Pagery: less i more

Poziom: podstawy **Dotyczy: wszystkie dystrybucje** **Czas: ~10 min**

Pager to program, który pokazuje plik ekran po ekranie i pozwala się po nim poruszać. `less` *jest tym, którego używasz codziennie — również wtedy, gdy o tym nie wiesz, bo to on wyświetla strony* `man`*.*

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
ANSI	American National Standards Institute: tu chodzi o kody sterujące kolorem i pozycją kursora w terminalu
CSV	w Windowsie: Cluster Shared Volume, dysk widoczny naraz dla wszystkich węzłów klastra; w danych: Comma-Separated Values, plik z wartościami oddzielonymi przecinkami
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
POSIX	Portable Operating System Interface: wspólny zestaw zasad, dzięki któremu polecenie działa tak samo na różnych Uniksach
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa

Fiszka

TYLKO ODCZYT

Otwarcie pliku w pagerze

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

Plik ląduje w pagerze zamiast lecieć całością na ekran. `spacja` przewija dalej, `q` wychodzi.

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Numerowanie linii: -N

```
less -N plik.log
```

Włącza numerację linii w lewej kolumnie. Można ją też przełączyć w locie, wpisując `-N` i Enter wewnątrz pagera, bez zamykania i ponownego otwierania pliku.

Przykład:

```
     1 2026-09-17 08:14:02 INFO  uruchomienie usługi
     2 2026-09-17 08:14:03 INFO  nasłuch na porcie 8080
     3 2026-09-17 08:14:03 INFO  połączenie z bazą danych OK
```

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Wyłączenie zawijania długich linii: -S

```
less -S plik.csv
```

Wyłącza zawijanie: przewijamy w bok strzałkami zamiast oglądać jeden rekord rozbity na kilka wierszy ekranu. Bez `-S` jeden szeroki wiersz CSV zajmuje kilka linii terminala i tabela przestaje być czytelna.

CSV = w Windowsie: Cluster Shared Volume, dysk widoczny naraz dla wszystkich węzłów klastra; w danych: Comma-Separated Values, plik z wartościami oddzielonymi przecinkami · **less** = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Interpretacja kolorów ANSI: -R

```
less -R kolorowy.log
```

Nakazuje interpretować kody kolorów ANSI zamiast wyświetlać je dosłownie jako `ESC[01;31m`. Obowiązkowe przy oglądaniu wyjścia komend uruchomionych z `--color=always`.

ANSI = American National Standards Institute: tu chodzi o kody sterujące kolorem i pozycją kursora w terminalu · **less** = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Pozostawienie treści na ekranie po wyjściu: -X

```
less -X plik.log
```

Wyłącza wysyłanie sekwencji inicjujących i kończących „ekran alternatywny", więc ostatni widoczny ekran zostaje w terminalu po zamknięciu `less`, zamiast zniknąć. Cena: przy kilkunastu wywołaniach historia terminala zapełnia się zrzutami, a w części emulatorów `-X` psuje obsługę myszy i kółka.

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Wyszukiwanie bez rozróżniania wielkości liter: -i

```
less -i plik.log
```

Wyszukiwanie przez `/` i `?` przestaje rozróżniać wielkość liter. Da się też włączyć w locie, wpisując `-i` i Enter wewnątrz pagera.

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Otwarcie od razu na końcu pliku: +G

```
less +G /var/log/messages
```

Wszystko po znaku `+` to polecenie wykonane od razu po otwarciu — tu: skok na koniec, dokładnie ten efekt, co ręczne wciśnięcie `G` po otwarciu pliku.

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Otwarcie z automatycznym wyszukiwaniem: +/wzorzec

```
less +/'Segmentation fault' core.log
```

Otwiera plik i natychmiast wyszukuje podany wzorzec, ładując na pierwszym trafieniu. Wygodne, gdy wiemy, czego szukamy, zanim jeszcze otworzymy plik.

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Otwarcie od razu w trybie podążania: +F

```
less +F /var/log/messages
```

Otwiera plik od razu w trybie śledzenia na żywo — tak, jakbyśmy zaraz po otwarciu ręcznie wcisnęli `F`. Pełne porównanie z `tail -f` jest na stronie o obserwacji logów na żywo.

