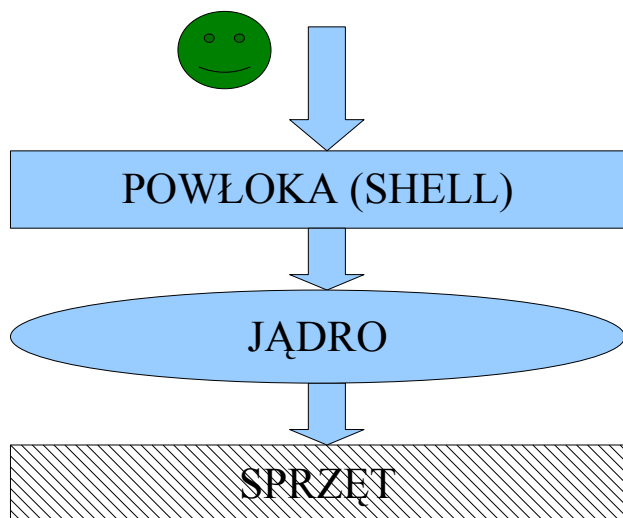




Podstawy systemów UNIX

Autor: Maciej Friedel <maciek@friedel.pl>
Zajęcia prowadzone dla Polskiej Szkoły IT
Wrocław, 2008

Struktura systemu UNIX



Pracę całego systemu koordynuje **jądro** (ang. kernel) – komunikuje się ono ze sprzętem pośrednicząc pomiędzy nim a procesami, zarządza przydziałem pamięci i czasu procesora i systemem plików. Żaden program nie ma bezpośredniego dostępu do urządzeń wej/wyj. Dostęp do jądra jest możliwy tylko przez wywołania funkcji systemowych (jest to charakterystyczna cecha systemu UNIX).

Między jądrem a użytkownikiem znajduje się **powłoka** (ang. shell) będąca interpretatorem poleceń użytkownika. Powłoka wywołuje także odpowiednie programy i zwraca ich wyniki wyprowadzane przez standardowe wyjście.

W systemach unixowych dostępnych jest kilka powłok, z których najbardziej popularne to:
Bourne Shell (sh),
C-Shell (csh),
Korn shell (ksh).

Każdy zbiór danych w systemach unixowych jest plikiem.

Występują tu następujące rodzaje plików:

- Pliki normalne (ang. regular file) – zawierające różne dane,
- Katalogi (ang. directories) – zawierają nazwy innych plików oraz informacje o ich miejscu na dysku, prawach dostępu, itp.
- Pliki specjalne – reprezentują urządzenia znakowe komputera (ang. character special file) i urządzenia blokowe (ang. block special file). Dzięki temu mechanizmowi są widzialne w systemie plików jako zbiory.
- Kolejki FIFO (ang. fifo queue), pliki będące kanałami wejścia wyjścia,
- Skróty symboliczne (ang. symbolic link) - pliki zawierające nazwę ścieżki do innego pliku lub katalogu,
- Potoki – jednokierunkowe ograniczone bufory łączące standardowe wyjście jednego procesu ze standardowym wejściem innego.

Struktura katalogów

W uniksach cały system plików widoczny jest w postaci katalogów umieszczonych w wirtualnym katalogu “/”.

/root – katalog domowy administratora

/sbin – polecenia przeznaczone do użytku administratora

/bin – zawierający podstawowe polecenia konsoli

/etc – zawierający pliki konfiguracyjne systemu

/proc – wirtualny katalog zawierający wirtualne pliki z informacjami o stanie systemu

/boot – zawierający m. in. informacje dla programu uruchamiającego jądro (LILO, Grub)

/dev – zawierający pliki urządzeń

/lib – zawierający podstawowe biblioteki systemowe

/sys – zawierający pliki systemowe

/usr – zawierający pliki i programy przeznaczone dla użytkowników

/var – zawierający m.in. logi systemowe oraz pliki używane przez tzw. demony, czyli aplikacje realizujące usługi systemowe (obsługa poczty, drukarki)

