pokusą, która może prowadzić do pozornej produktywności, będącej jedną z trzech form prokrastynacji:

1. *Nihilistie pvoevastieatioe*: Oglądanie telewizji i granie w gry komputerowe.
2. *Soptistieated pvoevastieatioe*: fałszywa produktywność, np. hakowanie Emacs, zmiana narzędzi zwiększających produktywność, robienie notatek pod kątem ilości, a nie jakości.
3. *Pvduetive pvoevastieatioe*: Marzenia na jawie.

Majstrowanie przy konfiguracji Emacs nie jest tak złe jak nihilistyczne zwleknięcie, ale może stać się formą fałszywej produktywności. Zyski w zakresie produktywności wynikające z dostosowania Emacs do wyeliminowania kilku naciśnięć klawiszy z przepływu pracy mogą zająć więcej czasu niż zaoszczędzisz dzięki nowemu przepływowi pracy przez resztę swojego życia. Pisanie to coś więcej niż tylko liczba słów, które można wpisać do bufora. Pisanie jest w równym stopniu aktem kontemplacyjnym, jak i kwestią wydajności klawiatury.

## 10.4 Tworzenie funkcji

Żałujmy, że komputer jest, według słów Steve'a Jobsa, „rowerem dla umysłu”. W takim razie Emacs jest Hot Rodem dla umysłu, zapewniającym najwyższą swobodę pracy z komputerem.

Trzecim krokiem do całkowitej swobody w korzystaniu z komputera jest pisanie niestandardowych funkcji do wykonywania zadań specyficznych dla Twojego przepływu pracy. Większość kodu Emacs istnieje, ponieważ użytkownik chce osiągnąć coś, czego nie potrafi ani Emacs, ani żaden istniejący pakiet. Sposób działania Emacs polega na tym, że to Ty opracowujesz rozwiązanie problemu.

Kod, który uruchamia Emacs, jest w większości napisany jako zbiór funkcji Lisp. Najprościej mówiąc, funkcja to coś, co zamienia dane wejściowe na dane wyjściowe. Na przykład, za każdym razem, gdy naciskasz klawisz na klawiaturze w Emacs, funkcja `self-insert` zamienia to naciśnięcie na znak w otwartym buforze.

Poniższy kod przedstawia architekturę prostej funkcji Emacs. Po uruchomieniu tego polecenia Emacs wyświetli powitanie w obszarze echo, pobierając Twoje imię z systemu.

```
(defun ews-greet-user ()
 „Wyświetl wiadomość powitalną w minibufferze”. (interactive)
 (message „Witaj %s, witaj w Emacsie.” user-full-name))
```

Teraz przeanalizujemy, jak działa ta funkcja:

- Funkcja rozpoczyna się od makra `defun`, które służy do zdefiniowania nowej funkcji. W tym przypadku nazwa funkcji to `ews-greet-user`. Nazwa funkcji zaczyna się od `ews`, zgodnie z nieformalnymi konwencjami nazewniczymi, które zalecają rozpoczynanie nazw funkcji od nazwy grupy lub pakietu. Nawiasy na końcu są puste, co oznacza, że funkcja ta nie ma żadnych danych wejściowych.
- Drugi wiersz opisuje działanie funkcji. Każda funkcja w systemie posiada ciąg dokumentacyjny, który można odczytać za pomocą `C-h f`. W przypadku funkcji interaktywnych pierwszy wiersz tych plików pomocy jest również wyświetlany w menu uzupełniania w minibufferze podczas używania `M-x`.
- Wiersz (interaktywny) konwertuje funkcję na polecenie. Oznacza to, że można ją wywołać za pomocą `M-x ews-greet-user` lub przypisać do sekwencji klawiszy.
- Ostatnia linia wyświetla komunikat w obszarze echo za pomocą funkcji `message`. W tym przypadku `%s` jest zastępowane zawartością zmiennej `user-full-name`.

Możesz zapisać tę funkcję w buforze roboczym, aby ją wypróbować. Następnie należy ją aktywować, umieszczając kursor za ostatnim nawiasem zamykającym i używając `C-x C-e` (`eval-last-sexp`). Ta czynność rejestruje funkcję w pamięci, po czym można ją wywołać za pomocą `M-x`.



Możesz dodać tę funkcję do pliku `init`, aby była dostępna w każdej sesji Emacs'a. Wątpię jednak w przydatność tego przykładu.

Sposób wykorzystania funkcji ogranicza jedynie Twoja wyobraźnia. EWS zawiera zestaw specjalnie dostosowanych funkcji w pliku `ews.el`, które rozszerzają niektóre podstawowe funkcje. Aby wyświetlić listę wszystkich dostępnych poleceń EWS, użyj `M-x ^ews`. Ta sekwencja wyświetla wszystkie polecenia w menu uzupełniania minibuffera, które zaczynają się od (^) „ews”.

## 10.4 Zbuduj pakiet

Kolejnym poziomem zaawansowania w Emacsie jest udostępnianie światu własnych funkcji za pomocą pakietu. Większość pakietów Emacs'a powstaje w wyniku próby rozwiązania przez użytkownika problemu w jego pracy, a następnie decyzji o udostępnieniu rozwiązania światu.

Na przykład mój pakiet Citar-Denote powstał z chęci używania Emacs'a do tworzenia notatek bibliograficznych. Nie miałem doświadczenia w kodowaniu w Emacsie, ale mimo to postanowiłem spróbować. Najpierw opublikowałem podstawową wersję Citar-Denote. Nie potrafiłem jednak wymyślić, jak wykonać konkretne zadania. Gdy tylko udostępniłem kod, inni szybko zaoferowali pomoc i wkrótce pakiet Citar-Denote stał się w pełni funkcjonalnym narzędziem do zarządzania notatkami bibliograficznymi.

## 10.5 Pomóż innym

Ostatnim krokiem do zostania mistrzem Emacs'a jest pomaganie innym w ich podróży. Możesz pomagać innym użytkownikom poprzez różne fora internetowe, takie jak Reddit, które mają aktywne społeczności Emacs'a.

Ta książka powstała jako strona internetowa, na której dzieliłem się swoimi doświadczeniami z Emacs'em jako autor.

W miarę jak strona zyskiwała popularność, przekształciłem ją w książkę, którą właśnie czytasz.

Projekt EWS jest moim sposobem na odwdzięczenie się społeczności Emacs'a, a także pomaga mi lepiej zrozumieć, jak działa to oprogramowanie. Niektórzy cynicy sugerują, że „ci, którzy nie potrafią

, uczyć”. Jednak nauczanie dowolnego przedmiotu jest najlepszym sposobem na usystematyzowanie swojej wiedzy i doskonalenie się w danej dziedzinie.

Twoim zadaniem jako ninja Emacsa jest więc pomaganie innym, którzy są mniej zaawansowani w swojej podróży. Opowiedz innym autorom o swobodzie, jaką daje Ci Emacs jako badaczowi, autorowi, nauczycielowi lub kimkolwiek innym jesteś.

---

# Dodatek

---

W niniejszym dodatku przedstawiono i wyjaśniono konfigurację *Zmaes Wvities Studio* (EWS). Konfiguracja została opatrzona komentarzami wyjaśniającymi logikę kodu i zawiera kilka opcji ulepszeń lub dodatkowych funkcji. Konfiguracja ta opiera się na następujących zasadach:

- Wykorzystaj funkcjonalność najnowszej wersji GNU Emacs
- Minimalistyczna konfiguracja
- Korzystanie ze standardowych skrótów klawiaturowych
- Skoncentrowany na trybie Org
- Brak konfiguracji do pisania kodu

Ta konfiguracja stanowi punkt wyjścia dla Twojej przygody z Emacsem. Możesz dowolnie modyfikować dowolną część pliku inicjalizacyjnego EWS i badać jego działanie. Najnowszą wersję tej konfiguracji znajdziesz na GitHubie:

<https://github.com/pprevos/emacs-writing-studio>

To repozytorium nie jest aktywnie utrzymywane. Dla początkujących użytkowników Emacs konfiguracja ta stanowi punkt wyjścia do opracowania własnej konfiguracji. Bardziej zaawansowani użytkownicy mogą czerpać z niej pomysły. Kod będzie aktualizowany tylko w przypadku wystąpienia błędów spowodowanych przyszłymi zmianami w pakietach lub samym Emacsie.

## Używanie EWS

Pierwsza część konfiguracji określa podstawowe zasady konfiguracji EWS, takie jak zarządzanie pakietami, wygląd i działanie interfejsu użytkownika, system uzupełniania minibuffera oraz podstawowe ustawienia umożliwiające pisanie dla ludzi.

