tmux: multiplekser terminala

Poziom: średnio **Dotyczy: wszystkie dystrybucje (działa też na macOS i BSD)**

Czas: ~18 min

tmux trzyma sesję terminala żywą, kiedy Ty się rozłączysz, i dzieli ją na dowolną liczbę okien i paneli. Jedna komenda uruchomiona w tmuksie przeżyje zerwane SSH, zamknięty laptop i restart klienta terminala.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
BSD	Berkeley Software Distribution: rodzina Uniksów z uczelni Berkeley, inna gałąź niż Linux, stąd czasem inna składnia tych samych poleceń
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
Wi-Fi	Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii

Fiszka

TYLKO ODCZYT

Utworzenie nowej, anonimowej sesji: `tmux`

```
tmux
```

Uruchamia sesję bez nazwy — system nada jej numer zamiast etykiety. Dobre do jednorazowego sprawdzenia czegoś, złe jako nawyk: po tygodniu mamy kilka anonimowych sesji i żadna z nich nic nie mówi o swojej zawartości.

tmux = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

TYLKO ODCZYT

Utworzenie nazwanej sesji: `tmux new -s NazwaSesji`

```
tmux new -s wdrozenie
```

Zakładamy sesję i od razu nadajemy jej nazwę `wdrozenie` — tak warto robić zawsze. Nazwana sesja daje się rozpoznać w `tmux ls` bez podłączania się do niej.

tmux = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

TYLKO ODCZYT

Wypisanie listy działających sesji: `tmux ls`

```
tmux ls
```

Pokazujemy wszystkie sesje tmuksa uruchomione na tej maszynie, niezależnie od tego, czy ktoś jest do nich akurat podłączony.

Przykład:

```
wdrozenie: 3 windows (created Thu Sep 17 14:02:11 2026)
backup: 1 windows (created Thu Sep 17 15:40:02 2026)
```

tmux = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

TYLKO ODCZYT

Podłączenie do sesji: `tmux attach -t NazwaSesji`

```
tmux attach -t wdrozenie
```

Podłącza bieżący terminal do sesji o podanej nazwie — dokładnie w tym stanie, w jakim ją zostawiono, łącznie z historią przewijania. Skrót: `tmux a -t wdrozenie`.

tmux = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

ZMIENIA SYSTEM

Zakończenie sesji: `tmux kill-session`

```
tmux kill-session -t wdrozenie
```

Kończy sesję razem z całą jej zawartością — wszystkimi oknami, panelami i procesami działającymi w środku. To nieodwracalne: nie jest odłączeniem, tylko zabiciem. `tmux kill-server` idzie dalej i kończy od razu wszystkie sesje tmuksa na maszynie, nie tylko jedną.

tmux = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

TYLKO ODCZYT

Utworzenie sesji w tle bez podłączania się: `tmux new -d -s NazwaSesji 'komenda'`

```
tmux new -d -s backup 'rsync -av /dane/ storage01.ad.zielona.example:/kopie/'
```

Flaga `-d` zakłada sesję i od razu zostawia ją odłączoną — nikt nigdy nie widzi jej ekranu, chyba że później zrobi `tmux attach`. Komenda w cudzysłowie startuje w tej sesji natychmiast, jako pierwszy proces jej pierwszego panelu. Wygodne w skryptach, które mają coś odpalić w tle bez otwierania interaktywnego okna. Nazwę sesji można później zmienić bez jej zabijania: `tmux rename-session -t backup backup-nocny`.

tmux = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

TYLKO ODCZYT

Wpisanie komend z zewnątrz sesji: `tmux send-keys`

```
tmux send-keys -t deploy 'cd /opt/app && git pull' Enter
```

Wpisujemy tekst do panelu wskazanej sesji tak, jakby ktoś wystukał go na klawiaturze. `Enter` jest osobnym argumentem — bez niego tekst wyląduje w linii poleceń, ale nikt go nie zatwierdzi. Używane w skryptach, które muszą coś odpalić w konkretnej, już istniejącej sesji, bez ręcznego podłączania się do niej.

