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

Proponuje 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 (skrócony 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 postac 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
Przejście 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 – pokzauje 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

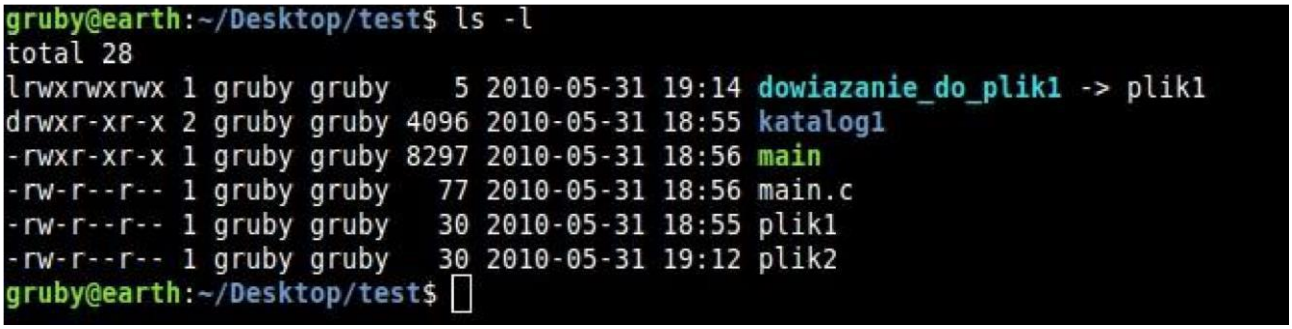
-F dopisz / po nazwie katalogu , * po pliku wykonywalnym , @ po nazwie plików powiązanych

-R wyświetlenie zawartości katalogu wraz z zawartością podkatalogów

-h – pokazuje rozmiary w wygodnych jednostkach

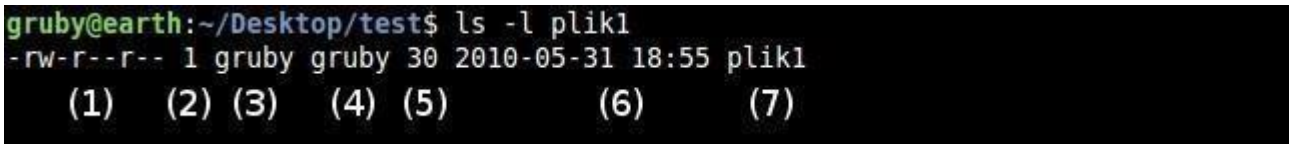
-T liczba – ustala ilość kolumn (oczywiście, jeżeli wystarczy miejsca)

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.



Polecenie ls -l

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



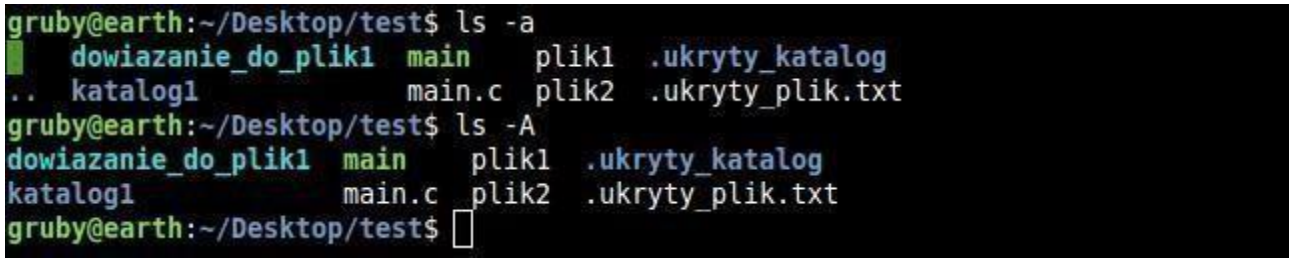
Polecenie ls -l plik1

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.

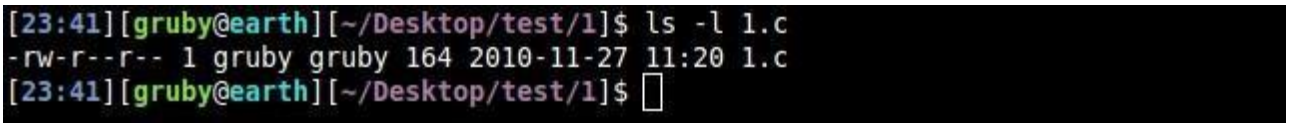
Opcje te, to -a, -A. Różnica pomiędzy nimi pokazana została na kolejnym przykładzie



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".

W Linuksie każdy plik (katalog) ma swoje określone prawa dostępu. Definiują one czy plik może zostać odczytany, czy można do niego pisać, czy można go wykonać. Prawa dostępu mogą występować w różnych kombinacjach. Aby sprawdzić prawa dostępu, trzeba użyć polecenia ls z opcją **-l**. Na rysunku poniżej pokazany został przykładowy plik z pewnymi prawami dostępu, które zostały poniżej poddane szczegółowej analizie.



Prawa dostępu do pliku.