### *Basie eoefiguvatioe*

Pierwsza część konfiguracji sprawdza, czy działa najnowsza wersja Emacs. EWS wykorzystuje niektóre z najnowszych funkcji, więc musisz zainstalować tę wersję. Zwróć uwagę, że wyrażenie lub wyższe (`< emacs-major-version 29`) jest w notacji polskiej. W języku Lisp operator umieszcza się przed operandami, w przeciwieństwie do bardziej powszechnej notacji infiksowej, w której operator umieszcza się między operandami, np. `emacs-major-version < 29`.

```
(when (< emacs-major-version 29)
 (error "Emacs Writing Studio wymaga wersji 29 lub nowszej"))
```

### *Zmaes packages*

Użytkownicy Emacs opracowali i udostępnili społeczności tysiące pakietów. Pakiety te są napisane w języku Emacs Lisp i rozszerzają możliwości programu. Ta część konfiguracji określa podstawowe elementy niezbędne do załadowania i zainstalowania pakietów z archiwum MELPA ([melpa.org](http://melpa.org)).

```
(use-package package
 :config
```

```
(add-to-list 'package-archives
 '("melpa" . "https://melpa.org/packages/"))
(package-initialize))
```

Ta konfiguracja implementuje Use-Package Johna Wiegleya. Pakiet ten upraszcza instalację i konfigurację pakietów dzięki znormalizowanej i łatwej do odczytania metodzie. Twórcy oprogramowania nazywają takie narzędzie „cukrem syntaktycznym”, czyli składnią zaprojektowaną w celu ułatwienia czytania lub pisania kodu, dzięki czemu język staje się „słodszy” dla ludzi (Landin, 1954).

System Use-Package składa się z zestawu instrukcji umieszczonych w nawiasach, które stanowią makro. W najprostszej postaci wygląda to mniej więcej tak: (use-package <nazwa\_pakietu>). Kod może również zawierać jedną lub więcej sekcji służących do ustawiania różnych opcji. Sekcja :custom poniżej ustawia trzy zmienne. Zmienne te uruchamiają trzy protokoły. Każdy brakujący pakiet jest automatycznie instalowany ze źródła internetowego, a kod źródłowy jest kompilowany w celu przyspieszenia działania programu Emacs. Po trzecie, ostrzeżenia kompilacyjne są ograniczone do minimum, aby nie odstraszać początkujących użytkowników plikami dziennika.

```
(use-package use-package
 :custom
 (use-package-always-ensure t)
 (package-native-compile t) (warning-
 minimum-level :emergency))
```

Aby zapoznać się ze szczegółowymi informacjami na temat makra Use-Package, przeczytaj instrukcję, wybierając C-h R use-package.

*Zmaes Wvitiag Studio eoeveeieeee fueetioes*

EWS oferuje również szereg dostosowanych do potrzeb funkcji ułatwiających różne aspekty procesu pisania. Pamiętaj, aby pobrać ten plik z repozytorium EWS.

```
(load-file (concat (file-name-as-directory user-emacs-directory) „ews.el”))
```

Funkcja `ews-missing-executables` sprawdza, czy w systemie dostępne jest oprogramowanie zewnętrzne. Emacs wyświetla komunikat w minibufferze, jeśli brakuje któregośkolwiek z zalecanych narzędzi. Aby przejrzeć wynik, można przejść do bufora komunikatów za pomocą skrótu `C-h e`. Emacs będzie działał normalnie, nawet jeśli oprogramowanie to jest niedostępne, ale niektóre funkcje mogą nie działać.

Dane wejściowe dla tej funkcji to lista, czyli ciąg znaków w nawiasach, zaczynający się od symbolu tyldy: `„(„this” „is” „a” „list”)`. Tylda zapobiega pomyleniu listy danych z funkcją przez program Emacs. W tej funkcji niektóre listy zawierają również inne listy.

Ta funkcja sprawdza, czy wszystkie te pakiety są dostępne w systemie. W przypadku oprogramowania w liście zagnieżdżonej, np. `(„convert” „gm”)`, wystarczy, że tylko jedno z nich jest dostępne, ponieważ programy te są alternatywnymi rozwiązaniami o tej samej funkcjonalności.

```
(ews-missing-executables '(„gs”
„mutool”)
„pdftotext”
„soffice”
„zip” „ddjvu”
„curl”
("mpg321" "ogg123" "mplayer" "mpv" "vlc")
("grep" "ripgrep")
(„convert” „gm”)
„dvipng” „latex”
„hunspell” „git”))
```

### *Wygląd i wrażenia*

EWS stosuje minimalistyczną estetykę, aby zapewnić środowisko wolne od rozpraszających elementów. Te cztery linijki kodu wyłączają ekran powitalny, pasek narzędzi, pasek menu i pasek przewijania.

Pasek menu może być przydatny dla początkujących użytkowników i nadal można uzyskać do niego dostęp za pomocą klawisza F10. Jeśli chcesz zachować pasek narzędzi, menu i/lub paski przewijania, usuń odpowiednie wiersze, zmień wartość -1 na 1 lub dodaj dwa średniki na początku odpowiednich wierszy, aby przekształcić je w komentarze.

```
(setq inhibit-splash-screen t)
(tool-bar-mode -1)
(menu-bar-mode -1)
(scroll-bar-mode -1)
```

Vanilla Emacs ma nieco paternalistyczny zwyczaj wymagania odpowiedzi „y” lub „n”, podczas gdy w niektórych przypadkach wymaga wpisania „yes” lub „no”, ze względu na postrzegane wyższe ryzyko wpisania błędnej odpowiedzi. Funkcja `setq` ustawia zmienną `use-short-answers` na `t`. Jeśli chcesz zachować to zachowanie, zmień `t` na `nil`.

W Emacs Lisp `t` oznacza `TRUE`, a `nil` jest równoważne `FALSE`. Dokumentacja Emacs’a często wspomina o ustawieniu wartości na „non-nil”, co jest podwójnym zaprzeczeniem sugerującym ustawienie zmiennej na `true`.

```
(setq-default use-short-answers t)
```

Bufor Scratch jest domyślnie ustawiony na tryb Emacs Lisp. Nie jest to zbyt przydatne dla autorów, więc zmieniamy to na tryb Org i dodajemy nową domyślną wiadomość.

```
(setq initial-major-mode 'org-mode
 initial-scratch-message (concat "#+title: Emacs Writing Studio\n"
 „#+subtitle: Bufor roboczy\n\n”
 „Tekst w tym buforze nie jest zapisywany” „po
 zamknięciu Emacs’a\n\n”))
```

Kolejne dwie sekcje kodu jeszcze bardziej ulepszają interfejs Emacs’a dzięki dwóm pakietom autorstwa guru Emacs’a, Protesilaosa Stavrou.

Pakiet przestrzennego wypełnienia tworzy przestrzeń wokół okien, zapobiegając zagęszczeniu tekstu na ekranie. Sekcja `:init` zawiera kod, który Emacs ocenia podczas ładowania pakietu

pakietu. W tym przypadku włącza tryb Spacious Padding. Sekcja :custom ustawia również większy odstęp między wierszami. Możesz przeczytać instrukcję obsługi tego trybu, wpisując C-h R spacious.

Ta konfiguracja modyfikuje również zmienną odstępów między wierszami, aby utworzyć pewną przestrzeń między wierszami logicznymi. Ta zmienna nie jest częścią pakietu Spacious Padding.

```
(use-package spacious-padding
 :custom
 (line-spacing 3)
 (spacious-padding-mode 1))
```

Następny pakiet ustawia motyw Emacs. Motyw to zestaw konfiguracji czcionek i kolorów. Motywy są dostępne w dwóch wersjach: z jasnym lub ciemnym tłem.

Pakiet motywów Modus jest wysoce konfigurowalny. Deklaracja Use-Package zawiera trzy sekcje. Sekcja :custom dostosowuje zmienne używane w pakiecie. W tym przypadku instruujemy pakiet, aby używał czcionek pochyłych i pogrubionych dla podkreślenia oraz zezwalał na czcionki o stałym i zmiennym rozstawie. Kod nieznacznie zwiększa również rozmiar nagłówków. Można przełączać się między ciemną i jasną wersją tego motywu, a ostatnia zmienna określa, między którymi wersjami ma się przełączać. EWS używa przyciemnionej wersji motywów, którą można modyfikować.

Sekcja :custom makra ustawia pewne zmienne w celu zdefiniowania czcionek. Sekcja ta określa również, które motywy są przełączane podczas zmiany między jasnym a ciemnym motywem za pomocą C-c w t t. Domyślnie są to wersje barwione. Jeśli chcesz, aby Twoja konfiguracja domyślnie korzystała z wersji o wysokim kontraście lub jednej z dwóch wersji bezpiecznych dla osób z daltonizmem, dostosuj zmienną modus-themes-to-toggle. Aby zobaczyć możliwe opcje motywów Modus, skorzystaj z pliku pomocy: C-h v modus-themes-collection.