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Pokazanie kolorów z grep w potoku: -R

```
grep --color=always -i 'error' /var/log/messages | less -R
```

`--color=always` wymusza kolorowanie, mimo że wyjście `grep` idzie do potoku, a nie na terminal — domyślne `auto` by je wyłączyło. `-R` każe `less` te kolory pokazać, zamiast wypisać surowe kody sterujące.

grep = global regular expression print: szukaj wzorca w tekście

TYLKO ODCZYT

Otwarcie wielu plików naraz

```
less /var/log/nginx/*.log
```

Otwieramy wszystkie pasujące pliki w jednym pagerze. Przełączanie między nimi opisuje tabela komend niżej (`:n`, `:p`, `:x`).

less = przeglądarka tekstu z przewijaniem (nazwa to żart: less is more)

TYLKO ODCZYT

Przeglądanie plików skompresowanych: zless

```
zless /var/log/messages-20260910.gz
```

Rozpakowuje archiwum `gzip` w locie i pokazuje zawartość w zwykłym `less`, bez tworzenia rozpakowanej kopii na dysku.

◫◫ ZMIENIA SYSTEM

Ustawienie domyślnych opcji na stałe: zmienna LESS

```
export LESS='-R -i -M -J'
export PAGER=less
export MANPAGER='less -R'
```

`LESS` jest doklejane do każdego wywołania `less`, więc nie trzeba pamiętać ulubionego zestawu flag. `PAGER` respektują `git`, `systemctl`, `journalctl`, `psql` i wiele innych narzędzi; `MANPAGER` ma pierwszeństwo dla `man`.

◫◫ **export** = ustaw zmienną środowiskową widoczną dla uruchamianych programów

◫◫ TYLKO ODCZYT

Skróty klawiszowe w środku pagera

Klawisz	Co robi
`spacja` / `b`	ekran w przód / w tył
`d` / `u`	pół ekranu w przód / w tył
`j` / `k`, strzałki	linia w przód / w tył
`g` / `G`	początek / koniec pliku
`50g`	skok do linii 50
`/wzorzec` / `?wzorzec`	szukaj w przód / wstecz
`n` / `N`	następne / poprzednie trafienie
`&wzorzec`	pokaż **tylko** linie pasujące (`&` + Enter kasuje filtr)
`&!wzorzec`	pokaż tylko linie **nie**pasujące
`F`	podążaj za plikiem; `Ctrl+C` przerywa
`v`	otwórz bieżący plik w `$EDITOR`

Klawisz	Co robi
`Ctrl+G`	nazwa pliku, pozycja, rozmiar
`-S` / `-N`	przełącz opcję bez wychodzenia z pagera
`:n` / `:p` / `:x`	następny / poprzedni / pierwszy plik (przy kilku otwartych)
`h` / `q`	pomoc / wyjście

Uważaj

- Bez `-R` kolory z `grep` czy `ls` wyświetlą się jako `ESC[01;31m` — śmieci w środku każdej linii.
- `less` na pliku binarnym zapyta „See it anyway?"; potwierdzenie rozsypie terminal, ratunek to `reset`.
- `export LESS=-F` w `.bashrc` sprawi, że krótkie strony `man` mrugną i znikną — `-F` to „zakończ, jeśli mieści się na ekranie".

Z życia wzięte

□□ Z ŻYCIA WZIĘTE

Czwartek, 12:20. Tomek na `syb01` otwiera 6-gigabajtowy zrzut archiwum przez `less` i wystraszony pyta, dlaczego Vimowi zajęłoby to minutę, a `less` otwiera się natychmiast. Ola tłumaczy, że `less` nie wczytuje pliku do pamięci, tylko doczytuje fragmenty i pamięta offsety, do których wraca przez `lseek()`. Tomek przestaje bać się otwierania dużych plików i zaczyna bać się tylko `dmesg | less`, bo tam trzeba buforować wszystko w pamięci.

Po co to istnieje

`cat /var/log/messages` na pliku z dwustoma tysiącami linii przewinie terminal w kilkanaście sekund i zostawi Cię z ostatnim ekranem. Reszta jest teoretycznie w buforze przewijania emulatora — tylko że bufor ma domyślnie kilka tysięcy linii, nie da się go przeszukiwać poleceniem ani skoczyć w nim do linii 48 000, a w konsoli tekstowej nie ma go w ogóle.

Pojęcie: pager. Program, który pokazuje długi tekst ekran po ekranie, zamiast wypisać go od razu w całości na standardowe wyjście, i pozwala się po nim poruszać: przewijać w obie

> strony, szukać wzorca, skoczyć do wybranej linii. W odróżnieniu od `cat`, które tylko przepycha bajty dalej, pager zna pojęcie „aktualnie widoczny ekran" i reaguje na klawisze w czasie rzeczywistym.