tmux = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

TYLKO ODCZYT

Wysłanie jednej komendy na kilka paneli naraz: `setw synchronize-panes`

```
prefiks, :
setw synchronize-panes on
```

Po włączeniu wszystko, co wpiszemy w jednym panelu bieżącego okna, trafia jednocześnie do wszystkich pozostałych paneli tego okna. Rozstawiamy cztery panele, w każdym łączymy się przez `ssh` z innym serwerem (`srv01`–`srv04`), włączamy synchronizację i wpisujemy komendę raz — wykona się na czterech maszynach naraz. Wyłączamy ją zaraz po użyciu (`setw synchronize-panes off`), bo inaczej kolejna literówka trafi na wszystkie panele jednocześnie.

> **Pojęcie: prefiks.** Klawisz, który zapowiada tmuksowi „to, co zaraz naciśniemy, to skrót tmuksa, nie zwykły znak do wpisania w powłoce". Domyślnie to `Ctrl+b`: wciskamy go, puszczamy, i dopiero potem osobno wciskamy właściwą literę. Bez tego dwuetapowego rytuału każdy skrót z tabeli niżej trafiłby po prostu do powłoki jako zwykły tekst.

TYLKO ODCZYT

Skróty po prefiksie

Skrót (po prefiksie)	Co robi	Kiedy używasz
`c`	nowe okno	zaczynasz kolejne zadanie w tej samej sesji
`,`	zmień nazwę bieżącego okna	masz pięć okien i już nie pamiętasz, które jest które
`n` / `p`	następne / poprzednie okno	przełączasz się liniowo
`0`–`9`	skok do okna o danym numerze	wiesz dokładnie, gdzie chcesz być
`w`	interaktywna lista okien	zgubiłeś się w numeracji
`%`	podziel panel w pionie (obok)	chcesz dwie kolumny obok siebie
`"`	podziel panel w poziomie (pod spodem)	chcesz drugi panel pod bieżącym
strzałki	przełącz panel w danym kierunku	masz więcej niż dwa panele w oknie
`o`	skocz do kolejnego panelu po kolei	nie chce Ci się celować strzałką

Skrót (po prefiksie)	Co robi	Kiedy używasz
`z`	zoom bieżącego panelu na cały ekran (i z powrotem)	panel jest za mały, żeby coś w nim przeczytać
`x`	zamknij bieżący panel (z potwierdzeniem)	kończysz pracę w jednym z paneli
`{` / `}`	zamień panel miejscami z sąsiednim	układ wyszedł w złej kolejności
`d`	odłącz się od sesji	kończysz na dziś, proces ma dalej działać
`[`	wejdź w tryb kopiowania	chcesz przewinąć historię albo skopiować tekst
`?`	lista wszystkich skrótów	zapomniałeś, co robi dana kombinacja
`:`	wiersz poleceń tmuksa	komenda, dla której nie ma skrótu

◫◫ ZMIENIA SYSTEM

Plik `~/.tmux.conf`

```
cat > ~/.tmux.conf <<'EOF'
# zmiana prefiksu z Ctrl+b na Ctrl+a — wygodniejsze, bo Ctrl+b zderza się
# z niektórymi skrótami w edytorach i pagerach
unbind C-b
set -g prefix C-a
bind C-a send-prefix

# mysz: klik przełącza panel, przeciąganie zmienia rozmiar, kółko przewija
set -g mouse on

# 50000 linii historii na panel zamiast domyślnych 2000
set -g history-limit 50000

# okna i panele numerowane od 1, nie od 0 — 0 leży daleko od 1 na klawiaturze
set -g base-index 1
setw -g pane-base-index 1

# przeładuj konfigurację bez zamykania sesji
bind r source-file ~/.tmux.conf \; display "Config przeładowany"
```

```
# czytelny pasek statusu: nazwa sesji po lewej, data i godzina po prawej
set -g status-bg colour235
set -g status-fg colour250
set -g status-left '#[bold]#S '
set -g status-right '%Y-%m-%d %H:%M'
EOF
tmux source-file ~/.tmux.conf
```

Plik czytany przy starcie każdej nowej sesji tmuksa. Zmiany nie działają wstecz na sesje już uruchomione, dopóki nie przeładujemy configu poleceniem `tmux source-file ~/.tmux.conf` albo skrótem `prefiks, r` — ten skrót zadziała dopiero po pierwszym ręcznym przeładowaniu, bo do tego momentu sesja jeszcze nie zna linijki `bind r ...`.