W poniższej sekcji przypisano niektóre klawisze do poleceń umożliwiających przełączanie między trybem ciemnym i jasnym lub wybór dowolnego dostępnego motywu modus. Wszystkie niestandardowe skróty klawiszowe EWS zaczynają się od klawisza prefiksowego C-c w oraz klawisza prefiksowego C-c w t dla funkcji związanych z motywami.



Możesz oczywiście zmienić je zgodnie z własnymi preferencjami. Szczegółowe informacje znajdziesz w instrukcji obsługi pakietu Modus Themes, wpisując C-h R modus.

Polecenie `consult-theme` wywołuje pakiet `consult`, który pomaga wybrać jeden z zainstalowanych motywów. Aby ustawić domyślny motyw, uruchom polecenie `customize-themes` i wybierz preferowaną wersję. Kliknij przycisk, aby zapisać wybrany domyślny motyw w pliku `custom.el`. Dodatkowo ten kod instaluje również pakiet `Ef-Themes Port`, który jest wspaniałą kolekcją jasnych i ciemnych motywów.

```
(use-package modus-themes
 :custom
 (modus-themes-italic-constructs t)
 (modus-themes-bold-constructs t)
 (modus-themes-mixed-fonts t)
 (modus-themes-to-toggle '(modus-operandi-tinted
 modus-vivendi-tinted))
 :bind
 („C-c w t t” . modus-themes-toggle) („C-
 c w t m” . modus-themes-select) („C-c w
 t s” . consult-theme)))
```

(użyj pakietu `ef-themes`)

Następna sekcja podłącza tryb zmiennej szerokości znaków do dowolnego bufora Org. Oznacza to, że napisany tekst jest wyświetlany ze zmienną szerokością znaków, podczas gdy metadane, kod i inne elementy są wyświetlane ze stałą szerokością znaków. Hook to konstrukcja w Emacsie, która łączy ze sobą różne tryby. W tym przypadku tekst ze zmienną szerokością znaków jest włączony dla wszystkich buforów w trybie tekstowym.

```
(use-package mixed-pitch
 :hook
 (org-mode . mixed-pitch-mode))
```

Ten ostatni fragment kodu w sekcji wyglądu i działania zmienia sposób, w jaki Emacs automatycznie dzieli okna, preferując podziały pionowe nad poziomymi, aby poprawić czytelność. To

Sekcja ta instaluje również pakiet *Balanced Windows*, który automatycznie zarządza rozmiarami okien. Na przykład, gdy masz otwarte trzy okna i zamykasz jedno z nich, pozostałe okna zajmują po połowie ekranu.

```
(setq split-width-threshold 120
 split-height-threshold nil)
```

```
(use-package balanced-windows
 :config
 (balanced-windows-mode))
```

### *Mieibuffev eompletioe*

EWS wykorzystuje w swojej standardowej konfiguracji pakiet *Vertico-Orderless-Marginalia* zawierający pakiety uzupełniania minibuffera. Rozdział 4 wyjaśnia, jak korzystać z uzupełniania minibuffera.

```
(use-package vertico
 :init (vertico-
 mode)
 :custom
 (vertico-sort-function 'vertico-sort-history-alpha))
```

```
(use-package savehist
 :init
 (tryb savehist))
```

```
(użyj pakietu orderless
 :custom
 (style uzupełniania '(bez porządku
 podstawowe)) (domyślne kategorie
 uzupełniania nil) (nadpisanie kategorii
 uzupełniania
 ((plik (style częściowe uzupełnianie)))))
```

```
(użyj pakietu marginalia
 :init
 (tryb marginalia))
```

### *Klawiatura slovteuts meeu*

Pakiet Which-Key poprawia wykrywalność skrótów klawiaturowych dzięki wyskakującemu okienku w minibufferze.

Ze względu na konwencje nazewnictwa w Emacs, większość funkcji zaczyna się od nazwy pakietu, więc niektóre mogą być długie. Problem polega na tym, że najciekawsza część nazwy funkcji znajduje się na końcu ciągu znaków, więc nie chcemy, aby była ona ukryta. Ta konfiguracja nieco poszerza kolumny, aby zapobiec skracaniu nazw funkcji. Konfiguracja ta instruuje również Which-Key, aby sortować listę według nazwy funkcji, a nie według klawisza.

```
(użyj-pakietu który-klucz
:config
(tryb klucza)
:custom
(which-key-max-description-length 40)
(which-key-lighter nil)
(kolejność sortowania which-key 'kolejność opisów which-key)
:init
(który-klawisz-dodaj-zastąpienia-oparte-
 na-klawiszach „C-c w” „Emacs
 Writing Studio” „C-c w b”
 „Bibliograficzne”
 „C-c w d” „Oznacz”
 „C-c w m” „Multimedia”
 „C-c w s” „Spelling and Grammar” „C-
 c w t” „Themes”
 „C-c w x” „Przeglądaj”))
```

### *Korzystanie z myszy*

Ten fragment kodu umożliwia wyświetlenie kontekstowego menu podręcznego po kliknięciu prawym przyciskiem myszy.

```
(when (display-graphic-p)
(context-menu-mode))
```

### *Ulepszona funkcjonalność pomocy*

Emacs jest reklamowany jako „samodokumentujący się edytor tekstu”. Chociaż nie jest to do końca prawdą (gdyby tylko kod komputerowy mógł sam się dokumentować), każdy aspekt Emacsa jest udokumentowany w kodzie źródłowym.

Emacs posiada dwa poziomy pomocy. Po pierwsze, istnieją podręczniki dotyczące samego Emacsa i niektórych pakietów. Ponadto każde polecenie i funkcja zawiera dokumentację. Pakiet Helpful autorstwa Wilfreda Hughesa dodaje informacje kontekstowe do wbudowanej pomocy Emacsa. Na przykład, gdy użytkownik prosi o dokumentację dotyczącą zmiennej, plik pomocy zawiera link do ekranu jej dostosowywania lub kodu źródłowego.

```
(use-package helpful
 :bind
 („C-h f” . pomocna-funkcja) („C-
 h x” . pomocna-komenda) („C-h
 k” . pomocny-klawisz)
 ("C-h v" . helpful-variable)))
```

### *Tryby tekstowe Coefiguve*

Emacs jest przeznaczony głównie do tworzenia kodu komputerowego, więc wymaga pewnych modyfikacji, aby umożliwić pisanie tekstu przeznaczonego dla ludzi. Konfiguracja najpierw zapewnia, że Emacs nie próbuje zainstalować Text-Mode jako pakietu, ponieważ jest on wbudowany.

Po drugie, podłączamy tryb Visual Line Mode do Text Mode. Tryb Visual Line Mode zawija długie linie do najbliższego słowa, aby zmieściły się w bieżącym oknie, tak jak ma to miejsce w programach do edycji tekstu.

Domyślnie Emacs nie zastępuje tekstu po zaznaczeniu fragmentu i rozpoczęciu pisania, co jest nietypowym zachowaniem podczas pisania prozy. Sekcja `:init` włącza bardziej powszechne domyślne ustawienie, dzięki czemu zaznaczony tekst jest usuwany podczas wpisywania. Sekcja `:custom` włącza klawisze Page Up i Page Down do przewijania do góry lub do dołu bufora. Sekcja ta redefiniuje również sposób, w jaki Emacs definiuje zdanie (patrz sekcja 7.3). Ostatnia

zmienna zapisuje każdy istniejący tekst ze schowka do pierścienia kill, aby zapewnić lepszą współpracę między schowkiem systemu operacyjnego a pierścieniem kill programu Emacs.

```
(use-package text-mode
 :ensure
 nil
 :hook
 (tryb tekstowy . tryb linii wizualnej)
 :init
 (tryb usuwania zaznaczenia t)
 :custom
 (podwójna spacja na końcu zdania
 nil) (błąd przewijania w górę/w dół
 t)
 (zapisz-międzyprogramowe-wklejanie-przed-usunięciem t))
```

### *Spelleeteekeig*

Pisanie bez automatycznej kontroli ortograficznej byłoby dość uciążliwe nawet dla najbardziej doświadczonych autorów. Pakiet flyspell współpracuje z oprogramowaniem Hunspell i odpowiednim słownikiem, aby na bieżąco sprawdzać pisownię.

