Podstawowe komendy systemu LINUX

Pomoc

---

man – wyswietla stronę manuala dla polecenia 'program'

- `man program`

---

info – podobnie jak man, wyświetla stronę pomocy dla polecenia 'program'.

Gdy nie znajdzie strony info, szuka strony man i ja wyświetla.

- `info program, pinfo program`

--help – każdy program ma opcję --help lub -h, która wyświetla któtką pomoc

- `jakis_program --help`

Podstawowe komendy

---

cat – wypisuje wszystkie podane mu pliki na standardowe wyjście

- `cat plik` – jeśli nie przekierujemy standardowego wyjścia do innego pliku (>, >>) lub programu (|), to wypisze plik na ekran
- `cat plik1 plik2 plik3` – wypisze po kolei zawartość wszystkich plików

---

tac – wypisuje wszystkie podane mu pliki na standardowe wyjście, ale w odwraca kolejność linii

- *tac plik1 plik2* – wypisze połączone oba pliki, od ostatniej do pierwszej lini

---

echo – powtarza na standardowym wyjściu słowa podane w argumencie

- *echo costam wypisz, czy echo "costam wypisz"* – wypisze 'costam wypisz' i zakończy znakiem nowej linii
- *echo -n "costam wypisz"* – po wypisaniu argumentów, nie wypisze znaku nowej linii

---

wc – liczy linie, słowa, i znaki w pliku

gdy nie podamy argumentu, czyta ze standardowego wejścia

- *cat plik1 plik2 | wc -l* – policzy wszystkie linie z połączonych plików plik1 plik2
- *wc plik* – wypisze linie słowa i znaki oraz nazwe pliku
- *wc -m* – tylko znaki (lub --chars)
- *wc -l* – tylko linie (lub --lines)
- *wc -w* – tylko słowa (lib --words)

---

less – wygodne i szybkie przegladanie plikow tekstowych

- *less plik* – wyświetla zawartośc pliku i pozwa przewijac strony (q-wyjscie)

Pliki i katalogi

---

touch – zmienia czas dostępu i modyfikacji pliku, lub jeśli plik nie istnieje - tworzy go.

- *touch plik*

---

cp – kopiuje plik

- *cp plik1 plik2* – stworzy ./plik2 identyczny z plik1
- *cp plik3 ../katalog/jakis/* – stworzy plik ../katalog/jakis/plik3
- *cp pom.* podkatalog/* – skopiuje wszystkie pliki zaczynajace się na 'pom.' do ./podkatalog/
- *cp plik5 ~/katalog/jakis/pliczek* – stworzy plik ~/katalog/jakis/pliczek

---

mv – przesuwa plik (tym samym służy również do zmiany nazwy)

- *mv plik1 plik2* – zmieni nazwę pliku z ./plik1 na plik2
- *mv plik3 ../katalog/jakis/* – przesunie plik do ../katalog/jakis/plik3
- *mv plik4 podkatalog/* – przesunie plik ./podkatalog/plik4
- *mv plik5 ~/katalog/jakis/pliczek* – przesunie i zmieni nazwę ~/katalog/jakis/pliczek

---

rm – kasuje plik

- *rm plik -*
- *rm -r katalog* – kasuje wszystko w katalogu i wszystkie jego podkatalogi (--recursive)
- *rm -f plik* – nie pyta się czy skasowac (--force)

---

mkdir – tworzy katalog

- *mkdir moj_nowy_katalog*
- *mkdir /home/users/ja/moj_nowy_katalog*

---

rmdir – usuwa pusty katalog

- *rmdir katalog*

---

ls – listuje katalog

- `ls` – listuje katalog . (ls .)
- `ls plik1 plik2 plik3` – listuje tylko wymienione pliki
- `ls *.txt` – wypisze wszystkie pliki o nazwie konczącej się na '.txt'
- `ls katalog1 katalog2` – listuje wymienione katalogi
- `ls -l` – szczegółowa lista
- `ls -a` – wypisuje również ukryte pliki (czyli te których nazwa zaczyna się kropką)
- `ls -R` – listuje katalogi rekursywnie (czyli wyświetla również zawartość podkatalógow)
- `ls -d` – wyświetla tylko nazwy katalogow, tak jak zwyczajnych plików, czyli nie listuje ich zawartości

---

chmod – zmienia prawa dostępu do pliku

grupy użytkowników: u - user, g - group, o - others, a - all
prawa dostępu: r - read, w - write, x - execute

- `chmod o+r plik` – udziel innym prawo do czylania pliku
- `chmod a-x plik` – zabierz wszystkim prawo do wykonywania pliku
- `chmod g=rw plik` – ustaw prawa do czytania i pisania dla swojej grupy
- `chmod -R go+w katalog` – ustawia prawa wszystkim plikom w katalogu i jego podkatalogach (--recursive)

Przekierowania

> – przekierowanie wyjścia z programu do pliku.