Uważaj

- Wersja tmux w chwili pisania to 3.7c (sierpień 2026), a domyślny prefiks `Ctrl+b` nie zmienił się od lat — źródło w sekcji Źródła.
- `exit` albo `Ctrl+d` w ostatnim panelu **zamyka** okno (i sesję, jeśli było ostatnie) — to nie to samo co odłączenie przez `d` po prefiksie.
- Domyślny limit historii to tylko 2000 linii na panel — bez wpisu w `~/.tmux.conf` stracisz starszy output.

cat = concatenate: sklej i wypisz pliki · **tmux** = terminal multiplexer: menedżer wielu okien terminala w jednej sesji, przeżywa rozłączenie SSH

Z życia wzięte

Z ŻYCIA WZIĘTE

O 21:30, w ostatni dzień zamknięcia miesiąca, Ola rozstawia w tmuksie cztery panele podłączone przez SSH do sql01, pg01, my01 i ora01. Włącza `synchronize-panes`, wpisuje jedną komendę sprawdzającą miejsce na dysku i widzi wynik z czterech baz naraz zamiast łączyć się z każdą osobno. Po skończonej kontroli pamięta wyłączyć synchronizację, zanim wpisze cokolwiek innego.

Po co to istnieje

Łączysz się przez SSH, odpalasz migrację bazy danych na 40 minut, i tracisz Wi-Fi. Razem z połączeniem SSH kończy się cała sesja terminala — powłoka dostaje `SIGHUP` i przekazuje go dalej (mechanizm opisany na stronie *Zadania powłoki: jobs, bg, fg, nohup, disown*), a migracja ginie w połowie transakcji.

tmux rozwiązuje to inaczej niż samo `nohup` czy `disown` — nie tylko chroni jeden proces przed sygnałem rozłączenia, ale utrzymuje **cały terminal**, ze wszystkim co na nim widać, jako osobny proces działający niezależnie od Twojego połączenia. Odłączasz się — proces dalej pracuje, ekran dalej się aktualizuje, tylko akurat nikt nie patrzy. Wracasz później, nawet z innego komputera, i podłączasz się dokładnie tam, gdzie skończyłeś, łącznie z historią przewijania.

Drugi problem: praca na kilku rzeczach naraz bez otwierania wielu okien terminala klienta i bez pamiętania, które SSH prowadzi do którego serwera. Jedna sesja mieści dziesiątki okien i paneli, a przełączasz się między nimi jednym skrótem zamiast Alt-Tabem po oknach systemu.

Analogia

Wyobraź sobie **pokój z kilkoma biurkami, a na każdym biurku po kilka monitorów**.

Sesja to pokój. Możesz mieć ich kilka naraz — jeden do wdrożenia na jednym serwerze, drugi do monitoringu innego — i wchodzić do dowolnego, kiedy chcesz; reszta pokoi trwa bez Ciebie. **Okno** to jedno biurko w tym pokoju — masz ich tyle, ile zadań: jedno na edycję configu, drugie na logi, trzecie na monitoring. Widzisz tylko biurko, przy którym akurat siedzisz, pozostałe czekają obok. **Panel** to monitor na biurku — możesz mieć ich kilka naraz w polu widzenia, każdy pokazujący coś innego, wszystkie widoczne bez przełączania.