Pager rozwiązuje dokładnie ten problem: **zatrzymuje wypisywanie na wysokości jednego ekranu i oddaje Ci sterowanie**. Ty decydujesz, kiedy iść dalej, cofnąć się i czego szukać.

Są dwa pagery, które spotkasz. `more` jest starszy i prymitywny. `less` to jego następca, nazwany żartem z powiedzenia „less is more" — i rzeczywiście robi więcej. W codziennej pracy używasz `less`; `more` znasz po to, żeby sobie poradzić, kiedy `less` nie ma.

Analogia

Pomyśl o **czytniku mikrofilmów w czytelni archiwum**.

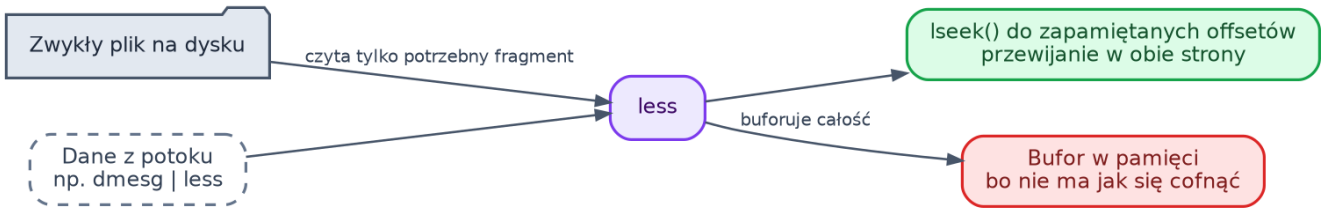
Rocznik gazety z 1958 roku jest na szpuli. Nie rozwijasz jej po podłodze — wkładasz do czytnika, który pokazuje jedną stronę naraz w okienku. Kręcisz korbą w przód, kręcisz w tył, zatrzymujesz się tam, gdzie coś Cię zainteresowało. Materiału są kilometry, a na biurku zajmuje tyle co pudełko. Tak działa `less`: plik może mieć 40 GB, a pamięci zajmie tyle, ile aktualnie wyświetla plus indeks pozycji.

`more` to ten sam czytnik, ale z korbą, która kręci tylko w jedną stronę. Przejechałeś obok interesującej strony? Wyjmujesz szpulę i zaczynasz od początku.

Analogia niesie ograniczenie, które tłumaczy realne zachowanie `less`. Operator nie wie z góry, ile stron jest na szpuli — dowie się, dopiero gdy przewinie ją do końca. Dlatego `less` na zwykłym pliku pokazuje procent postępu (zna rozmiar z systemu plików), ale ten sam `less` czytający z potoku pokazuje w pasku stanu tylko pozycję w bajtach: nie ma pojęcia, ile jeszcze przyjdzie. Z tego samego powodu `G` na wielkim pliku potrafi chwilę potrwać — czytnik musi faktycznie przewinąć szpulę do końca.

Jak to działa

`less` **nie wczytuje pliku do pamięci**. Otwiera go, czyta tyle, ile potrzebuje do zapełnienia ekranu, i zapamiętuje offsety początków linii, przez które przeszedł. Kiedy przewijasz w przód, doczytuje kolejne fragmenty. Kiedy cofasz się, używa zapamiętanych offsetów i `lseek()`, żeby wrócić do właściwego miejsca. Dlatego otwarcie pliku 40 GB jest natychmiastowe, a zużycie pamięci nie zależy od jego rozmiaru.



less czyta plik fragmentami przez lseek, a dane z potoku musi buforować w pamięci (kliknięcie otwiera pełny rozmiar)

Ma to dwie widoczne konsekwencje:

Numery linii nie są darmowe. Plik ich nie ma — są tylko bajty i znaki nowej linii. Żeby pokazać, że jesteś w linii 48 217, `less` musi policzyć wszystkie znaki nowej linii od początku. Na bardzo dużym pliku `-N` albo `-M` potrafi zauważalnie spowolnić skok na koniec.

Dane z potoku są jednorazowe. Przy `dmesg | less` pager musi buforować w pamięci to, co przeszło, bo w potoku nie ma jak się cofnąć. Wniosek: `less plik` zawsze, gdy dane są w pliku; potok tylko wtedy, gdy naprawdę filtrujesz.