Należy zmienić standardowy słownik na lokalną odmianę, dostosowując zmienną `ews-hunspell-dictionaries`. EWS używa tej konkretnej zmiennej, ponieważ słowniki są ustawione w dwóch miejscach, aby umożliwić wielojęzyczną pisownię. Można ustawić wiele słowników dla tego samego bufora. W mojej konfiguracji używam „en\_AU,nl\_NL”, dzięki czemu mogę pisać zarówno w australijskiej odmianie języka angielskiego, jak i w języku niderlandzkim bez konieczności zmiany słowników. Sekcja 4.5 wyjaśnia, jak korzystać z tego pakietu.

```
(use-package flyspell
 :custom
 (ispell-program-name „hunspell”)
 (ispell-dictionary ews-hunspell-dictionaries)
 (flyspell-mark-duplications-flag nil) ;; Tryb Writegood wykonuje tę
 czynność (org-fold-core-style 'overlays) ;; Napraw błąd trybu Org
```

```

:config
(spell-set-spellchecker-params)
(spell-hunspell-add-multi-dic ews-hunspell-dictionaries)
:hook
(tryb tekstowy . tryb flyspell)
:bind
(„C-c w s s” . ispell)
("C-;" . flyspell-auto-correct-previous-word)))

```

### *Tryb Rieieg Ovg*

Ta część konfiguracji ustawia szereg zmiennych w celu poprawy wyglądu buforów Org. Aby dowiedzieć się, do czego służą te zmienne, użyj C-h v i wprowadź nazwę zmiennej.

Org posiada wiele zmiennych, które pozwalają zmieniać jego interfejs. Możesz dodać inne zmienne lub usunąć niektóre, aby Org wyglądał tak, jak chcesz. Na przykład, aby włączyć listy alfabetyczne i numeryczne, musisz dostosować zmienną `org-list-allow-alphabetical` do t. Spowoduje to dodanie a., A., a) i A) jako dodatkowych opcji numeracji listy.

```

(use-package org
 :custom
 (org-startup-indented t) (org-
hide-emphasis-markers t)
 (org-startup-with-inline-images t)
 (org-image-actual-width '(450)) (org-
pretty-entities t)
 (org-use-sub-superscripts „{ }”)
 (org-id-link-to-org-use-id t)
 (org-fold-catch-invisible-edits 'show))

```

Powyższy fragment kodu ukrywa znaczniki podkreślenia, aby ekran był bardziej przejrzysty. Znaczniki podkreślenia to symbole używane do wskazania kursywy, pogrubienia i innych dekoracji czcionki, na przykład \_kursywa\_. Ukrywanie składni dokumentu tekstowego nie jest idealnym rozwiązaniem, ponieważ zaciemnia istotne informacje. Pakiet Org Appear autorstwa Alice P. Hacker

pokazuje ukryte znaczniki w buforach Org, gdy kursor znajduje się na podkreślonym słowie, co daje nam najlepsze z obu światów.

```
(use-package org-appear
 :hook
 (org-mode . org-appear-mode))
```

Pakiet Org fragtog jest podobny do Org Appear, ale przeznaczony dla fragmentów kodu LaTeX. Automatycznie przełącza podgląd fragmentów kodu LaTeX w trybie Org, gdy kursor wchodzi w nie lub z nich wychodzi. Domyślnie tekst jest mały i może stać się nieczytelny podczas przełączania między ciemnym a jasnym motywem.

Zmienna `org-format-latex-options` kontroluje sposób wyświetlania fragmentów przez Emacs. Zmienna ta jest listą zawierającą takie właściwości, jak kolory i rozmiar. Funkcja `plist-put` pozwala zmieniać opcje na liście. Pierwszy plan i tło są ustawione na ten sam kolor co tekst. Jeśli zmienisz tryb z ciemnego na jasny lub odwrotnie, powinieneś wywołać funkcję `org-latex-preview` (`C-c C-x C-l`), aby zmienić podgląd obrazów.

Automatyczne podglądy LaTeX są wyłączone, ponieważ mogą opóźniać ładowanie strony i powodować problemy, gdy użytkownik nie ma zainstalowanego LaTeX.

```
(use-package org-fragtog
 :after org
 :hook
 (org-mode . org-fragtog-mode)
 :custom
 (org-startup-with-latex-preview nil)
 (org-format-latex-options
 (plist-put org-format-latex-options :scale 2)
 (plist-put org-format-latex-options :foreground 'auto) (plist-
 put org-format-latex-options :background 'auto)))
```

Ostatnim pakietem modyfikującym bufor Org jest Org Modern. Jednak większość funkcji została wyłączona, ponieważ dla początkujących użytkowników lepiej jest nie ukrywać

symbole semantyczne. Możesz eksperymentować ze zmianą tych ustawień, aby zmienić wygląd i działanie buforów Org.

```
(use-package org-modern
 :hook
 (org-mode . org-modern-mode)
 :custom
 (org-modern-table nil) (org-
modern-keyword nil) (org-
modern-timestamp nil) (org-
modern-priority nil) (org-
modern-checkbox nil) (org-
modern-tag nil)
(nazwa bloku org-modern nil)
(słowo kluczowe org-modern
nil) (przypis org-modern nil)
(org-modern-internal-target nil)
(org-modern-radio-target nil) (org-
modern-statistics nil) (org-modern-
progress nil))
```

## Inspiracja

### *Czytanie e-booków*

Wbudowany pakiet Doc-View może odczytywać różne formaty plików przy pomocy zewnętrznego oprogramowania. Ta konfiguracja zwiększa rozdzielczość generowanego pliku obrazu i podnosi próg ostrzegania przed otwarciem dużych plików do pięćdziesięciu MB ( $50 \times 2^{20}$ ). Sekcja

5.1 wyjaśnia, jak korzystać z tego pakietu.

Do odczytu plików PDF wymagany jest pakiet GhostScript lub MuPDF. Jeśli dostępny jest pakiet Poppler, można przekonwertować plik PDF na tekst, aby ułatwić wyszukiwanie i kopiowanie. Aby wyświetlić pliki DjVu, potrzebna jest biblioteka DjVuLibre do ich analizowania.



```
(use-package doc-view
 :custom
 (doc-view-resolution 300)
 (large-file-warning-threshold (* 50 (expt 2 20))))
```

Pakiet Nov autorstwa Vasilija Schneidermanna zapewnia cenne funkcje do przeglądania książek w formacie ePub w programie Emacs. Sekcja init gwarantuje, że każdy plik z rozszerzeniem epub jest powiązany z tym pakietem. Plik ePub jest zasadniczo skompresowaną stroną internetową, więc do odczytania tych plików potrzebny jest program Zip. Informacje na temat odczytu plików ePub znajdują się w sekcji 5.1.

```
(use-package nov
 :init
 (add-to-list 'auto-mode-alist '("\\.epub\\'" . nov-mode)))
```

Emacs może odczytywać dokumenty utworzone w standardowym oprogramowaniu biurowym. W tym celu konwertuje te pliki do formatu PDF za pomocą LibreOffice i wyświetla je w tej postaci.

### *Bibliogvapties*

Te linie kodu dodają dwa typy pól do wpisów BibTeX: słowa kluczowe, które pomagają uporządkować literaturę, oraz link do pliku, dzięki czemu można odczytać wszelkie załączniki w Emacs.

Funkcja `ews-register-bibtex` przypisuje wszystkie pliki z rozszerzeniem `.bib` w katalogu przechowywanym w zmiennej `ews-bibliography-directory` do listy globalnych plików BibTeX. Należy ustawić tę zmienną na lokalizację, w której przechowuje się bibliografię. Sekcja 5.2 wyjaśnia tworzenie bibliografii i zarządzanie nią. Używamy specjalnej zmiennej, aby zapewnić, że inne zmienne mają tę samą wartość.

```
(use-package bibtex
 :custom
 (bibtex-user-optional-fields
 '(("keywords" „Słowa kluczowe opisujące wpis" „")
 („plik" „Względna lub bezwzględna ścieżka do załączników" „"))))
```

```
(bibtex-align-at-equal-sign t)
:config
(ews-bibtex-register)
:bind
(„C-c w b r” . ews-bibtex-register)))
```

Pakiet Biblio umożliwia wyodrębnianie literatury z różnych baz danych.

```
(use-package biblio
:bind
(„C-c w b b” . ews-bibtex-biblio-lookup)))
```

Citar to pakiet służący do zarządzania bibliografią i cytatami. Zapewnia on interfejs między tekstem a bibliografią.

```
(use-package citar
:defer t
:custom
(citar-bibliography ews-bibtex-files)
:bind
(„C-c w b o” . citar-open)))
```

### *Gotowe strony internetowe*

Pakiet Elfeed pomaga w odczytywaniu plików RSS, a pakiet Elfeed-Org pozwala konfigurować kanały RSS za pomocą pliku Org. Repozytorium EWS zawiera przykładowy plik. Więcej informacji na temat korzystania z tego narzędzia można znaleźć w sekcji 5.3. Elfeed używa oprogramowania cURL do pobierania kanałów. Jeśli oprogramowanie to jest niedostępne, używa wolniejszej wersji wbudowanej w Emacs.

```
(use-package elfeed
:custom
(elfeed-db-directory
(expand-file-name "elfeed" user-emacs-directory)))
```

```
(elfeed-show-entry-switch 'display-buffer)
:bind
("C-c w e" . elfeed))

(use-package elfeed-org
 :config
 (elfeed-org)
 :custom
 (rmh-elfeed-org-files
 (lista (concat (nazwa-pliku-jako-katalog (getenv „HOME”))
 „elfeed.org”))))
```