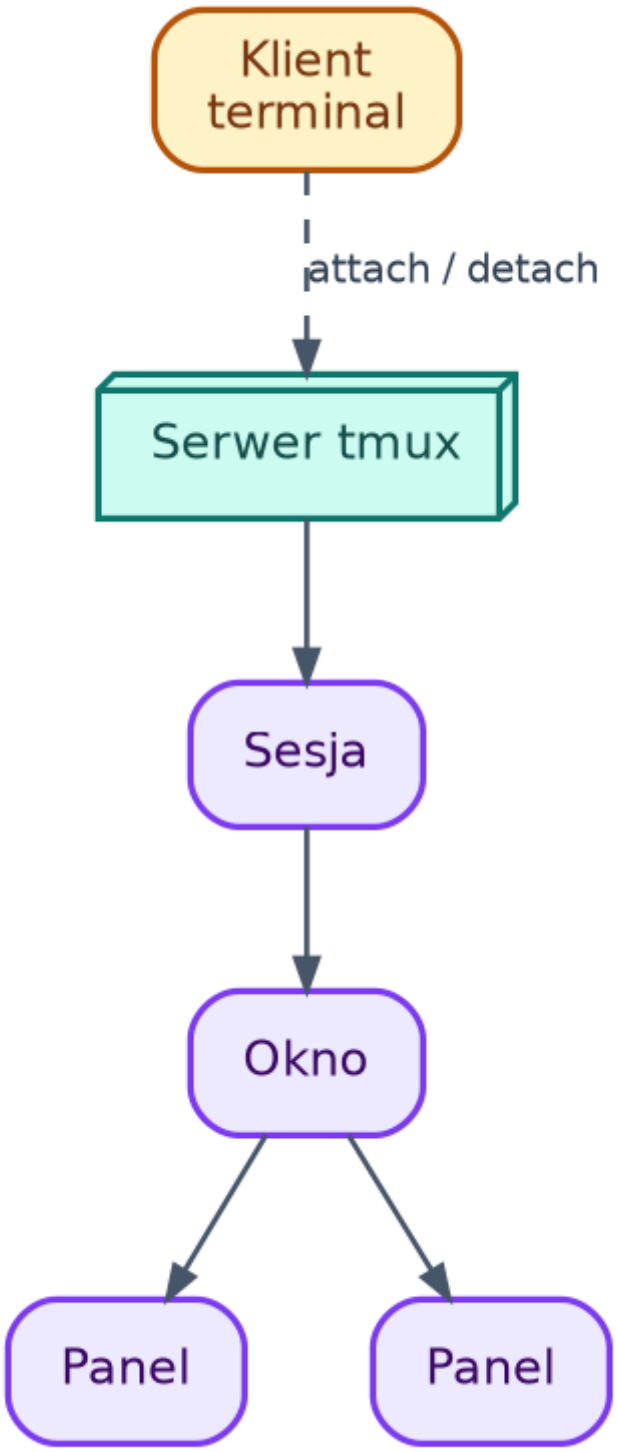
To realna przewaga nad `screen`, który zna tylko sesję i okno. Chcesz zobaczyć dwie rzeczy naraz w screenie? Musisz otworzyć drugie okno terminala klienta i podłączyć się do tej samej sesji drugi raz. W tmuksie dzielisz jedno biurko na dwa monitory jednym skrótem, bez drugiego połączenia SSH.

Ograniczenie tej analogii jest realne: **na biurku widzisz tylko to jedno biurko** — nie da się jednym spojrzeniem ogarnąć dwóch różnych biurek (okien) w tym samym pokoju tylko dlatego, że są w tej samej sesji. Żeby zobaczyć drugie, trzeba się do niego fizycznie przesiąść (`n`, `p` albo numer). To rozróżnienie — panel dzieli ekran, okno go zastępuje — myli początkujących najczęściej.

Jak to działa

Rozwiń: jak to działa pod spodem

tmux działa w modelu klient-serwer, i to jest klucz do zrozumienia, dlaczego odłączanie w ogóle działa. Kiedy wpisujesz `tmux`, uruchamia się (jeśli jeszcze nie działa) proces `tmux` w tle jako **serwer** — to on trzyma stan wszystkich sesji, okien i paneli oraz same procesy powłoki działające w każdym z nich. Terminal, który widzisz, to tylko **klient** podłączony do tego serwera przez gniazdo Uniksowe w `/tmp/tmux-<uid>/`. Kiedy się odłączasz (`d` po prefiksie) albo tracisz połączenie SSH, ginie wyłącznie klient. Serwer żyje dalej — nigdy nie był potomkiem Twojej powłoki SSH i nigdy nie dostał jej `SIGHUP`.