Standardowym wyjściem każdego programu jest ekran (konsola tekstowa) a standardowym wejściem jest klawiatura. Można te wejścia i wyjścia przekierowywać dowolnie.

- `echo "ala ma kota" > plik.txt` – wyjście z programu echo wpisze do pliku plik.txt
- `ls -l > lista.dat` – wypisze liste plików do pliku lista.dat

>> – doklejenie wyjścia z programu na koniec pliku

- *echo "ala ma psa" >> plik.txt* – dopisze "ala ma psa" na koniec pliku plik.txt
- *ls -l > lista.dat* – wypisze liste plików do pliku lista.dat

---

| – przekierowanie wyjścia jednego programu na wejście drugiego

- *cat plik.txt | wc -l* – cat wypisuje plik.txt na wyjście które przekierowujemy na program liczący wiersze.
- *ls -l | lpr* – program drukujący 'lpr' dostanie na wejście liste plików
- *cat plik.txt | tac | grep "coś" | head > cosie.txt* – wypisanie pliku.txt na program 'tac', który odwaca kolejność wierszy, wynik tego przekierwany na 'grep', który wypisze tylko linie zawierające słowo "coś", wynik tego wysłany na program 'head', który pośle dalej tylk opierwsze 10 wierszy na wyjście, które przekierowaliśmy do pliku cosie.txt.

>! – przekierowanie do pliku. Działa podobnie jak >, ale kontynuuje nawet gdy plik już istnienie. Działa w "tcsh".

- *echo "ala ma kota" > plik.txt* – gdy plik.txt istnieje, ta komenda może się nie powieść.
- *echo "ala ma kota" >! plik.txt* – konieczne będzie użycie wykrzyknika >!
- *echo "ala ma kota" >| plik.txt* – to samo tylko w 'bash'

< – przekierowanie pliku jako standardowego wejścia.

- *szachy < ruchy.txt* – jeśli program szachy przyjmuje ruchy na standardowe wejście, możemy te ruchy spisać do pliku i podać programowi w ten sposób
- *cat ruchy.txt | szachy* – to samo można też zrobić tak