`more` z pakietu `util-linux` (ten na RHEL, Debianie i openSUSE) umie cofnąć się o ekran klawiszem `b`, ale wyłącznie na zwykłym pliku. Wersja z `busybox` nie umie nawet tego. Spotkasz go tam, gdzie nikt nie instalował nic ponad minimum: obrazy kontenerów typu `debian:stable-slim`, initrd, awaryjna powłoka `dracut`, systemy wbudowane. `more` jest wymagany przez POSIX, `less` nie — stąd ta asymetria.

Krok po kroku

1. Otwórz i rozejrzyj się.

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

`spacja` przewija ekran, `b` cofa, `G` skacze na koniec, `g` na początek, `q` wychodzi. `Ctrl+G` mówi, gdzie jesteś:

```
/var/log/messages (file 1 of 1) lines 412-460/28104 byte 39821/2412887 1%  (press RETURN)
```

2. Szukaj.

`/timeout` szuka w przód, `?timeout` wstecz, `n` idzie do następnego trafienia, `N` do poprzedniego. Wzorzec to wyrażenie regularne, więc `/error.*504` też zadziała; podświetlenie kasujesz przez `ESC`, a potem `u`. Domyślnie wyszukiwanie rozróżnia wielkość liter — zmienia to `less -i plik` albo wpisane w pagerze `-i` i Enter.

3. Pokaż tylko to, co pasuje.

Wciśnij `&`, wpisz wzorzec, Enter:

```
&timeout
```

Ekran pokazuje **wyłącznie** linie zawierające `timeout` — reszta jest ukryta, ale nadal jest w pliku. `&!debug` robi odwrotnie. Filtr zdejmujesz, wciskając `&` i Enter z pustym wzorcem. Przewaga nad `grep wzorzec plik | less`: filtr możesz w dowolnej chwili zdjąć i zobaczyć kontekst wokół trafienia, bez ponownego otwierania pliku.

4. Szerokie pliki — wyłącz zawijanie.

```
less -S /tmp/eksport.csv
```

Linie nie są zawijane; strzałka w prawo i w lewo przewija widok w bok. Bez `-S` jeden rekord CSV zajmuje cztery wiersze ekranu i tabela przestaje być czytelna. Opcję możesz przełączać w trakcie: wpisz `-S` i Enter.

5. Kolory.

```
grep --color=always -iE 'error|fail' /var/log/messages | less -R
```

`--color=always`, bo w potoku `auto` wyłącza kolory. `-R` mówi `less`, żeby przepuszczał sekwencje ANSI do terminala, zamiast pokazywać je jako tekst.

6. Kilka plików naraz, edycja bez zamykania pagera.

```
less /var/log/nginx/access.log /var/log/nginx/error.log
```

`:n` przechodzi do następnego pliku, `:p` do poprzedniego, `:x` wraca do pierwszego, `:e inny.log` dokłada kolejny; pasek stanu pokazuje `(file 1 of 2)`. Wciśnięcie `v` otwiera bieżący plik w `$EDITOR` mniej więcej w miejscu, w którym byłeś — po wyjściu z edytora wracasz do `less`.

7. Ustaw sobie domyślne opcje raz.

Zmienna `LESS` jest doklejana do każdego wywołania:

```
# ~/.bashrc
export LESS='-R -i -M -J'
```

`-R` kolory, `-i` wyszukiwanie bez rozróżniania wielkości liter, `-M` rozbudowany pasek stanu, `-J` kolumna z zaznaczeniem trafień na lewym marginesie.

8. Pliki skompresowane i nietekstowe.

```
zless /var/log/messages-20260910.gz
```

`zless` (z pakietu `gzip`) rozpakowuje w locie. Ogólniejszy mechanizm to `lesspipe`: filtr wstępny uruchamiany przez `less` przed wyświetleniem, dzięki któremu `less raport.pdf` pokazuje tekst PDF-a, a `less pakiet.tar.gz` listę plików w archiwum.

- **RHEL / Rocky / openSUSE** — `lesspipe.sh` jest częścią pakietu `less`, włączaną przez `/etc/profile.d/less.sh`. Działa domyślnie.
- **Debian / Ubuntu** — osobny pakiet: `sudo apt install lesspipe`, potem `eval "$(lesspipe)"` w `~/.bashrc`.

Sprawdzenie, czy masz to włączone: `echo "$LESSOPEN"` powinno zwrócić coś w rodzaju `||/usr/bin/lesspipe.sh %s`.