Serwer tmux trzyma sesje, każda sesja mieści okna, a każde okno panele; klient tylko podłącza się do serwera i może odłączyć się bez wpływu na procesy w środku (kliknięcie otwiera pełny rozmiar)

Dokładnie dlatego `tmux new -d -s backup 'komenda'` z Fiszki działa bez podłączania się w ogóle: tworzy sesję na serwerze i od razu ją zostawia, bez otwierania klienta.

Struktura wewnątrz jednej sesji to drzewo: sesja zawiera dowolną liczbę okien, każde okno dowolną liczbę paneli, każdy panel to osobny pseudoterminal (PTY) z własnym procesem powłoki na końcu. Podział panelu (`%` albo `"`) nie kopiuje niczego — tworzy nowy PTY obok istniejącego, dzieląc dostępną przestrzeń ekranu.

tmux kontra screen — skąd bierze się przewaga. `screen` (1987) zna tylko sesję i okno — podział ekranu jest w nim doklejonym dodatkiem, nie pełnoprawnym poziomem hierarchii. tmux od pierwszego wydania w 2007 roku miał od razu trzy poziomy i myślał o skryptowalności (`send-keys`, format stringów). Pełne zestawienie i migrację ze screena opisuje strona *screen — sesje, które przeżyją rozłączenie*; tu wystarczy skrót:

Cecha	tmux	screen
Poziomy struktury	sesja → okno → panel	sesja → okno (podział ekranu to dodatek)
Pasek statusu	konfigurowalny, włączony domyślnie	trzeba włączyć i skonfigurować ręcznie
Format konfiguracji	jeden spójny `~/.tmux.conf`	starszy, mniej spójny składniowo
Sterowanie z zewnątrz (`send-keys`)	pełnoprawna część API	dostępne, mniej naturalne
Aktywny rozwój	regularne wydania (patrz Źródła)	sporadyczny

Krok po kroku

1. Załóż nazwaną sesję.

```
tmux new -s wdrozenie
```

Zawsze z `-s nazwa`. Sesja anonimowa (samo `tmux`) dostaje numer zamiast nazwy — po tygodniu, kiedy masz ich sześć, `tmux ls` pokaże `0: 1 windows`, `3: 2 windows` bez żadnej wskazówki, co w której siedzi.

2. Otwórz kolejne okno, nazwij je, przełącz się między nimi.

```
prefiks, c        # nowe okno
prefiks, ,        # nazwij bieżące okno
prefiks, 0        # wróć do okna numer 0
```

```
[wdrozenie] 0:bash* 1:logi- 2:monitor
```

Gwiazdka oznacza okno aktywne, myślnik — ostatnio odwiedzone.

3. Podziel okno na panele i poruszaj się między nimi.

```
prefiks, %         # podział pionowy — dwie kolumny obok siebie
prefiks, "         # podział poziomy — drugi panel pod bieżącym
prefiks, strzałka  # skok do sąsiedniego panelu
prefiks, z         # zoom panelu na cały ekran; jeszcze raz — powrót
```

`z` przydaje się, gdy panel jest za wąski, żeby przeczytać długą linię logu.

4. Odłącz się i podłącz z powrotem.

```
prefiks, d
```

```
tmux ls
tmux attach -t wdrozenie
```

```
wdrozenie: 3 windows (created Thu Sep 17 14:02:11 2026)
```

5. Zabij sesję, kiedy zadanie się skończyło.

```
tmux kill-session -t wdrozenie
```

6. Skonfiguruj `~/.tmux.conf` i przeładuj go bez restartu serwera tmuksa.

Treść pliku i komenda przeładowania są w Fiszce, w karcie „Plik `~/.tmux.conf`". Pierwsze przeładowanie po dodaniu `bind r` musisz zrobić ręcznie — dopiero od tego momentu `prefiks, r` działa jako skrót do kolejnych przeładowań. Jeśli zmieniasz przy okazji prefiks, kolejne przeładowanie robisz już nowym.

