Wiele plików: bufory, okna, karty

Poziom: średni **Dotyczy: Vim 9.x** **Czas: ~20 min**

Administrator rzadko pracuje na jednym pliku naraz: poprawka w konfiguracji jednej usługi często wymaga zerknięcia do drugiej. Ta strona pokazuje, jak Vim trzyma kilka plików otwartych jednocześnie, oraz dwa narzędzia do samego przeglądania długich plików bez ich edycji.

Fiszka

TYLKO ODCZYT

Otwarcie kolejnego pliku bez zamykania Vima: :e

```
:e /etc/hosts
```

`:e` (edit) otwiera wskazany plik w tym samym oknie, zamiast bieżącego bufora. Poprzedni plik nie znika, zostaje w tle jako bufor, do którego można wrócić.

TYLKO ODCZYT

Lista otwartych buforów: :ls

```
:ls
```

Pokazuje wszystkie pliki otwarte w bieżącej sesji Vima, z numerem, nazwą i znacznikiem, czy mają niezapisane zmiany (`+`).

Przykład:

```
  1 %a   "/etc/fstab"                    line 1
  2      "/etc/hosts"                    line 1
```

TYLKO ODCZYT

Przełączanie między buforami: :bn

```
:bn
:bp
:b2
```

`:bn` (buffer next) przechodzi do kolejnego bufora na liście, `:bp` (buffer previous) do poprzedniego, `:b2` bezpośrednio do bufora o numerze 2 z listy `:ls`.

TYLKO ODCZYT

Podział okna poziomo: :sp

```
:sp /etc/hosts
```

Dzieli okno edytora na górę i dół, pokazując dwa pliki (albo dwa razy ten sam) jednocześnie na ekranie.

TYLKO ODCZYT

Podział okna pionowo: :vsp

```
:vsp /etc/hosts
```

To samo co `:sp`, tylko podział biegnie pionowo, na lewo i prawo. Wygodniejsze przy szerokich monitorach i krótkich liniach konfiguracji.

TYLKO ODCZYT

Przełączanie między oknami: Ctrl+w

```
Ctrl+w Ctrl+w
Ctrl+w h
Ctrl+w l
```

`Ctrl+w` naciśnięte dwa razy z rzędu przeskakuje do następnego okna. `Ctrl+w` a potem `h`, `j`, `k` albo `l` przenosi w konkretnym kierunku, gdy okien jest więcej niż dwa.

◫◫ TYLKO ODCZYT

Nowa karta: :tabnew

```
:tabnew /etc/hosts
```

Otwiera plik w zupełnie nowej karcie, widocznej jako osobna zakładka u góry ekranu. Karty w Vimie grupują okna, nie same pliki: każda karta może mieć swój własny podział na kilka okien.

◫◫ TYLKO ODCZYT

Porównanie dwóch plików: vimdiff

```
vimdiff /docker/compose/gitea/compose.yaml /docker/compose/gitea/compose.yaml.bak
```

Osobny tryb uruchamiany od razu z terminala, otwiera oba pliki obok siebie i podświetla różnice kolorem. Przydatny przed nadpisaniem pliku konfiguracyjnego kopią zapasową, żeby zobaczyć dokładnie, co się zmieni.

Przykład:

```
 services:
   gitea:
-    image: gitea/gitea:1.26
+    image: gitea/gitea:1.27
```

◫◫ **vimdiff** = Vim w trybie porównania: otwiera dwa albo więcej plików obok siebie i podświetla różnice między nimi

Podgląd bez edycji: less i more

Nie każde spojrzenie na plik wymaga otwierania go w edytorze. Do samego przeglądania, bez ryzyka przypadkowej zmiany, służą dwa mniejsze narzędzia.