Pakiet Org-Webtools ułatwia wstawianie hiperłączy poprzez konwersję zawartości pętli kill ring na hiperłącze Org.

```
(use-package org-web-tools
 :bind
 („C-c w w” . org-web-tools-insert-link-for-url)))
```

### *Odtwarzanie plików multimedialnych*

Pakiet EMMS (Emacs MultiMedia System) zapewnia interfejs do różnych odtwarzaczy multimedialnych. Musisz mieć zainstalowany jeden z tych programów: mpg321, ogg123 (vorbis-tools), mplayer, mpv lub VLC.

```
(use-package emms
 :config
 (require 'emms-setup)
 (require 'emms-mpris)
 (emms-all)
 (emms-default-players)
 (emms-mpris-enable)
 :custom
 (emms-browser-covers #'emms-browser-cache-thumbnail-async)
 :bind
```

```
(„C-c w m b” . emms-browser)
(„C-c w m e” . emms)
("C-c w m p" . emms-play-playlist)
("<XF86AudioPrev>" . emms-previous)
("<XF86AudioNext>" . emms-next)
("<XF86AudioPlay>" . emms-pause)))
```

*Otwórz pliki witł za pomocą oprogramowania exteveal*

Pakiet OpenWith autorstwa Markusa Triska umożliwia otwieranie plików w zewnętrznym oprogramowaniu. Więcej szczegółów można znaleźć w sekcjach 5.5 i 5.4.

```
(use-package openwith
 :config (openwith-
 mode t)
 :custom
 (openwith-associations nil))
```

## **Ideation**

*Ovg eaptuve*

Możliwości szablonów przechwytywania są szerokie i zależą od konkretnych przypadków użycia. Ta konfiguracja jest jedynie przykładem opcji. Dokumentacja Org zawiera wiele szczegółowych informacji (C-h R org <ret> capture). Domyślne ustawienia Org zapisują Twoje ulotne notatki w ukrytym pliku (~/.notes.org). Dostosuj zmienną org-default-notes-file.

```
(use-package org
 :bind
 („C-c c” . org-capture)
 ("C-c l" . org-store-link))
:custom
(szablony org-capture
```

```
'(("f" "Fleeting note" item
 (plik+nagłówek org-default-notes-file „Notatki”) „-
 %?“)
 („p” „Trwała notatka” zwykły
 (plik oznacz ostatnią
 ścieżkę) #'oznacz-org-capture
 :no-save t
 :natychmiastowe zakończenie nil
 :kill-buffer t
 :przejdź-do-
 przechwyconego t) („t”
 „Nowe zadanie” wpis
 (plik+nagłówek org-default-notes-file „Zadania”) „*
 TODO %i%?“))))
```

### *Deeote*

Denote to elastyczny pakiet do tworzenia notatek i zarządzania plikami. Zapoznaj się z sekcją 5.4 lub obszernym podręcznikiem Denote, dostępnym po wpisaniu C-h R denote. Musisz skonfigurować co najmniej zmienną denote-directory, aby wskazać lokalizację swoich notatek.

Pakiet EWS zawiera wygodną funkcję poprawiającą sposób wyświetlania przez Denote linków do załączników, powiązaną z funkcją denote-link-description-function.

```
(use-package denote
 :defer t
 :custom
 (denote-sort-keywords t)
 (denote-link-description-function #'ews-denote-link-description-title-case) (denote-
 rename-buffer-mode 1)
 :hook
 (tryb-dired . tryb-dired)
 :custom-face
 (denote-faces-link ((t (:slant italic))))
 :bind
```

```
(„C-c w d b” . denote-find-backlink) („C-
c w d d” . denote-date)
("C-c w d l" . denote-find-link)
("C-c w d i" . denote-link-or-create)
("C-c w d k" . denote-rename-file-keywords)
("C-c w d n" . denote)
(„C-c w d r” . denote-rename-file)
(„C-c w d R” . denote-rename-file-using-front-matter)))
```

Twórcy Denote zapewniają ekosystem pakietów, które zwiększają funkcjonalność, z których dwa są włączone w EWS.

```
(use-package denote-journal)
```

```
(use-package denote-org
:bind
(„C-c w d h” . denote-org-link-to-heading))) (use-
package denote-sequence)
```

Pakiet Consult zapewnia kilka wygodnych funkcji, które pod wieloma względami ułatwiają pracę w Emacsie. Funkcja `consult-grep` pozwala przeszukiwać pliki w bieżącym katalogu. Funkcja wyszukiwania wymaga dostępu do oprogramowania Grep. Polecenie `consult-org-heading` zapewnia spis treści pliku w trybie Org, umożliwiając szybkie poruszanie się po dużym pliku.

Consult ma wiele innych funkcji, które zastępują niektóre podstawowe polecenia Emacs'a wygodniejszymi wersjami. Większość z nich nie została włączona, aby zapewnić jak największą zgodność z podstawową wersją Emacs'a. Dokumentacja Consult zawiera szczegółowe opisy tych ulepszonych poleceń.

```
(use-package consult
:bind
(„C-c w h” . consult-org-heading))
```

```
("C-c w g" . consult-grep))
:config
(dodaj do listy „consult-preview-allowed-hooks” „visual-line-mode”))
```

Consult Notes to wygodny pakiet oparty na Consult. Zapewnia dostęp do plików Denote, a także umożliwia przeszukiwanie notatek.

```
(use-package consult-notes
 :custom
 (consult-notes-denote-display-keywords-indicator "_")
 :bind
 (("C-c w d f" . consult-notes)
 ("C-c w d g" . consult-notes-search-in-all-notes))
 :init
 (tryb-notatek-denote))
```

Pakiet Citar-Denote pozwala tworzyć relacje wiele-do-wielu między notatkami Denote a pozycjami w bibliografii (sekcja 5.4).

```
(use-package citar-denote
 :custom
 (citar-open-always-create-notes t)
 :init
 (citar-denote-mode)
 :bind
 („C-c w b c” . citar-create-note)
 („C-c w b n” . citar-denote-open-note)
 („C-c w b x” . citar-denote-nocite)
 :map org-mode-map
 ("C-c w b k" . citar-denote-add-citekey) ("C-c
w b K" . citar-denote-remove-citekey) ("C-c w b
d" . citar-denote-dwim)
 („C-c w b e” . citar-denote-open-reference-entry)))
```

Pakiet Denote-Explore zapewnia wygodne funkcje do zarządzania kolekcją notatek i załączników (sekcja 5.5).

```
(use-package denote-explore
 :bind
 (;; Statystyki
 („C-c w x c” . denote-explore-count-notes) („C-c w
 x C” . denote-explore-count-keywords)
 ("C-c w x b" . denote-explore-barchart-keywords) ("C-c
 w x e" . denote-explore-barchart-filetypes)
 ;; Losowe spacery
 („C-c w x r” . denote-explore-random-note) („C-c w
 x l” . denote-explore-random-link) („C-c w x k” .
 denote-explore-random-keyword) („C-c w x x” .
 denote-explore-random-regex)
 ;; Oznaczaj konserwatora
 („C-c w x d” . denote-explore-identify-duplicate-notes) („C-c
 w x z” . denote-explore-zero-keywords)
 („C-c w x s” . denote-explore-single-keywords) („C-
 c w x o” . denote-explore-sort-keywords) („C-c w x
 w” . denote-explore-rename-keyword)
 ;; Wizualizacja oznaczenia
 („C-c w x n” . denote-explore-network)
 („C-c w x v” . denote-explore-network-regenerate) (
 „C-c w x D” . denote-explore-barchart-degree)))
```

## Produkcja

### *Maeagieg tle wvitięg pvoeess*

Repozytorium EWS udostępni kilka plików ułatwiających pracę z Org, służących do wstawiania notatek, szuflad i liczenia słów. Na tym etapie polecenie screenshot ma charakter eksperymentalny.

```
(use-package org
 :bind
 (:map org-mode-map
 ("C-c w n" . ewe-org-insert-notes-drawer)
```



```
("C-c w p" . ews-org-insert-screenshot)
("C-c w c" . ews-org-count-words)))
```