<< – podanie na wejście nastepujących później linii.

- ```
  szachy << DO_KONCA
  E2-B4
  H5-A1
  ```
  ```

C6-F5
DO_KONCA – podanie tzw. dokumentu w miejscu. Wszystkie następujące linie między etykietami 'DO_KONCA' będą podane na standardowe wejście programu 'szachy'.

- *echo "E2-B4*
  H5-A1
  C6-F5" | *szachy* – to samo można też zrobić tak
- *echo -e "E2-B4\nH5-A1\nC6-F5"* | *szachy* – to samo można też zrobić tak

---

2> – przekierowanie standardowego wyjścia dla błędów do pliku. Oprócz standardowego wyjścia i wejścia, każdy program ma jeszcze standardowe wyjście dla błędów. Je też możemy przekierowywać, na przykład w inne miejsce niż wyjście zwykłe. Działa w "bash", nie w "tcsh".

- *find -name "plik.*" >znalezione.log 2>bledy.log* – pliki znalezione przez 'find' wylądują w znalezione.log, komunikaty o błędach nie zaciemnią nam wyniku i wpiszą sie do innego pliku – bledy.log
- *cp -r dane/ backup/ 2>error.log* – jeśli podczas kopiowania całego katalogu wystąpią błędy, wszystkie komunikaty zostaną wpisane do error.log
- *(ls > plik.log) >& plik.err* – w 'tcsh' nie można przekierować samego wyjścia dla błędów, stąd konieczność takiej konstrukcji.

&> lub >& – przekierowanie obu wyjść do pliku.

- *ls >& plik.log* – standardowe wyjście i standardowe wyjście dla błędów jest przekierowane do plik.log
- *ls >plik.log 2>&1* – to samo, ale działa tylko w 'bash'. Przekierowanie wyjścia, a potem skopiowanie go na wyjście dla błędów.
- *ls &> plik.log* – to samo co >& ale w notacji bardziej właściwej dla 'bash'.

---

man bash

- *polecam aby uzyskać więcej informacji*

---

man tcsh

- *polecam aby uzyskać więcej informacji*

Procesy

---

ps – wypisuje procesy uruchomione na komputerze

- *ps* – wyświetla procesy uruchomione przez użytkownika
- *ps a* – wyświetl również procesy innych użytkowników
- *ps -l, ps -f, ps -F* – więcej informacji o procesach (od: long, full, extra full)
- *ps f* – wyswietla drzewo zaleznosci procesow (od: forest)
- *ps --help* – :P

---

bg – uruchamia na nowo zatrzymane (Ctrl-Z) zadanie, ale w tle, tak jakby zostało ono uruchomione z &

- *bg* – uruchamia ostatnio zatrzymane zadanie
- *bg NUMER* – uruchamia zadanie o danym numerze na liście zatrzymanych zadań (jobs)

---

fg – uruchamia na nowo zatrzymane (Ctrl-Z) zadanie, na pierwszym planie

- *fg* – uruchamia ostatnio zatrzymane zadanie
- *fg NUMER* – uruchamia zadanie o danym numerze na liście zatrzymanych zadań (jobs)

---

jobs – wyświetla liste zatrzymanych zadań

---

kill – zabija podany proces

PID - jest to numer identyfikacyjny procesu (process id), można go odczytać np. używając polecenia ps

- *kill PID* – wysyła do procesu o numerze PID sygnał do przerwania procesu
- *kill -KILL PID* – zabija proces bez pytania

& – modyfikator który pozwala uruchomić od razu proces w tle.

- *xcalc &* – uruchamia program xcalc w tle, dzięki temu mamy wolną konsole

Edytory

---

vim – zaawansowany edytor tekstowy w trybie tekstowym

Vi iMproved - nowa wersja znanego edytora Vi. Posiada

- *podświetlanie składni dla wielu języków programowania*
- *możliwość edycji kilku plików na raz*
- *bogatego helpa*
- *bardzo zaawansowane funkcje edycji*
- *...*

---

gvim – vim w trybie graficznym

---

emacs – zaawansowany edytor tekstowy w trybie graficznym

Emacs podobnie jak Vim jest wszechstronnym edytorem obsługującym wiele języków i posidającym bogate funkcje.

---

uemacs – edytor telstowy w trybie teksowym

Micro Emacs jest tekstową wersją Emacsa

---

joe – prosty edytor tekstowy

Joe's Own Editor. Nadaje się do pisania prostych dokumentów

---

mcedit – edytor tekstowy w trybie tekstowym

mcedit jest wbudowanym w Midnight Commandera edytorem.
Posiada m.in. podświetlanie składni.

Sieć

---

pine – program do obsługi poczty

Bardzo dobry, szybki i wygodny program do sprawdzania i wysyłania poczty elektronicznej. Jego największą zaletą jest to, że działa w trybie tektowym, wic można uruchomić go na zdalnym terminalu.

---

ssh – program do zdalnego logowania używając szyfrowanego połączenia

Najważniejszy sieciowy program. Umożliwia logowanie się na dowolny komputer na świecie, przy czym połączenie jest bezpieczne dzięki algorytmom szyfrującym opartym na kluczach RSA.

- `ssh anatres` – zaloguje mnie na 'antares'a
- `ssh ja@anatres` – zaloguje mnie jako użytkownika 'ja' na 'antares'a
- `ssh ja@antares komenda` – zaloguje mnie tylko by wykonac na 'antares'ie komendę

---

scp – program do kopiowania plików używając szyfrowanego połączenia SSH

scp łączy się z podanym serwerem i kopiuje podane pliki między oboma komputerami
'scp'do połączenia używa programu 'ssh'

- `scp plik ja@antares:~/moje_pliki/` – skopiuje plik do mojego katalogu na antaresie ~/moje_pliki/
- `scp ja@antares:/var/log/plik .` – do bierzącego katalogu skopiuje podany plik z antaresa

---

rlogin – prosty protokól zdalnego logowania

- *`rlogin antares`* – zaloguje mnie na 'antares'a

---

ping – program diagnostyczny sprawdzający czy istnieje połączenie sieciowe z danym komputerem.

- *`ping antares.astrouw.edu.pl`* – sprawdzamy czy antares odpowiada (i jak szybko)