Narzędzie	Przewijanie w obie strony	Szukanie	Kiedy sięgamy
`less`	tak, strzałki, spacja, `b`	`/wzorzec`, `n`, `N`	domyślny wybór do przeglądania logów i długich plików

Narzędzie	Przewijanie w obie strony	Szukanie	Kiedy sięgamy
`more`	tylko w przód, spacja, Enter	`/wzorzec`	starsze systemy, minimalne instalacje bez `less`

```
less /var/log/messages
```

Wewnątrz `less` samo `g` (bez podwojenia) skacze na początek pliku, `G` na koniec, podobnie jak `gg` i `G` w Vimie, a `q` naprawdę kończy podgląd, w odróżnieniu od Vima, gdzie `q` ma zupełnie inne znaczenie, opisane na stronie Makra, rejestry, bloki i inne sztuczki.

Z życia wzięte

> **Z ŻYCIA WZIĘTE**
>
> Tomek poprawia plik `compose.yaml` na `docker01` i chce jednocześnie zerknąć do pliku `.env` obok, żeby sprawdzić nazwę zmiennej. Zamiast zamykać jeden plik i otwierać drugi, wpisuje `:vsp .env`, dostaje oba pliki obok siebie i przepisuje nazwę zmiennej bez pomyłki.

Jak to działa

> **Pojęcie: bufor, okno, karta.** Bufor to zawartość jednego otwartego pliku w pamięci Vima, niezależnie od tego, czy jest akurat widoczny na ekranie. Okno to widoczny na ekranie fragment pokazujący jeden bufor, tego samego bufora może dotyczyć kilka okien naraz. Karta grupuje jedno albo więcej okien pod jedną zakładką u góry ekranu. Rozróżnienie ma znaczenie przy komendach z części Pierwsze kroki: `:q` zamyka bieżące okno, `:qa!` wszystkie okna wszystkich kart w całej sesji.

Częste błędy

Objaw	Przyczyna	Co robimy
`:e innyPlik` odmawia otwarcia, komunikat o niezapisanych zmianach	bieżący bufor ma niezapisane zmiany, Vim nie chce ich porzucić po cichu	zapisujemy `:w` przed `:e`, albo wymuszamy `:e!` świadomie porzucając zmiany

Objaw	Przyczyna	Co robimy
Nie wiemy, ile plików jest otwartych i który ma niezapisane zmiany	żaden pasek Vima domyślnie tego nie pokazuje na pierwszy rzut oka	`:ls`, szukamy znacznika `+` przy nazwie
Naciśnięcie `q` w Vimie nie kończy podglądu jak w `less`	`q` w Vimie zaczyna nagrywanie makra, nie jest komendą wyjścia	wyjście z Vima zawsze przez `:q`, `:wq` albo `:q!`, patrz Pierwsze kroki: otwarcie, zapis, wyjście

□□ Ćwiczenia

1. Pracujemy nad `compose.yaml` i musimy jednocześnie widzieć plik `.env` obok, bez zamykania pierwszego. Jakiej komendy użyjemy?

Odpowiedź

`:vsp .env` (albo `:sp .env`, jeśli wolimy podział poziomy zamiast pionowego). Otwiera drugi plik w nowym oknie tej samej sesji, oba widoczne naraz.

1. Dlaczego `q` w `less` kończy podgląd, a `q` w Vimie nie wychodzi z edytora?

Odpowiedź

To dwa różne programy z różnym znaczeniem tej samej litery. `less` to prosty program do przeglądania, gdzie `q` (quit) jest jego jedyną komendą wyjścia. Vim ma znacznie bogatszy zestaw komend w trybie normalnym, i `q` zarezerwował dla rozpoczęcia nagrywania makra, opisanego na stronie Makra, rejestry, bloki i inne sztuczki, a wyjście przypisał do komend z dwukropkiem.

Powiązane: Pierwsze kroki: otwarcie, zapis, wyjście · Makra, rejestry, bloki i inne sztuczki

Źródła: `man 1 vim`, `man 1 less`, `man 1 more` · `:help windows.txt`, `:help diff.txt` w samym Vimie

Revision #31
Created 2025-11-23 13:26:30 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:27 UTC by Tomasz Konieczny