Pakiet Olivetti usuwa elementy rozpraszające uwagę z ekranu i przekształca sesję Emacsa w elektroniczną maszynę do pisania. Funkcja `ews-olivetti` zapisuje konfigurację ekranu przed aktywacją trybu pisania bez rozpraszania uwagi. Po wyłączeniu trybu Olivetti za pomocą tej funkcji przywracana jest pierwotna konfiguracja.

```
(use-package olivetti
 :demand t
 :bind
 (("C-c w o" . ews-olivetti)))
```

Vundo zapewnia graficzny podgląd różnych wersji bieżącego bufora.

```
(use-package vundo
 :bind
 („C-M-/” . vundo)))
```

### *Cytaty*

Ta konfiguracja ustawia globalną bibliografię równą zmiennej `ews-bibtex-files`. Aby ustawić tę zmienną, skonfiguruj katalog `ews-bibtex-directory` w celu rejestrowania plików bibliograficznych i uruchamiaj funkcję `ews-bibtex-register` za każdym razem, gdy dodajesz nowe pliki bib.

```
(require 'oc-natbib)
(require 'oc-csl)
```

```
(setq org-cite-global-bibliography ews-bibtex-files org-
 cite-insert-processor 'citar
 org-cite-follow-processor 'citar org-
 cite-activate-processor 'citar)
```

*Jakość asuwae*

Emacs może połączyć się z serwerem słownika pod adresem dict.org w celu uzyskania definicji i tezauryusa. Szczegółowe informacje znajdują się w sekcji 7.2.

```
(use-package dictionary
 :custom
 (dictionary-server "dict.org")
 :bind
 (("C-c w s d" . dictionary-lookup-definition)))
```

Pakiet Writegood pomaga wykrywać niejasne sformułowania, styl pasywny i powtarzające się słowa. Zawiera również funkcje służące do oceny złożoności tekstu za pomocą testu Flescha-Kincaida.

```
(use-package writegood-mode
 :bind
 („C-c w s r” . writegood-reading-ease))
:hook
(tryb tekstowy . writegood-mode))
```

Pakiet TitleCase dąży do jak najdokładniejszego stosowania wielkich liter w tytułach zdań, wierszy i fragmentów tekstu w prozie angielskiej. Można dostosować zmienną titlecase-style. Funkcja EWS może wykonać to zadanie dla wszystkich nagłówków w pliku Org, aby zapewnić (sekcja 7.2).

```
(use-package titlecase
 :bind
 („C-c w s t” . titlecase-dwim)
 ("C-c w s c" . ews-org-headings-titlecase)))
```

### *Skróty*

Tryb skrótów to wbudowany program, który pomaga przyspieszyć pisanie poprzez definiowanie skrótów i typowych błędów ortograficznych oraz automatyczne zastępowanie ich słowami, zdaniami lub całymi akapitami.

```
(add-hook „text-mode-hook” „abbrev-mode”)
```

Generator Lorem Ipsum może być pomocny podczas projektowania układu dokumentu. Ten pakiet wstawia do bufora fikcyjny tekst łaciński.

```
(use-package lorem-ipsum
 :custom
 (lorem-ipsum-list-bullet "- ") ;; Punktory trybu Org
 :init
 (setq lorem-ipsum-sentence-separator
 (if sentence-end-double-space " " " "))
 :bind
 („C-c w s i” . lorem-ipsum-insert-paragraphs)))
```

### *Vevsioe eoetvol*

Wbudowany pakiet Ediff porównuje różne pliki i pokazuje różnice między nimi. Pozwala również zdecydować, w jaki sposób połączyć dwie wersje, podobnie jak funkcja śledzenia zmian w edytorze tekstu. Rodzina funkcji ediff domyślnie nie dzieli okien w sposób przejrzysty, więc te ustawienia sprawiają, że program jest łatwiejszy w użyciu.

Zaawansowana kontrola wersji wymaga systemu kontroli wersji, takiego jak Git.

```
(use-package ediff
 :ensure nil
 :custom
 (ediff-keep-variants nil)
 (ediff-split-window-function 'split-window-horizontally)
 (ediff-window-setup-function 'ediff-setup-windows-plain))
```

*Otłev tekst ie tryby*

Org jest fantastyczny, ale nie jest to jedyny tryb tekstowy przydatny dla autorów. EWS instaluje zarówno Markdown, jak i fountain. Szczegółowe informacje można znaleźć w sekcji 7.5.

```
(use-package fountain-mode)
(use-package markdown-mode)
```

**Publikacja***Ustawienia podstawowe*

Ten fragment kodu ustawia podstawowe opcje eksportu dla trybu org. Możesz ustawić je jako zmienne, żeby były domyślnie stosowane do wszystkich plików, albo wstawić je jako słowa kluczowe w nagłówku. Sygnatura czasowa eksportowanych plików jest ustawiona na europejski format daty: dzień, miesiąc i rok. Jeśli publikujesz dla amerykańskiej publiczności, możesz zmienić `org-export-date-timestamp-format` na „`%B %e %Y`”. Litery te oznaczają pełną nazwę miesiąca, numer dnia bez zer wiodących oraz rok w czterech cyfrach. Szczegółowe informacje na temat formatowania dat w innych metodach można znaleźć w dokumentacji funkcji `format-time-string`. Szczegółowy opis możliwych konfiguracji znajduje się w sekcji Ustawienia eksportu w podręczniku Org.

```
(użyj pakietu org
:custom
(org-export-with-drawers nil)
(org-export-with-todo-keywords nil)
(org-export-with-toc nil)
(org-export-with-smart-quotes t)
(org-export-date-timestamp-format „%e %B %Y”))
```

*Zpub*

Pakiet `ox-ePub` eksportuje pliki Org do najpopularniejszego formatu e-booków. Kod `ox-org` eliminuje niektóre problemy związane ze spisem treści (sekcja 8.5).

```
(use-package ox-epub
 :demand t
 :init
 (require 'ox-org))
```

### *LaTeX*

Ta część konfiguracji definiuje proces eksportu z Org do TEX do PDF. Ta konfiguracja usuwa również wszelkie pliki tymczasowe utworzone w trakcie procesu. Oczywiście potrzebna będzie działająca wersja LaTeX z zainstalowanymi na komputerze wszystkimi odpowiednimi pakietami.

```
(use-package ox-latex
 :ensure nil
 :demand t
 :custom
 ;; Wielokrotne przejścia LaTeX dla bibliografii
 (org-latex-pdf-process
 („„pdflatex -interaction nonstopmode -output-directory %o %f” „bibtex
 %b”
 „pdflatex -shell-escape -interaction nonstopmode -output-directory %o %f” „pdflatex -
 shell-escape -interaction nonstopmode -output-directory %o %f”))
 ;; Wyczyść pliki tymczasowe po eksporcie
 (org-latex-logfiles-extensions
 (quote ("lof" "lot" "tex~" "aux" "idx" "log" "out"
 „toc” „nav” „snm” „vrb” „dvi” „fdb_latexmk”
 „blg” „brf” „fls” „entoc” „ps” „spl” „bbl” „tex”
 „bcf”))))
```

W następnej sekcji zdefiniowano klasę dokumentu EWS, która służy do tworzenia papierowej wersji tej książki. Pierwsza część kodu definiuje nazwę używaną w pliku Org, w tym przypadku `#+latex_class: ews`. Następna część to preambuła w kodzie LaTeX. Należy pamiętać, że znaki odwrotnego ukośnika muszą być poprzedzone dwoma znakami odwrotnego ukośnika. Org dodaje również standardowe pakiety. Szczegółowe informacje na temat modyfikacji można znaleźć w dokumentacji `org-latex-classes`.

Sformatyzuj standardowe włączenia. Ostatnia sekcja definiuje sposób tłumaczenia poziomów nagłówków w pliku Org na polecenia LaTeX. Ten kod definiuje pierwsze trzy poziomy nagłówków Org.

```
(with-eval-after-load 'ox-latex
 (add-to-list
 'org-latex-classes
 '("ews"
 "\\documentclass[11pt, twoside, hidelinks]{memoir}
 \\setstocksize{9.25in}{7.5in}
 \\settrimmedsize{\\stockheight}{\\stockwidth}{*}
 \\setlrmarginsandblock{1,5 cala}{1 cal}{*}
 \\setulmarginsandblock{1in}{1.5in}{*}
 \\sprawdź i popraw układ
 \\layout
 \\ustaw licznik głębokości spisu treści na 0
 \\odnów polecenie{\\baselinestretch}{1,25}
 \\ustaw nagłówek i stopkę{0,5 cala}{0,75 cala}
 \\setlength{\\footskip}{0,8 cala}
 \\chapterstyle{bianchi}
 \\setseheadstyle{\\normalfont \\raggedright \\textbf}
 \\setsubseheadstyle{\\normalfont \\raggedright \\emph}
 \\setsubsubseheadstyle{\\normalfont\\centering}
 \\pagestyle{myheadings}
 \\usepackage[font={small, it}]{caption}
 \\usepackage{ccicons}
 \\usepackage{ebgaramond}
 \\usepackage[authoryear]{natbib}
 \\bibliographystyle{apalike}
 \\usepackage{svg}
 \\hyphenation{mini-buffer}" ("\\chapter{%s}" .
 "\\chapter*{%s}")
 ("\\section{%s}" . "\\section*{%s}") ("\\subsection{%s}" .
 "\\subsection*{%s}") ("\\subsubsection{%s}" .
 "\\subsubsection*{%s}")))))
```

## Administracja

### *Gettieg Tliegs Doe*

Konfiguracja Org do zarządzania działaniami i projektami ustawia niestandardowy element agendy, który pokazuje agendę na najbliższe trzy dni, listę zadań oznaczonych jako NEXT oraz listę zadań oznaczonych jako WAIT.