Każdy plik posiada uprawnienia podzielone na trzy części, które odpowiednio oznaczają: **prawa dostępu właściciela pliku**, **prawa dostępu grupy**, oraz **prawa dostępu pozostałych**. W naszym przypadku pierwsza część to **rw-**, druga to **r--** oraz trzecia to **r--**. Na poniższym rysunku przedstawiony został wykaz praw dostępu z wartościami literowymi oraz liczbowymi (te przydadzą się podczas nadawania praw dostępu później)

Wartość		Prawo dostępu
Ósemkowa	Literowa	
4	r	Czytanie
2	w	Pisanie
1	x	Wykonywanie

Prawa dostępu do pliku - wykaz.

Prawa dostępu można łączyć, dlatego też **prawo dostępu właściciela** pliku oznacza, że może on czytać (**r**) plik, oraz pisać (**w**) do pliku, natomiast nie może pliku wykonywać (uruchamiać np. pliku

wykonywalnego, skryptu). **Prawo dostępu grupy** oraz **prawo dostępu pozostałych** oznaczają, że użytkownicy grupy oraz pozostałe osoby mogą ten plik tylko czytać.

Na rysunku z „Prawa dostępu do pliku" prawa dostępu zaczynają się od minusa (-), ten znak nie jest określeniem jakiegokolwiek prawa, a informacją o tym, że plik ten jest plikiem zwykłym. Na poniższym rysunku pokazano kilka możliwych znaków, które mogą występować na pierwszej pozycji.

Lp	Znak	Znaczenie
1	-	Zwykły plik
2	d	Katalog
3	l	Dowiązanie symboliczne
4	s	Gniazdo
5	f	FIFO
6	c	Urządzenie znakowe
7	b	Urządzenie blokowe

Prawa dostępu do pliku - Typ pliku.

Jeśli chcemy zmienić prawa dostępu do pliku, musimy użyć polecenia chmod, którego sposób użycia w ogólnych przypadkach zostanie pokazany później.

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 wykonań 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/

Aliasy

Bash daje nam ciekawy mechanizm aliasów umożliwiających zastępowanie złożonych poleceń krótszymi, łatwiejszymi do zapamiętania nazwami.
Aliasy tworzymy korzystając z polecenia **alias** , którego składnia jest następująca :

alias
'nasza_definicja'='polecenie' np.:
alias 'p'='ps -aux'

Teraz poprzez wydanie polecenia **p** otrzymamy listę wszystkich procesów. Zdefiniowane aliasy możemy zlikwidować poleceniem **unalias** :

unalias p

Dowiązania stałe i symboliczne

Mechanizm dowiązań (linków) ułatwia dostęp do plików ukrytych głęboko w strukturze katalogów lub umożliwia zgromadzenie potrzebnych plików w jednym katalogu.

Linki dzielimy na **stałe** i **symboliczne**.

Dowiązania stałe (łącze sztywne) charakteryzują się tym, że posiadają ten sam i-węzeł co wskazywany plik. W praktyce oznacza to, że dowiązanie takie zawiera pełną informację o pliku i w razie skasowania oryginalnego pliku dowiązanie nadal będzie zawierać informacje o tym pliku (jego zawartość).
Dowiązanie stałe tworzymy poleceniem :

ln plik_lub_katalog link

Dowiązania symboliczne (łącznik symboliczny) służą jedynie do wskazywania na jakiś plik. Umożliwiają one prace na danym pliku poprzez jego dowiązanie. W przypadku skasowania pliku, do którego prowadzi link symboliczny niemożliwe staje się korzystanie z tego dowiązania (nie zawiera on wówczas żadnych informacji).
Dowiązania symboliczne tworzymy poleceniem :

ln -s plik_lub_katalog link

Przekierowania wejścia i wyjścia

Przekierowania zazwyczaj wykorzystywane są do odczytania danych z jakiegoś urządzenia lub pliku albo do wysłania danych na urządzenie lub plik.

Do przekierowań danych służą specjalne znaki :
< dane na wejście
> dane z wyjścia

Najprostszym przykładem przekierowania może być wyświetlenie zawartości jakiegoś pliku

poleceniem : **cat < plik.txt**

Aby utworzyć nowy plik i wpisać do niego dowolny tekst możemy użyć przekierowania danych z wyjścia :

cat > plik.txt

Polecenie to utworzy nam plik : **plik.txt** i umieści w nim tekst wpisany z klawiatury. Jeśli

będziemy chcieli ponownie dodać jakiś tekst do tego pliku należy użyć polecenia :

cat >> plik.txt

Spowoduje to dodanie następnych wierszy z informacją do pliku.

Innym przykładem przekierowań może być zapisanie do pliku zawartości jakiegoś katalogu :

ls -al > plik.txt

Łączenie poleceń za pomocą potoków

Połączenie poleceń za pomocą potoków oznacza wysłanie wyniku z jednego polecenia na wejście drugiego polecenia. Potoki są bardzo często wykorzystywane podczas pisania wszelakich skryptów.

Polecenie : **ls -al | less** łączy

ze sobą dwa polecenia :

ls -al - listuje zawartość katalogu **less** - umożliwia przeglądanie dużych
plików za pomocą klawiszy kursora.
Wynikiem tego połączenia jest możliwość obejrzenia zawartości dużego katalogu za pomocą klawiszy kursora.