

#### 4. LOGOWANIE I PODSTAWOWE POLECENIA

##### Co to jest powłoka ?

Powłoka (shell) jest interpretatorem poleceń przeczytanych z klawiatury lub pliku - jest czymś w rodzaju command.com w systemach windowsowych.

Każde polecenie wpisane z klawiatury zostaje zamienione przez powłokę na język zrozumiały dla jądra systemu, które może odpowiednio zareagować na to polecenie.

Linuks posiada kilka powłok : bash, sh, csh, tcsh, ale domyślnym shell'em jest bash, dlatego pozostałe opisy będą dotyczyć tej właśnie powłoki.

Po załadowaniu systemu system przywita nas graficznym menadżerem logowania (jeżeli zainstalowaliśmy) lub konsolą tekstową.

W obu przypadkach system poprosi nas o podanie nazwy użytkownika (login) i jego hasła (password).

Jeśli podczas instalacji stworzyliśmy dodatkowego użytkownika, dobrym rozwiązaniem jest zalogowanie się właśnie jako ten użytkownik.

**Należy pamiętać, że będąc zalogowanym w systemie, jako ROOT posiadamy pełne uprawnienia, co oznacza, że możemy wykonać dowolną operację - nawet przypadkowo skasować cały system.**

Będąc zalogowanym, jako zwykły użytkownik w każdej chwili możemy stać się superużytkownikiem lub innym użytkownikiem mającym konto na naszym komputerze. Do zmiany użytkownika służy komenda wydawana z linii poleceń konsoli :

**su nazwa\_użytkownika**

po wydaniu komendy system poprosi nas o podanie hasła dla tego użytkownika.

Aby stać się z powrotem poprzednim użytkownikiem należy wydać polecenie :

**exit**

#### SZUKANIE POMOCY I PODRĘCZNIKI MAN

Nawet najlepszy administrator w końcu dochodzi do wniosku, że nie wie wszystkiego o systemie i w pewnych sytuacjach potrzebuje pomocy.

Strony man stanowią pierwszą pomoc w takich sytuacjach. Zawierają one definicje i objaśnienia poleceń systemu wraz z opisem opcjonalnych parametrów dla specjalnych funkcji poleceń.

**man polecenie** np.: **man ls** - pokaże nam opis polecenia **ls** służącego do listowania zawartości katalogów.

## PODSTAWOWE POLECENIA

Proponuję Państwu stworzyć sobie listę podstawowych poleceń Linuksa, posegregowanych w grupy w zależności od przeznaczenia. Opis każdego z tych poleceń można zobaczyć poprzez wydanie komendy :

**man polecenie polecenie --help**

**(skrótowy opis)**

**Polecenia można podzielić na następujące grupy :**

- polecenia związane z plikami i katalogami
- polecenia związane z systemem plików
- polecenia związane z zarządzaniem użytkownikami
- polecenia związane z zarządzaniem modułami jądra
- polecenia związane z procesami
- polecenia związane z siecią
- polecenia związane z wyszukiwaniem
- polecenia związane z szukaniem pomocy - inne polecenia

Podstawowe polecenia związane z **plikami i katalogami** np.: **pwd**

– **gdzie jesteśmy** ☺

**cd ścieżka\_dostępu**

Opis : zmiana bieżącego katalogu

Aby wejść do jakiegoś katalogu, możemy podać bezpośrednią ścieżkę dostępu do katalogu, albo pośrednią (liczoną od miejsca, w którym się aktualnie znajdujemy). Dajmy na to przykład. Chcemy wejść do katalogu test, który znajduje się w naszym katalogu domowym. Więc w konsoli wpisujemy: `cd /home/sXXXXX/test` lub postać równoważną:

`cd ~/test`

Obie wyżej pokazane postaci są metodami bezpośrednimi, ponieważ podajemy bezpośrednią ścieżkę do katalogu test (licząc od katalogu głównego).

Znak tylda (~) oznacza /home/ sXXXXX/. Czyli katalog domowy użytkownika sXXXXX. Jeśli jesteśmy w katalogu domowym, to możemy dostać się do katalogu test inaczej. W konsoli wpisując:

`cd test`

Jest to metoda pośrednia, ponieważ podajemy nazwę katalogu (test), który znajduje się w bieżącym katalogu

Przejdźcie do katalogu nadrzędnego odbywa się poprzez dwie kropki (..). W konsoli wpisujemy

```
cd ..
```

Innym ważnym operatorem jest minus (-). Wpisując w konsoli:

```
cd -
```

Przechodzimy do katalogu, w którym byliśmy ostatnio.