Zmienna `org-agenda-custom-commands` zapewnia bardzo elastyczny system tworzenia widoków agendy. Można na przykład utworzyć agendę dla działań prywatnych i agendę dla pracy (sekcja y.1).

```
(use-package org
 :custom
 (org-agenda-custom-commands
 '(("e" "Agenda, następne działania i oczekiwanie"
 ((agenda "" ((org-agenda-overriding-header „Najbliższe trzy dni:”)
 (org-agenda-span 3)
 (org-agenda-start-on-weekday nil)))
 (todo „NEXT” ((org-agenda-overriding-header „Następne działania:”)))
 (todo „WAIT” ((org-agenda-overriding-header „Oczekiwanie:”))))))
 :bind
 („C-c a” . org-agenda)))
```

### *Pliki Maeage*

Pakiet Dired jest wygodnym i potężnym narzędziem do porządkowania dysków i uzyskiwania dostępu do informacji. Dired wyświetla pliki i katalogi w porządku alfabetycznym. Ja preferuję inny widok, w którym katalogi są wyświetlane u góry, a pliki poniżej. Zmienna o nazwie `dired-listing-switches` określa sposób wyświetlania plików w buforze Dired.

Zmienna `dired-dwim-target` instruuje Emacsa, aby odgadł domyślny katalog docelowy. Oznacza to, że jeśli bufor Dired jest wyświetlany w innym oknie, używa on tego katalogu zamiast bieżącego katalogu bufora Dired. To ustawienie ułatwia kopiowanie lub przenoszenie plików między katalogami.

Zmienna `delete-by-moving-to-trash` przenosi usunięte pliki do kosza zamiast całkowicie je usuwać, dzięki czemu zmniejsza się ryzyko utraty cennych danych.

Ostatnia linia w tej części konfiguracji umożliwia otwieranie nowych katalogów w tym samym buforze co bieżący (za pomocą klawisza `a`), zapobiegając zaśmiecaniu sesji buforami `Dired`. Przy pierwszym użyciu Emacs poprosi o potwierdzenie, czy chcesz używać tej opcji.

```
(use-package dired
 :ensure nil
 :commands
 (dired dired-jump)
 :custom
 (dired-listing-switches
 „-goah --group-directories-first --time-style=long-iso”)
 (dired-dwim-target t)
 (usuń-przenosząc-do-kosza t)
 :init
 (put 'dired-find-alternate-file 'disabled nil))
```

Domyślnym ustawieniem programu `Dired` jest wyświetlanie ukrytych plików, nawet jeśli są one ukryte z jakiegoś powodu. Ta konfiguracja wykorzystuje tryb `dired-omit-mode`, aby usunąć te ukryte pliki z widoku. Można przełączać to zachowanie za pomocą klawisza kropki.

```
(use-package dired
 :ensure nil
 :hook (dired-mode . dired-omit-mode)
 :bind (:map dired-mode-map
 („.” . dired-omit-mode))
 :custom (dired-omit-files „^\\.[a-zA-Z0-9]+”))
```

Kolejna część konfiguracji określa sposób zarządzania automatycznymi kopiami zapasowymi przez Emacs. Domyślnym ustawieniem jest przechowywanie tych plików w folderze, w którym znajdują się oryginalne pliki, co powoduje zaśmiecanie folderów kopiami Twoich plików.



Poniższe ustawienie modyfikuje zmienną `backup-directory-alist` tak, aby Emacs zapisywał wszystkie kopie zapasowe (oznaczone symbolem „.”) w podkatalogu `bak` folderu `init`. Alternatywnie można nakazać Emacsowi, aby nie zapisywał kopii zapasowych, ustawiając `make-backup-files` na `nil`. Ja wolę zachowywać kopie zapasowe, ponieważ kilka razy uratowały mi skórę.

Ta konfiguracja eliminuje również pliki blokujące, które są przydatne tylko podczas pracy w folderach współdzielonych. Pliki blokujące uniemożliwiają innym użytkownikom otwarcie pliku, gdy inny użytkownik już go edytuje, ale powodują duże zagrzenie podczas samodzielnego pisania. Zmień tę zmienną na `t`, jeśli współpracujesz z innymi lub zarządzasz plikami w wielu systemach za pośrednictwem usługi udostępniania plików, takiej jak Nextcloud.

```
(setq-default backup-directory-alist
 '(("." . ,(expand-file-name "backups/" user-emacs-directory)))
 version-control t
 delete-old-versions t
 create-lockfiles nil)
```

Emacs zapisuje listę ostatnio używanych plików za pomocą pakietu `recentf`. Pakiet ten przechowuje listę ostatnio otwartych plików i ułatwia ich otwieranie. Lista ostatnio używanych plików jest automatycznie zapisywana między sesjami Emacs'a. Domyślnie tryb ostatnio używanych plików przechowuje dwadzieścia ostatnio otwartych plików, co można zmienić, dostosowując zmienną `recentf-max-saved-items`, która w EWS wynosi pięćdziesiąt.

```
(use-package recentf
 :config (recentf-
 mode t)
 :custom
 (recentf-max-saved-items 50)
 :bind
 („C-c w r” . recentf-open)))
```

Ten ostatni pakiet plików umożliwia ustawienie zakładek dla ulubionych lokalizacji.

Flaga `bookmark-save-flag` jest ustawiona na jeden, więc plik zakładek jest zapisywany za każdym razem, gdy dodajesz

nowy. Wartość domyślna zapisuje go tylko po zamknięciu programu Emacs, co oznacza, że w mało prawdopodobnym przypadku awarii programu Emacs lub systemu można utracić zakładki.

```
(use-package bookmark
 :custom
 (bookmark-save-flag 1)
 :bind
 ("C-x r d" . bookmark-delete))
```

### *Wyświetlanie obrazów*

Emacs posiada dwa tryby przeglądania i zarządzania obrazami. Przeglądarka obrazów wyświetla pojedyncze obrazy, ale można również przeglądać katalog za pomocą klawiszy strzałek w lewo i w prawo.

Aby umożliwić edycję obrazów, należy zainstalować program ImageMagic.

Klawisze C-<ret> otwierają obraz w buforze Dired w Twoim ulubionym edytorze. Zmienna `image-dired-external-viewer` określa program używany do edycji obrazów, w moim przypadku jest to GIMP, czyli GNU Image Manipulation Program.

```
(use-package emacs
 :custom
 (image-dired-external-viewer "gimp")
 :bind
 ((:map image-mode-map
 ("k" . image-kill-buffer) ("<right>" .
 image-next-file) ("<left>" . image-
 previous-file))
 (:map dired-mode-map
 ("C-<return>" . image-dired-dired-display-external))))
```

Wbudowany pakiet Image-Dired może generować miniatury z bufora Dired i umożliwia pracę nad obrazami z tego miejsca.

```
(use-package image-dired
 :bind
 („C-c w I” . image-dired))
(:map image-dired-thumbnail-mode-map
 ("C-<right>" . image-dired-display-next)
 ("C-<left>" . image-dired-display-previous)))
```

### *Dostosowania*

Dostęp do funkcji dostosowywania zmiennych można uzyskać za pomocą skrótu C-c w v.

```
(keymap-global-set „C-c w v” 'customize-variable)
```

Wszelkie zmiany wprowadzone przez system dostosowywania Emacs w EWS są zapisywane w pliku custom.el, a nie bezpośrednio w pliku inicjalizacyjnym. Takie podejście zapobiega modyfikowaniu pliku inicjalizacyjnego przez system dostosowywania. Plik custom.el jest ładowany, gdy jest dostępny. Jeśli zmienne są ustawione zarówno w pliku custom, jak i w pliku inicjalizacyjnym, pierwszeństwo ma plik custom.el.

```
(setq-default custom-file (expand-file-name
 „custom.el”
 user-emacs-directory))
```

```
(load custom-file :no-error-if-file-is-missing)
```



