

Poruszanie się po terminalu:

Cd – pozwala przechodzić między plikami

Cd .. - cofa nas o jeden plik

Cd ~ - cofa nas do katalogu home

Clear - czyści okno terminala

Ls – pokazuje wnętrze katalogu w którym jesteśmy

Pwd – pokazuje obecną lokalizację

Tree – drzewo katalogów

Zarządzanie plikami:

Echo "napis" >> plik.txt - dodawanie napisu do pliku.txt

Mv staryplik.txt nowyplik.txt - zmiana nazwy pliku

Mv /lokalizacja1/katalog /lokalizacja2 - przenoszenie katalogu z lokalizacji1 do lokalizacji 2

Mv /lokalizacja1/katalog . - przeniesienie katalogu z lokalizacji1 do naszej obecnej lokalizacji

Nano plik.txt - tworzenie pliku z zawartością (edytor nano)

Touch plik.txt - tworzenie pustego pliku

Chmod opcja plik.txt - zmienia uprawnienia do pliku

"- **rwx** **rwx** **rwx** " - **plik(-)/katalog(d)** **użytkownik** **grupy** **reszta**

" r " - prawa do odczytywania

" w " - prawa do nadpisywania

" x " - prawa do wykonywania

Chmod **uga**+w plik.txt - dodanie uprawnienia do zapisu dla **użytkownika**, **grupy** i pozostałych

Tar -czvf archiwum.tar.gz pliki_do_archiwizacji – tworzenie archiwum z plików

Tar -tzvf archiwum.tar.gz – wyświetlanie archiwum

2>plik.txt - przekierowanie błędu do pliku.txt

2>/dev/null - przekierowanie błędu do kosza

Szukanie i zagłębienie do plików:

Cat plik.txt - pokazuje w terminalu zawartość pliku (wyświetla tyle ile się zmieści na 1 stronie terminala)

More plik.txt - pokazuje zawartość pliku (nie pozwala na cofanie raz przeczytanej linijki)

Less plik.txt - pokazuje zawartość pliku (Idealny do przeglądania dużych plików)

Grep szukane_słowo plik.txt - szuka wybranego słowa w pliku.txt

Grep słowo plik1.txt plik2.txt - przeszukiwanie 2 plików

Grep -c słowo plik.txt - komenda liczy ile razy w plik.txt padło dane słowo

Grep -E "..słowo.*" plik.txt - komenda będzie czytać znaki specjalne

" ." - dowolny znak

" .* " - dowolna ilość znaków

" \$ " - koniec

" ^ " - początek

" [] " - zakres (to lub to ... np. [jJ] małe lub duże J)

" [-] " - zakres ciągu (np. [1-4] oznacza ciąg 1234)

" | " - lub

" \s " - znak pusty (spacja, tabulacja)

-v – przeczenie

-i - ignorowanie wielkości liter

-B liczba – pokazuje dodatkowo 2 wcześniejsze linijki

-A liczba – pokazuje dodatkowo 2 następne linijki

-n – pokazuje numer linii

egrep szukane_słowo plik.txt - jest to w zasadzie grep -E

Tail plik.txt - wyświetla ostatnie 10 linijek tekstu z pliku.txt

Tail -n 20 plik.txt - wyświetli ostatnie 20 linijek tekstu

Head plik.txt - wyświetla pierwsze 10 linijek tekstu z pliku.txt

Head -n 20 plik.txt - wyświetla pierwsze 20 linijek z tekstu

Skrypty Bash:

Which bash – sprawdza lokalizację interpretera bash

#!/bin/bash - inicjowanie skryptu bash (! + lokalizacja interpretera)

Chmod +x nazwa.sh - nadawanie uprawnień dla skryptu by mógł działać

Zmienna=10 - tworzenie zmiennej (uwaga na spację)

\$zmienna - wywołanie zmiennej

Wynik=\$((liczba1+liczba2)) - operacje na zmiennych

Let wynik=\$liczba1+\$liczba2 - przypisuje wynik do zmiennej

Expr \$liczba1+\$liczba2 - wykonanie obliczenia

Wynik=`expr \$liczba1+\$liczba2` - przypisanie wyniku polecenia do zmiennej

Touch \${zmienna}_cokolwiek – tworzenie pliku z zmienną w nazwie

./plik.sh argument1 argument2 - włączanie skryptu wraz z argumentami

\$0 – nazwa skryptu

\$1 \$2 – 1 i 2 argument

\$@ - wszystkie argumenty

\$# - ilość argumentów podanych w terminalu

If [warunek] then ... fi – funkcja warunkowa if

-eq - ==

-gt - >

-lt - <

-ge - >=

-le - <=

-ne - !=

If [warunek1] && [warunek2] - funkcja warunkowa dla większej ilości argumentów

Elif [warunek] – w innym przypadku jeżeli (else if)

Else – w innym wypadku

While [warunek] do ... done - pętla while

((zmienna++)) - `zmienna = zmienna+1`

While ((warunek)) do ... done - pętla while (ale z warunkami typu <=)

Until [warunek] do ... done - pętla wykonująca się tak długo aż warunek nie zostanie spełniony

For zmienna in \$zdanie do echo \$zmienna done - Pętla wypisze nam słowo po słowie ze zdania

For ((warunek)) do ... done - pętla for

Case \$zmienna in opcja1) ...;; opcja2) ...;; *) ...;; esac – Linuxowy switche

Używanie C++:

G++ plik.cpp - `kompilewanie pliku do pliku domyślnego (a.out)`

G++ plik.cpp -o nowanazwa.bin - `kompilewanie pliku do pliku o nazwie nowanazwa.cpp`

2>plik.txt ./plik.bin - `przekierowanie błędów z pliku plik.bin do programu.txt`

Reszta komend:

./plik.txt - `otwieranie pliku`

Cat /etc/os-release – `wyświetla podstawowe informacje o wersji Linuxa`

History - `wyświetla historię użytych przez nas komend`

Komenda 2> plik.txt - `przekierowanie błędów do pliku.txt`

Komenda 2>/dev/null - `wrzucenie błędów do śmietnika`

Cal – `pokazuje kalendarz`

Date – `pokazuje obecną datę`

Date '+%m/%d/%Y' - `pokazuje datę w formacie dzień/miesiąc/Rok`

Whoami – `zwraca nazwę użytkownika`

sl- `pociąg`

Cmatrix – `matrix w terminalu`

Fortune – `losowe zdanie`

Fortune | Cowsay – `Krowa mówi losowe zdanie`

Cowsay – `krowa mówi`

Openttd - `gierka`

Instalowanie komend:

Komenda - -version – [sprawdzanie wersji komendy](#)

Sudo apt-get install build-essential – [instalacja kompilatorów do C++](#)