

Podstawowe polecenia w Linux

22 I 2025

S.K.

Spis treści

1	Poruszanie się między folderami, podstawowe operacje na plikach.	2
1.1	Przejdźcie do folderu:	2
1.2	Listowanie zawartości katalogu:	2
1.3	Tworzenie nowego katalogu:	2
1.4	Usuwanie katalogu (tylko gdy jest pusty):	2
1.5	Usuwanie dowolnego pliku:	2
1.6	Przenoszenie plików:	2
1.7	Kopiowanie plików:	3
1.8	Bardzo przydatne opcje, z których warto wyrobić sobie nawyk korzystania:	3
2	Łączenie się z serwerem/innym komputerem, kopiowanie plików.	3
3	Prawa dostępu do plików.	4
4	Zarządzanie procesami.	5
5	Midnight Commander.	5
6	VI(M).	5
6.1	Poruszanie się po dokumencie	6
6.2	Wprowadzanie tekstu	6
6.3	Inne przydatne funkcje i polecenia	6

1 Poruszanie się między folderami, podstawowe operacje na plikach.

1.1 Przejście do folderu:

`cd ścieżka`

W miejsce "ścieżka" można wpisać np.:

1. kolejny katalog `cd dane`
2. całą ścieżkę, np. `/home/sebastian/dane/`
3. katalog domowy `~/`
4. jeden katalog wyżej `../`
5. dwa katalogi wyżej `../../` itp.

1.2 Listowanie zawartości katalogu:

`ls`

Po spacji można podać adres katalogu, który chcemy wylistować, np `ls ~/dane`. W innym przypadku zostanie wylistowana zawartość aktualnego katalogu. Dodatkowe przydatne opcje:

- `-a` "all" pokazuje również ukryte katalogi
- `-l` wyświetla w formie listy
- `-lh` razem z opcją `-l` podaje rozmiary plików w bardziej "ludzkich" jednostkach np. `ls -alh ~/` wylistuje w ten sposób zawartość katalogu domowego.

1.3 Tworzenie nowego katalogu:

`mkdir nazwa_katalogu`

1.4 Usuwanie katalogu (tylko gdy jest pusty):

`rmdir nazwa_katalogu`

1.5 Usuwanie dowolnego pliku:

`rm nazwa pliku`

W przypadku folderów z zawartością należy użyć jeszcze opcji `-r` (rekursywnie).

1.6 Przenoszenie plików:

`mv nazwa_pliku miejsce_docelowe`

1.7 Kopiowanie plików:

```
cp nazwa_pliku miejsce_docelowe
```

Ponownie, w przypadku kopiowania folderu z zawartością trzeba użyć opcji `-r`.

1.8 Bardzo przydatne opcje, z których warto wyrobić sobie nawyk korzystania:

- Przy wpisywaniu ścieżki, nazwy pliku itp. wciśnięcie klawiszu TAB uzupełni nazwę. Jeżeli jest kilka opcji zaczynających się na dane znaki, to należy wcisnąć TAB dwukrotnie - wyświetlą się nam wtedy wszystkie pliki zaczynające się na to, co już wpisaliśmy.
- gwiazdka `*` zastępuje pewien fragment nazwy. Przykłady:
 - chcąc wylistować w danym katalogu wszystkie pliki, których nazwy zaczynają się słowem "dark" wpisujemy `ls dark*`,
 - chcąc skopiować z danego katalogu tylko pliki o rozszerzeniu ".fits" wpisujemy np. `cp ~/obserwacje/obiekt1/*.fits ~/obserwacje/obiekt2/`
 - chcąc usunąć wszystkie pliki, które zaczynają się na "v1787ori" mają w nazwie "V" i kończą się na ".fits" wpisujemy `rm v1787ori*V*.fits`
- wciśnięcie strzałki do góry przywołuje poprzednio wpisane komendy.
- kopiowanie w linux - zaznaczamy tekst myszką, a wklejamy wciskając środkowy przycisk (scroll).
- chcąc przekierować wynik danego polecenia do pliku tekstowego wpisujemy `>` (utworzenie nowego pliku) lub `>>` (dopisanie do pliku). Np. `ls dark* > dark.lst` utworzy nam plik tekstowy "dark.lst" z listą wszystkich plików z danego folderu zaczynających się na "dark".
- Informacje o danym poleceniu, dostępnych opcjach itp. można wyświetlić wpisując `man` lub `--help`, np. dla polecenia listującego zawartość katalogu: `ls --help` lub `man ls`.
- Odpalając w terminalu różne programy, często warto na końcu dodać znak `&`. Dzięki temu będziemy mieli wciąż wolny terminal (np. `gedit plik.txt &`). Jeżeli zapomnimy to zrobić, a chcemy do czegoś użyć terminal, który jest zajęty, to wciskamy CTRL+Z, który wstrzymuje dany program, a następnie wpisujemy `fg` (foreground) aby do niego wrócić, albo `bg` (background) aby do niego wrócić, ale w tle (tak jakbyśmy go włączyli z opcją `&`).