7. Wejdź w tryb kopiowania i wyciągnij tekst.

```
prefiks, [        # tryb kopiowania — strzałki/PageUp zaczynają przewijać
spacja            # zaznacz początek fragmentu (przy mode-keys vi)
Enter             # skopiuj zaznaczenie do bufora tmuksa
prefiks, ]        # wklej z bufora
```

Domyślnie tmux ma własny, wewnętrzny schowek, niezależny od schowka X11/Wayland. Żeby kopiowanie trafiało też do schowka systemowego, dopisz w `~/.tmux.conf` (wymaga `xclip` na X11 albo `wl-clipboard` na Wayland):

```
bind-key -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel "xclip -in -selection
clipboard"
```

8. Zautomatyzuj wpisywanie komend z zewnątrz sesji przez `send-keys`.

```
tmux new -d -s deploy
tmux send-keys -t deploy 'cd /opt/app && git pull' Enter
tmux send-keys -t deploy './deploy.sh --env=prod' Enter
```

`send-keys` wpisuje tekst do panelu tak, jakbyś sam go wystukał na klawiaturze — łącznie z `Enter` jako osobnym argumentem, bo bez niego komenda tylko wyląduje w linii poleceń, nie zostanie wykonana. Przydaje się w skryptach, które muszą coś odpalić w konkretnej, już istniejącej sesji.

9. Synchronizuj panele, żeby wpisywać tę samą komendę na kilku serwerach naraz.

```
prefiks, :
setw synchronize-panes on
```

Od tej chwili wszystko, co wpiszesz w jednym panelu, trafia do wszystkich paneli tego okna. Rozstaw cztery panele, w każdym `ssh` do innego serwera (`srv01`–`srv04`), włącz synchronizację i wpisz `sudo dnf update -y` raz — wykona się na czterech maszynach naraz. Wyłącz zaraz po użyciu (`setw synchronize-panes off`), bo inaczej kolejna literówka trafi na wszystkie cztery serwery jednocześnie.

Częste błędy

Nacisnąłeś `Ctrl+d` zamiast prefiksu i litery `d`, licząc na odłączenie.

```
[wdrozenie] logout
[proces zakończony]
```

`Ctrl+d` to sygnał końca pliku dla powłoki, nie skrót tmuksa. Zamknąłeś panel — a jeśli to był ostatni panel ostatniego okna, razem z nim całą sesję. Odłączenie to zawsze dwa oddzielne naciśnięcia: prefiks, puść, potem `d`.

Zapomniałeś nazwać sesję i po tygodniu masz sześć anonimowych.

```
$ tmux ls
0: 1 windows (created Mon Sep 14 09:12:03 2026)
1: 3 windows (created Tue Sep 15 11:40:21 2026)
2: 1 windows (created Wed Sep 16 08:03:55 2026)
```

Żadna nazwa nic nie mówi o zawartości — musisz podłączyć się do każdej po kolei, żeby sprawdzić, co w której siedzi. Zawsze `tmux new -s coś-konkretnego`.

Edytowałeś `~/.tmux.conf`, zapisałeś, i nic się nie zmieniło.

Zmiany w pliku konfiguracyjnym nie działają wstecz na już uruchomioną sesję, dopóki jej nie przeładujesz. Bez wpisu `bind r source-file ~/.tmux.conf` w configu musisz za każdym razem pamiętać `tmux source-file ~/.tmux.conf` albo `prefiks, :` i wpisać to ręcznie. Zapomnisz raz na kilka tygodni i będziesz się zastanawiał, dlaczego mysz albo nowy prefiks dalej nie działają.

Zagnieżdżony tmux przez SSH — prefiks trafia do złej sesji.

Jesteś w tmuksie lokalnie i przez SSH podłączasz się do serwera, na którym też odpalasz tmuksa. Naciskasz `Ctrl+b` — łapie go **lokalna** sesja, nie ta na serwerze, bo prefiks przechwytuje pierwszy klient w łańcuchu. Rozwiązanie: naciśnij prefiks dwa razy (`Ctrl+b Ctrl+b litera`) albo ustaw w lokalnym configu inny prefiks niż na serwerach zdalnych.

□□ Ćwiczenia

> Wszystko poniżej wykonujesz na własnej maszynie, w katalogu `/tmp`. Nic nie wymaga uprawnień administratora.