9. `less` jako pager systemu.

`man` nie wyświetla stron sam — przekazuje je pagerowi. Dlatego nawigacja w `man` jest identyczna jak w `less`: to jest `less`.

```
export PAGER=less
export MANPAGER='less -R'
```

`PAGER` respektują `git`, `systemctl`, `journalctl`, `psql` i wiele innych narzędzi; `MANPAGER` ma pierwszeństwo dla `man`.

Częste błędy

Zapomniane `-R` — plik pełen `ESC[01;31m`.

```
grep --color=always -i error /var/log/messages | less
```

```
2026-09-17 09:12:44 srv01 app[2291]: ESC[01;31mESC[KerrorESC[mESC[K: upstream unreachable
```

Kolory nie są magią — to znaki sterujące wplecione w tekst, a `less` bez `-R` pokazuje je dosłownie. Poprawka to `less -R` albo na stałe `export LESS='-R'`. Odwrotnie: jeśli chcesz się kodów pozbyć, a nie je interpretować, użyj `--color=never`.

`less` na pliku binarnym — terminal przestaje odpowiadać sensownie.

```
less /usr/bin/ssh
```

```
"/usr/bin/ssh" may be a binary file.  See it anyway?
```

Wciśnięcie `y` wypycha do terminala surowe bajty, w tym sekwencje przełączające zestaw znaków. Po wyjściu ramki rysują się krzaczkami, a wpisywany tekst jest niewidoczny. Ratunek: `reset`, a jeśli go nie ma — `stty sane; printf '\033c'`. Zamiast tego oglądaj bajty w postaci, która terminala nie dotyka:

```
xxd /usr/bin/ssh | less
strings /usr/bin/ssh | less
```

Treść znika z ekranu po naciśnięciu `q`.

Wyszukałeś w logu linię z identyfikatorem transakcji, wyszedłeś z `less`, chcesz ją skopiować — a terminal wygląda tak, jakbyś nic nie otwierał. `less` domyślnie korzysta z „ekranu alternatywnego" i przy wyjściu przywraca poprzednią zawartość. Rozwiązanie:

```
less -X /var/log/messages
```

`-X` wyłącza wysyłanie sekwencji inicjujących i kończących, więc ostatni ekran zostaje. Cena: przy kilkunastu wywołaniach historia terminala zapełnia się zrzutami ekranu, a w części emulatorów `-X` psuje obsługę myszy i kółka.

`export LESS='-FRX'` i znikające strony `man`.

Popularna rada z internetu. Problem leży w `-F` (`--quit-if-one-screen`): jeżeli treść mieści się na jednym ekranie, `less` wypisuje ją i natychmiast kończy. Krótka strona `man`, `git log` z dwoma commitami albo `systemctl status` małej usługi mrugną i znikną. Jeśli już `-F`, to wyłącznie razem z `-X` — inaczej treść zostanie wymazana przy wyjściu.

Zostawiony filtr `&wzorzec` — plik wygląda na okrojony.

Wpisałeś `&error`, obejrzałeś trafienia, wcisnąłeś `g` i widzisz plik, który „ma tylko dwadzieścia linii". Filtr wciąż działa — pasek stanu pokazuje `&error`. Skasowanie: `&` i Enter z pustym wzorcem. Ta sama pomyłka zdarza się z `-S`: dane wyglądają na ucięte, a naprawdę są poza prawą krawędzią ekranu.

□□ Ćwiczenia

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

Ćwiczenie 1 — nawigacja

```
cd /tmp
seq 1 200000 | awk '{printf "%06d %s zdarzenie %d\n", $1, ($1%1000==0?"ERROR":"INFO"), $1*7}'
> duzy.log
less -N /tmp/duzy.log
```

Skocz na koniec (`G`), na początek (`g`), do linii 150 000 (`150000g`). Sprawdź `Ctrl+G`. Znajdź piąte wystąpienie `ERROR` (`/ERROR`, potem `n`).

Czego się spodziewać

`Ctrl+G` pokaże pasek statusu w rodzaju `/tmp/duzy.log lines 149991-150039/200000 byte 3149800/4288896 73%`. Skok `150000g` jest natychmiastowy, mimo że numery linii wymagają policzenia znaków nowej linii od początku pliku.