2 Łączenie się z serwerem/innym komputerem, kopiowanie plików.

Połączenie z serwerem:

```
ssh nazwa_uzytkownika@adres_serwera
```

Połączenie z dostępem do trybu graficznego:

```
ssh -Y nazwa_uzytkownika@adres_serwera
ssh -X nazwa_uzytkownika@adres_serwera
ssh -XY nazwa_uzytkownika@adres_serwera
```

Kopiowanie plików z serwera:

```
scp nazwa_uzytkownika@adres_serwera:ściezka_do_pliku docelowa_ściezka_lokalna
```

```
np. scp sebastian@serwer.costam.pl:~/obserwacje/v1787ori/*.fits ~/obserwacje/obiekt1/
```

Kopiowanie plików na serwer:

```
scp ściezka/nazwa_pliku nazwa_uzytkownika@adres_serwera:ściezka_docelowa
```

I znowu, chcąc kopiować folder z zawartością dodajemy opcję `-r`. Przydatne narzędzia dostępne pod Windows:

- Putty - programik pozwalający łączyć się przez ssh (tylko w trybie tekstowym).
- WinScp - wygodny programik pozwalający na kopiowanie plików z i na serwer.

3 Prawa dostępu do plików.

Prawa dostępu do poszczególnych plików możemy zobaczyć wylistowując zawartość folderu poleceniem:

```
ls -l
```

Pierwszy znak mówi nam czy jest to folder (d) czy plik (-). Kolejne trzy opisują prawa dostępu użytkownika (r - odczyt, w - zapis, x - wykonanie), następne trzy prawa dostępu dla grupy, a ostatnie dla innych.

Prawa dostępu możemy zmienić poleceniem `chmod`. Najprostrzy sposób to po nazwie polecenia wpisanie dla kogo chcemy zmienić uprawnienia (u - użytkownik, g - grupa, o - inni, a - wszyscy), następnie co chcemy zrobić (+ - dodać, - - odjąć = - przypisać), potem jakie uprawnienia (r - odczyt, w - zapis, x - wykonanie), a na końcu nazwę pliku. Przykład, w którym użytkownikowi przypisujemy wszystkie prawa, grupa dostaje prawa zapisu, a inni tracą prawa do czytania, zapisu i wykonywania):

```
chmod u=rwx,g+w,o-rwx plik.txt
```

Jeżeli chcemy tylko nadać prawa do wykonywania dla użytkownika to wpisujemy:

```
chmod u+x plik.txt
```

Inną opcją (myślę, że wygodniejszą i szybszą po przyzwyczajeniu) jest zapis uprawnień w trybie ósemkowym. Po poleceniu `chmod` wpisujemy wtedy trzy cyfry odpowiadające uprawnieniom kolejno użytkownika, grupy i innych. Prawa odczytu, zapisu i wykonania mają przypisane wartości odpowiadające potęgom dwójki: $r=2^2=4$, $w=2^1=2$, $x=2^0=1$. Dana cyfra po poleceniu `chmod` to po prostu suma liczb odpowiadających danym uprawnieniom. Np. chcąc nadać użytkownikowi wszystkie uprawnienia ($rwx=4+2+1=7$), grupie tylko odczyt i zapis ($rw=4+2=6$), a pozostałym nie dawać żadnych uprawnień wpisujemy:

```
chmod 760 plik.txt
```

Chyba najlepszy opis różnych opcji polecenia `chmod` znajduje się na Wikipedii: <https://pl.wikipedia.org/wiki/Chmod>.

4 Zarządzanie procesami.

Aktywne procesy możemy wyświetlić na kilka sposobów:

- `ps` - wyświetla aktualną listę procesów. Opcja `-e` wyświetla wszystkie procesy, a `-f` wyświetla więcej informacji o poszczególnych procesach.
- `top` - na bieżąco odświeżana lista procesów. Wychodzimy wciskając "q".
- `htop` - podobnie jak poprzednie, ale w trybie interaktywnym (na dole okna mamy wypisane skróty klawiszowe).