Ćwiczenie 1 — sesja, podział okna, odłączenie

```
tmux new -s cwiczenie
echo $$
```

Podziel okno (`prefiks, %`) i w drugim panelu uruchom coś, co działa w pętli:

```
watch -n 1 date
```

Wróć do pierwszego panelu (`prefiks, o`), odłącz się (`prefiks, d`) i z zewnątrz sprawdź, że `watch` dalej działa:

```
ps -ef | grep watch
```

Posprzątaj: `tmux kill-session -t cwiczenie`.

Ćwiczenie 2 — sesja w tle bez podłączania się w ogóle

```
tmux new -d -s licznik 'for i in $(seq 1 300); do echo "tick $i"; sleep 1; done'
tmux attach -t licznik
```

Odłącz się (`prefiks, d`), odczekaj chwilę i podłącz się ponownie: `sleep 5 && tmux attach -t licznik`
.

Czego się spodziewać

Licznik będzie o kilka sekund dalej niż w momencie odłączenia — proces działał cały czas, niezależnie od tego, czy ktoś patrzył. To sedno różnicy między tmuksem a zwykłym uruchomieniem komendy: obserwator jest opcjonalny, proces nie. Na koniec: `tmux kill-session -t licznik`.

Pytania kontrolne

1. Dlaczego zamknięcie okna terminala klienta nie kończy procesów w sesji tmuksa, a `exit` wpisany w ostatnim panelu — kończy?

Odpowiedź

Zamknięcie okna klienta zrywa tylko połączenie z serwerem tmuksa, dokładnie tak jak zerwane SSH — serwer i procesy w środku żyją dalej. `exit` w ostatnim panelu kończy powłokę działającą w tym panelu; skoro to był ostatni panel ostatniego okna, kończy się całe okno, a wraz z nim cała sesja po stronie serwera. Różnica: kto inicjuje zakończenie — sam serwer (przy `exit`) czy tylko klient, który się od niego odpina.

2. Czym różni się okno od panelu i dlaczego to rozróżnienie ma znaczenie przy planowaniu układu pracy?

Odpowiedź

Okno to osobna „strona" w sesji — widoczna tylko jedna naraz. Panel to podział jednego okna — wszystkie panele widoczne równocześnie. Rzeczy, które chcesz obserwować jednocześnie (log usługi i powłoka do testów), powinny być panelami tego samego okna. Rzeczy niezwiązane,

które i tak przełączasz pojedynczo, lepiej trzymać jako osobne okna.

3. Masz cztery panele podłączone do czterech różnych serwerów i włączone `synchronize-panes`. Wpisujesz komendę z literówką. Co się stanie i jak to naprawić na przyszłość?

Odpowiedź

Literówka trafi na wszystkie cztery serwery jednocześnie — synchronizacja nie odróżnia poprawnej komendy od błędnej, tylko powiela każde naciśnięcie klawisza. Na przyszłość: włączaj ją tuż przed wpisaniem właściwej komendy i wyłączaj zaraz po (`setw synchronize-panes off`).

4. `bind r source-file ~/.tmux.conf` w samym `~/.tmux.conf` działa jako skrót dopiero po pewnym warunku. Jakim?

Odpowiedź

To instrukcja wewnątrz pliku konfiguracyjnego — żeby zaczęła obowiązywać, sesja musi najpierw wczytać nową wersję tego pliku. Pierwsze przeładowanie po dodaniu tej linijki robisz więc ręcznie (`tmux source-file ~/.tmux.conf`); dopiero potem `prefiks, r` działa jako skrót do kolejnych przeładowań.

Powiązane: screen: sesje, które przeżyją rozłączenie · Zadania powłoki: jobs, bg, fg, nohup, disown · Sygnały: kill, pkill, killall, trap · Terminal, TTY, konsola, powłoka

Źródła: `man 1 tmux` · tmux/tmux — Releases (GitHub) · tmux/tmux — plik CHANGES

Revision #19
Created 2026-09-17 15:07:22 UTC by Tomasz Konieczny
Updated 2026-10-03 19:08:20 UTC by Tomasz Konieczny