---

## Referencje

---

- Ahrens, S. (2017). *Jak robić inteligentne notatki: prosta technika zwiększająca wiedzę, pozostawiająca trwałe ślady w pamięci: dla studentów, pracowników akademickich i autorów książek*. North Charleston, SC: CreateSpace.
- Allen, D. (2005). *Jak osiągnąć sukces: sekret produktywności bez stresu*. Londyn: Piatkus.
- Berry, R. (1988). Common User Access. Spójny i użyteczny interfejs człowiek-komputer dla środowisk SAA. *Ibm systems journal*, 27(3), 281–300. doi: 10.1147/sj.273.0281
- Blevins, J. R. (2017). *Przewodnik po trybie Mavkdowne dla Zmaes*. LeanPub.
- Bottiroli, S., Rosi, A., Russo, R., Vecchi, T. i Cavallini, E. (2014). Wpływ słuchania muzyki w tle na funkcje poznawcze osób starszych: szybkość przetwarzania informacji poprawia się przy muzyce w szybkim tempie, natomiast pamięć wydaje się korzystać zarówno z muzyki w szybkim, jak i wolnym tempie. *Fvoetievs ie agieg eeuvoseieeee*, 6. doi: 10.338y/fnagi.2014.00284
- Cameron, D. (2005). *Leaveieg GNU Zmaes* (wydanie trzecie). Sebastopol, Kalifornia: O'Reilly.
- Clark, A. i Chalmers, D. (1988). The Extended Mind. *Aealysis*, 7(1), 7–19. doi: 10.10y3/analysis/58.1.7
- Covey, S. R. (1989). *The Sevee Habits of Higltly Zffeetive People: Restovieg tle Cłavaetev Zttlie* (wydanie pierwsze). Nowy Jork: fireside Book.

- Even-Ezra, A. (2021). *Lies of Thought: Bizarre Diagrams and the Medieval Mind*. Chicago: The University of Chicago Press.
- Ferriss, T. (2011). *The 4-Hour Work Week. Escape the 9–5, Live Anywhere and Join the New Rich*. Londyn: Vermilion.
- forte, T. (2022). *Budowanie silnej marki: sprawdzona metoda na poprawę cyfrowego życia i odkrycie swojego kreatywnego potencjału*. Londyn: Profile Books Ltd.
- fox, J., & Tigchelaar, M. (2015). Creating an engineering academic formulas list. *Journal of Teaching Zeglist for Specific and Academic Purposes*, 3(2), 295–304.
- Johnson, T. (2022). Emacs jako narzędzie współczesnej nauki: Wykorzystanie narzędzi open source do usprawnienia procesów naukowych. *Journal of Mathematics Review*, 66 (2), 122–129. doi: 10.1515/205551322x1531595y040478
- Kadavy, D. (2021). *Digital Zettelkasten: Principles, Methods, & Examples*. Kindle Edition.
- Kahn, C. H. (1977). *Plato and the Socratic Dialogue*. Cambridge University Press. doi: 10.1017/cbo978051158557y.013
- Kelly, L. (2015). *Kod pamięci: tradycyjna technika pamięciowa wykorzystująca elementy filozofii, nauki i sztuki*. Allen & Unwin. Ovev.
- Khalili, A. i Auer, S. (2015). WYSIWYM — Zintegrowana wizualizacja, eksploracja i tworzenie treści nieustrukturyzowanych wzbogaconych semantycznie. *Semantic Web*, 6 (3), 259–275. doi: 10.3233/sw-140157
- Kim, K., Erickson, A., Lambert, A., Bruder, G. i Welch, G. (2019). Wpływ trybu ciemnego na zmęczenie wzroku i ostrość widzenia w przezroczystych wyświetlaczach montowanych na głowie. *Symposium Spatial User Interfaces*. ACM. doi: 10.1145/3357251.3357584
- Knuth, D. E. (1984). *The TeXbook*. Reading, Massachusetts: Addison-Wesley.
- König, R. (2020). *Getting yourself organized with Obsidian. A supplement for the video course*. Kurs Udemy.
- Lampert, L. (1994). *LATX: System projektowania oprogramowania: Podręcznik użytkownika i instrukcja obsługi* (wydanie drugie). Reading, Mass: Addison-Wesley Pub. Co.

- Landin, P. J. (1954). Mechaniczna ocena wyrażeń. *The Computer Journal*, 6 (4), 308–320. doi: 10.1093/comjnl/5.4.308
- Lipovetsky, S. (2023). Struktura wskaźników czytelności i optymalne cechy. *Axioms*, 12(5), 421. doi: 10.3390/axioms12050421
- Monnier, S. i Sperber, M. (2020). Ewolucja Emacs Lisp. *Proceedings of the ACM on Programming Languages*, 4(74), 1–55. doi: 10.1145/3385324
- Mueller, P. A. i Oppenheimer, D. M. (2014). Pióro jest potężniejsze niż klawiatura: zalety pisania odręcznego w porównaniu z robieniem notatek na laptopie. *Psychological Science*, 25(5), 1159–1168. doi: 10.1177/0956797614524581
- Omanson, R., Miller, C. S., Young, E. i Schwantes, D. (2010). *Porównanie myszy i keyboarda*. 500–404.
- Petersen, M. (2022). *Mastering Emacs*.
- Pohle, J. i Thiel, T. (2020). Suwerenność cyfrowa. *Internet Policy Review*, 9(4). doi: 10.14753/2020.4.1532
- Prevos, P. (2013). *Perceptions of Magic: Scientific Views of Magical Magic*. Third Hemisphere.
- Rayner, K., Slattery, T., Drieghe, D. i Liversedge, S. (2011). Ruchy gałek ocznych i pomijanie słów podczas czytania: wpływ długości słów i przewidywalności. *Journal of Experimental Psychology: Human Perception and Performance*, 23(2), 514–528.
- Ryder, C. (2021). *Metoda bullet journal*. Londyn: 4th Estate.
- Stallman, R. (2023). *GNU Emacs Manual*. Fundacja Wolnego Oprogramowania. Pobrane z <https://www.gnu.org/software/emacs/manual/emacs.html>
- Stallman, R. M. (1981a). *Emacs manual for its users*. Pobrane z <http://hdl.handle.net/1721.1/6329>
- Stallman, R. M. (1981b). EMACS – rozszerzalny, konfigurowalny edytor tekstu z funkcją samodokumentacji. *Acm SIGOOS Newsletter*, 2(1-2), 147–155. doi: 10.1145/115980.805455
- Stavrou, P. (2024). *Re: Advice regarding editing in emacs*. Pobrane z <https://protesilaos.com/>
- Stephenson, N. (1999). *The beginning was the end of the line*. Avon Books.

- Stickgold, R., Malia, A., Maguire, D., Roddenberry, D. i O'Connor, M. (2000). Powtórka gry: obrazy hipnagogiczne u osób zdrowych i cierpiących na amnezję. *Science*, 290(54y0), 350–353. doi: 10.1125/science.2y0.54y0.350
- Tesler, L. (2012). Osobista historia edycji tekstu bez trybu i funkcji wycinania/kopiowania-wklejania. *letev-aetioes*, 19(4), 70–75. doi: 10.1145/2212877.22128y5
- Tognazzini, B. (1yy2). *Tog oe letevfaae*. Reading, Mass: Addison-Wesley.
- Tognazzini, B. (1yy3). Zasady, techniki i etyka magii scenicznej oraz ich zastosowanie w projektowaniu interfejsów użytkownika. *Pvoeediegs of tle letevaet '93 aed CHI'93 Coefeveeee oe Humaefaeovs ie Computieg Systems*, 355–352. Amsterdam. doi: 10.1145/15y05y.15y284
- Tracy, B. (2015). *Zat Tłat Fvog! 21 Gveat Ways to Stop Pvoevastiatieg aed Get Move Doe ie Less Time*. Londyn: Hodder.
- Travis, B. E. i Waldt, D. C. (2015). Ewolucja systemów publikacji. W: *Tle SGML Implemeetatioe Guide* (s. 21–35). Springer Berlin Heidelberg. doi: 10.1007/y78-3- 542-57850-1\_2
- Umejima, K., Ibaraki, T., Yamazaki, T. i Sakai, K. L. (2021). Notatniki papierowe a urządzenia mobilne: różnice w aktywności mózgu podczas przywoływania wspomnień. *Fvoetievs ie Bełavioval Neuvoseieeee*, 1/, 534158. doi: 10.338y/fnbeh.2021.534158
- University of Chicago Press. (2017). *Tle Ckieago Maeualof Style* (wydanie siedemnaste). Chicago: Wydawnictwo Uniwersytetu Chicagowskiego.
- Watson, D. (2004). *Watsoe's Dietioeavy of Weasel Wovds, Coetempovavy Cliełés, Caet & Maeagemeet Javgoe*. Milsons Point, N.S.W: Knopf.