Każdy proces ma przypisany numer (PID). Aby "ubić" dany proces wpisujemy `kill numer_procesu`. Jeżeli chcemy "ubić" jakiś proces/program, który zajmuje terminal to wciskamy CTRL+C. Np. uruchomiliśmy `gedit` (ale bez `&` na końcu), który się zawiesił, albo jakiś program z nieskończoną pętlą, to wciśnięcie CTRL+C powinno wymusić jego zamknięcie).

5 Midnight Commander.

Włączamy go poleceniem `mc`. Klawiszem TAB przeskakujemy między prawą i lewą częścią okna. Przechodzimy do poszczególnych katalogów strzałkami i enterem. `/. .` oznacza katalog wyżej. U dołu okna mamy legendę, gdzie 1,2,3... oznaczają klawisze F1,F2,F3 itp. F9 przechodzi do górnego menu. Jeżeli chcemy w prawej części mieć jakiś katalog na zdalnym komputerze/serwerze to wybieramy opcję "połączenie po powłóce" i wpisujemy `nazwę_użytkownika@adres_serwera`.

W skrócie, ten program umożliwia szybkie poruszanie się po różnych folderach, kopiowanie i przenoszenie plików (również z/na serwer), pakowanie i rozpakowywanie plików itp. Nie musi być jednak domyślnie zainstalowany na każdym komputerze.

6 VI(M).

Włączamy program poleceniem `vim nazwa_pliku`. Są trzy podstawowe tryby:

- "Normalny", w którym poruszamy się po dokumencie, usuwamy linijki itp.
- "Wprowadzania", w którym możemy wpisywać tekst, przechodzimy do niego np. wciskając klawisz "i", a wychodzimy wciskając ESC.
- "Komend", gdy w trybie "Normalnym" wpisujemy polecenia zaczynając od dwukropka ":".

Z Vima wychodzimy wpisując komendę `:q` lub jeżeli wprowadziliśmy jakieś zmiany i chcemy wyjść bez zapisu to `:q!`, a jeśli chcemy wyjść z zapisem to `:wq`. Sam zapis to `:w`.

6.1 Poruszanie się po dokumencie

- poruszanie się po znakach i liniijkach - strzałkami lub **h j k l**
- poruszanie się po wyrazach - **w W b B** (zastosowanie małych liter traktuje również znaki specjalne jako separatory wyrazów).
- przed każdym z powyższych klawiszy można wpisać liczbę, które mówi o ile znaków, wyrazów lub liniijek chcemy przeskoczyć, np. **10j** przeskoczy o 10 linii w dół.
- **e E** - przeskakuje do końca wyrazu.
- **0** lub **^** - początek linii.
- **\$** - koniec linii.
- **1G** lub **gg** - pierwsza linia tekstu.
- **G** - ostatnia linia tekstu.
- **10G** - 10 linia tekstu.
- **H M L** - przejście do pierwszej, środkowej i ostatniej linii w aktualnym widoku.
- **CTRL+d**, **CTRL+f**, **CTRL+y** - scrollowanie w dół (kolejno o pół aktualnego widoku, całą aktualną stronę i jedną linię).
- **CTRL+u**, **CTRL+b**, **CTRL+e** - to samo co wyżej, tylko scrollowanie w górę.

6.2 Wprowadzanie tekstu

- **i** - wprowadzanie tekstu przed zaznaczonym znakiem.
- **I** - na początku aktualnej linii.
- **a** - za zaznaczonym znakiem.
- **A** - na końcu aktualnej linii.
- **o** - w nowej linii pod aktualną.
- **O** - w nowej linii nad aktualną.

6.3 Inne przydatne funkcje i polecenia

- **u** - cofnij (undo).
- **CTRL+R** - cofnięcie cofnięcia.
- **dd** usunięcie aktualnej linii.
- **/słowo** - wyszukanie w tekście "słowo" (zastąpienie znaku / na ? wyszukuje "do góry"). Po wciśnięciu entera **n** wyszukuje następny, a **N** poprzedni.

- `:[range]s/pattern/string/[flags]` - znajdź ("pattern") i zamień (na "string"). Zamiast [range] wpisujemy w których liniach, np. od 2 do 5: `2,5` lub od 3 do końca: `3,$` lub całość: `%`. Zamiast [flags] możemy podać dodatkowe opcje np `c` - confirm lub `i` - case insensitive. Np. chcąc zamienić każde słowo "bo" na "ponieważ" wpisujemy `:%s/bo/ponieważ/`.

Vim ma duuuuuuużo więcej opcji, które są wypisane np. tutaj: <https://leniwy.eu/news,5,Podstawy-edytora-Vim-skroty-klawiszowe.html>.