Ćwiczenie 2 — filtr

W tym samym pliku wciśnij `&`, wpisz `ERROR`, Enter. Policz, ile linii widzisz. Zdejmij filtr (`&`, Enter) i porównaj z:

```
grep -c ERROR /tmp/duzy.log
```

Odpowiedź

Liczby powinny się zgadzać — filtr `&ERROR` w `less` i `grep -c ERROR` liczą to samo, tylko `less` pokazuje pasujące linie w kontekście pliku (z zachowanymi numerami), a `grep -c` daje samą liczbę.

Ćwiczenie 3 — kolory i zawijanie

```
grep --color=always -n ERROR /tmp/duzy.log | less
grep --color=always -n ERROR /tmp/duzy.log | less -R
paste -d';' /tmp/duzy.log /tmp/duzy.log /tmp/duzy.log | less
paste -d';' /tmp/duzy.log /tmp/duzy.log /tmp/duzy.log | less -S
```

Czego się spodziewać

Pierwsza komenda pokaże `ESC[01;31m` wokół każdego `ERROR`. Druga — czerwone `ERROR` bez śmieci. Trzecia zawinie każdą linię na trzy wiersze ekranu. Czwarta utnie je na krawędzi; strzałka w prawo przewinie widok w bok.

Sprzątanie: `rm -f /tmp/duzy.log`.

Ćwiczenie 4 — `more` kontra `less`

Otwórz `more /tmp/duzy.log`, przewiń kilka ekranów spacją, spróbuj wrócić klawiszem `b`. Potem `cat /tmp/duzy.log | more` i ta sama próba.

Odpowiedź

Na zwykłym pliku `b` zadziała — `more` z `util-linux` może zrobić `lseek()`. W potoku ta sama komenda nic nie da: dane przeszły i nie ma po czym się cofać. W `less` ta różnica nie istnieje, bo buforuje strumień w pamięci.

Pytania kontrolne

1. Dlaczego `less` otwiera plik o rozmiarze 40 GB natychmiast, a edytor tekstu na tym samym pliku się zawiesza?

Odpowiedź

`less` czyta tylko tyle, ile potrzebuje do zapełnienia ekranu, i zapamiętuje offsety miejsc, przez które przeszedł; resztę doczytuje na żądanie przez `lseek()`. Edytor wczytuje cały plik do pamięci, bo musi umieć go modyfikować w dowolnym miejscu. Do przeglądania — `less`; do edycji wielkiego pliku — narzędzia strumieniowe (`sed`, `awk`), nie edytor.

2. Otworzyłeś `journalctl -u nginx | less` i pasek stanu pokazuje pozycję w bajtach zamiast procentu. Dlaczego?

Odpowiedź

Dane przychodzą potokiem, a nie z pliku. `less` nie zna docelowego rozmiaru strumienia, więc nie ma z czego policzyć procentu — poda tylko, ile bajtów przeczytał. Po `G` (co wymusza doczytanie do końca) procent się pojawi. Przy okazji: `journalctl` sam uruchamia pager, więc `| less` jest zwykle zbędne.

3. Kiedy `&wzorzec` w `less` jest lepsze niż `grep wzorzec plik`?

Odpowiedź

Kiedy potrzebujesz kontekstu. `grep` daje wynik i na tym koniec — żeby zobaczyć, co było wokół trafienia, musisz uruchomić go ponownie z `-C 5` albo otworzyć plik. W `less` zdejmujesz filtr jednym klawiszem, zostajesz w tym samym miejscu i widzisz otoczenie; filtr możesz też zmieniać w kółko bez ponownego czytania pliku.

4. Kolega dał Ci do wklejenia `export LESS='-FRX'`. Co z tym zrobisz?

Odpowiedź

`-R` (kolory) przyjmiesz bez zastrzeżeń. `-X` ma sens, jeśli świadomie chcesz, żeby treść zostawała na ekranie — kosztem zaśmiecania historii terminala i czasem obsługi myszy. `-F` odrzuć. Rozsądny punkt wyjścia to `export LESS='-R -i -M'`.

Powiązane: cat: wypisywanie i sklejanie plików · head i tail — początek i koniec pliku · Obserwacja plików na żywo: tail -f, less +F, multitail, watch · man — czytanie dokumentacji systemu · grep — wyszukiwanie w plikach

Źródła: `man 1 less` · `man 1 more` · `man 1 lesspipe` · `man 1 zless` · `man 1 man` · POSIX.1-2024 (`more`)

Revision #19
Created 2026-09-17 11:56:31 UTC by Tomasz Konieczny
Updated 2026-10-03 19:07:41 UTC by Tomasz Konieczny