### **ls [-al] plik\_lub\_katalog**

Opis : wyświetla informacje o plikach i katalogach Opcje

:

**-a** - wyświetla wszystkie pliki w katalogu, wraz z plikami ukrytymi

**-l** - listuje w długim formacie pełną informację o plikach i katalogach, pokazuje prawa do pliku

**-o** – pokazuje nie tylko nazwę pliku, ale także prawa dostępu, rozmiar, właściciela oraz datę modyfikacji

**-f** zawartość nieposortowana

**-i** – pokazuje i-węzły (i-node)

**-p** zaznaczenia katalogów przez dodanie ukośnika ( katalog/ ) **--sort**

– sortuje pliki - można go ustawić na:

**--sort=size** – sortuje wg rozmiaru

**--sort=time** – sortuje według czasu modyfikacji

**--sort=extension** – sortuje według rozszerzenia. Pliki bez rozszerzenia będą na początku

**-r** – odwraca sortowanie

**-F** dopisz / po nazwie katalogu , \* po pliku wykonywalnym , @ po nazwie plików powiązanych

**-R** wyświetlenie zawartości katalogu wraz z zawartością podkatalogów

**-h** – pokazuje rozmiary w wygodnych jednostkach

**-T** liczba – ustala ilość kolumn (oczywiście, jeżeli wystarczy miejsca)

**maski** – pisząc np. \*.txt na końcu polecenia, ls pokaże wszystkie pliki z końcówką .txt

Przykłady:

Aby dowiedzieć się więcej informacji o zawartości katalogu przychodzi z pomocą opcja -l. Rysunek poniżej zawiera więcej informacji na temat zawartości katalogu, o których zaraz powiem.

```
gruby@earth:~/Desktop/test$ ls -l
total 28
lrwxrwxrwx 1 gruby gruby 5 2010-05-31 19:14 dowiazanie_do_plik1 -> plik1
drwxr-xr-x 2 gruby gruby 4096 2010-05-31 18:55 katalog1
-rwxr-xr-x 1 gruby gruby 8297 2010-05-31 18:56 main
-rw-r--r-- 1 gruby gruby 77 2010-05-31 18:56 main.c
-rw-r--r-- 1 gruby gruby 30 2010-05-31 18:55 plik1
-rw-r--r-- 1 gruby gruby 30 2010-05-31 19:12 plik2
gruby@earth:~/Desktop/test$
```

Polecenie ls -l

Na następnym rysunku pokazane zostało polecenie ls, z opcją -l, oraz konkretnym plikiem.

```
gruby@earth:~/Desktop/test$ ls -l plik1
-rw-r--r-- 1 gruby gruby 30 2010-05-31 18:55 plik1
(1) (2) (3) (4) (5) (6) (7)
```

Polecenie ls -l plik1

Gdzie:

- (1) - Uprawnienia do pliku (o uprawnieniach to w innym miejscu)
- (2) - Liczba łączy
- (3) - ID użytkownika
- (4) - ID grupy
- (5) - Liczba bajtów
- (6) - Data ostatniej modyfikacji
- (7) - Nazwa pliku

Oba w/w polecenia nie pokazują plików ukrytych (tj. zaczynających się od kropki). Tak więc, aby zobaczyć ukryte pliki, które znajdują się w katalogu należy użyć polecenia ls, z odpowiednią opcją.

Opcje te, to -a, -A. Różnica pomiędzy nimi pokazana została na kolejnym przykładzie

```
gruby@earth:~/Desktop/test$ ls -a
.  dowiazanie_do_plik1  main  plik1  .ukryty_katalog
.. katalog1            main.c plik2  .ukryty_plik.txt
gruby@earth:~/Desktop/test$ ls -A
dowiazanie_do_plik1  main  plik1  .ukryty_katalog
katalog1            main.c plik2  .ukryty_plik.txt
gruby@earth:~/Desktop/test$
```

Polecenie ls -a, ls -A

Jak widać, polecenie ls -a, wyświetla zawartość katalogu, wraz z wszystkimi ukrytymi plikami, oraz pokazuje "." (kropka), oraz ".." (dwie kropki) co odpowiednio oznacza katalog bieżący, oraz katalog nadrzędny. Polecenie ls -A, wyświetla zawartość katalogu wraz z plikami ukrytymi, lecz pomija "kropki".

## HISTORIA POLECEŃ

Powłoka bash posiada zdolność przywoływania wcześniej wydawanych poleceń przez użycie klawiszy kursora. Bash rejestruje je w pliku `.bash_history`, który znajduje się w katalogu domowym użytkownika.

Historię poleceń określają dwa parametry :

HISTFILE - wskazuje na plik zawierający wywołane wcześniej polecenia

HISTSIZE - określa ile ostatnio wydanych poleceń ma być przechowywanych w pliku historii

Aby obejrzeć listę ostatnio wydawanych poleceń bez otwierania powyższego pliku należy skorzystać z polecenia history z parametrem określającym liczbę wierszy, które mają być wyświetlone :

`history 10`

pokaże 10 ostatnio użytych poleceń.

Aby powtórzyć wcześniej wydane polecenie można użyć następującej komendy :

`!3`

gdzie 3 jest trzecią komendą z listy wyświetlonej poleceniem history .

### Szybkie uruchamianie poprzednich poleceń

Bash udostępnia nam kilka skrótów, dzięki którym możemy wykonać ponownie polecenie jakie właśnie się zakończyło. Istnieje kilka metod na wykonanie tej czynności:

Wciskając strzałkę do góry pojawi się nam poprzednie polecenie

`!!` – uruchomione zostanie poprzednio wydane polecenie

`!-1` – uruchomione zostanie poprzednio wydane polecenie

`[Ctrl] + P` – wyświetli się poprzednio wydane polecenie

<http://osworld.pl/sztuczki-z-bash-history/>