/tmp – katalog plików tymczasowych

/home – katalog zawierający katalogi domowe użytkowników

Pomoc w systemie UNIX

man lub `info` – podręczniki ekranowe

`man -k` lub `apropos` – szybkie szukanie pomocy

opcja `--help` lub `-h`

klawisz TAB

wyszukiwarki (internet)

grupy dyskusyjne

`pl.comp.os.bsd`

`pl.comp.os.linux`

`pl.comp.os.unix`

Wydawanie komend

Komunikacja użytkownika z systemem UNIX następuje poprzez wydawanie komend interpretowanych oraz realizowanych przez powłokę (shell). Składnia polecenia w shellu jest następująca:

nazwa_polecenia opcje argumenty

nazwa_polecenia - komenda dostępna w systemie lub własny program użytkownika (skrypt).

opcje - jedna lub kilka opcji dostępnych z daną komendą. Opcje modyfikują i precyzują wykonanie danego polecenia. Najczęściej podaje się je jako grupy liter lub pojedyncze litery po znaku „-” .

argumenty - obiekty na których wykonywane są polecenia, np. pliki, urządzenia zewnętrzne.

Podstawowe polecenia

passwd - umożliwia zmianę hasła wpisywanego przez użytkownika przy logowaniu się do systemu. Na żądanie systemu należy podać kolejno stare hasło i dwukrotnie nowe.

quota -v - wyświetla wykorzystanie przydzielonego limitu dyskowego w obrębie konta.

clear - czyści ekran.

logout, exit, quit, CTRL + D - zakończenie pracy

date - aktualna data i godzina,

env – wyświetlenie informacji o środowisku (ang. environment), w którym pracuje użytkownik.

tty – podaje bieżący kod terminala,

w, who - wyświetla listę użytkowników aktualnie pracujących w systemie

who am i - wyświetla identyfikator użytkownika wypisującego tę komendę

id – wyświetla informacje o przynależności do grup użytkowników

uname – polecenie to powoduje wyświetlenie nazwy systemu operacyjnego. W połączeniu z opcją -a otrzymuje się kilka dodatkowych informacji na temat danego systemu operacyjnego.

Znaki ogólnego zastosowania

& - przenoszenie procesu w tło tj. uruchamianie dodatkowego okna obsługującego proces

>, >> - przekierowanie wyniku procesu,

< - pobranie danych do rozkazu,

| - rozdzielenie kilku poleceń zapisanych w jednym wierszu w tzw. Potok.

np:

```
ls > lista_plikow.txt
```

```
ls >> lista_plikow.txt
```

```
find . -name szukany_plik | grep szukane_slowo
```


Najważniejsze polecenia

ls – polecenie pozwala wylistować zawartość katalogu. Składnia polecenia jest następująca:

```
ls [opcje][nazwa_katalogu / zbioru(ów)]
```

Najczęściej stosowane *opcje* to:

- a wyświetlane są wszystkie zbiory (również te, których nazwa zaczyna się od znaku kropki).
- D wyświetlane są tylko katalogi
- l wyświetlanie informacji o poszczególnych zbiorach w danym katalogu w długim formacie (ang. long) – nazwy, atrybuty zbiorów, właściciel i grupa, czas modyfikacji, itp.

np.:

```
ls -al | more - wyświetla listę plików, dzieląc je na ekrany
```

pwd – służy do sprawdzenia w jakim katalogu użytkownik znajduje się w danym momencie (jaki katalog jest katalogiem bieżącym).

Najważniejsze polecenia

cd – zmiana katalogu (ang. change directory) – umożliwia poruszanie się po katalogach.

Składnia polecenia jest następująca:

`cd [opcje]`

`cd /` - przejście do katalogu podstawowego,

`cd ..` - przejście o jeden katalog w górę (w hierarchii drzewa katalogów),

`cd ../..` - przejście o dwa katalogi w górę,

`cd nazwakatalogu` - przejście do katalogu o wybranej nazwie

`cd` - przejście do katalogu domowego z dowolnego miejsca.

np.:

`cd mucha` - przejście do katalogu mucha znajdującego się w katalogu bieżącym

`cd /mucha` - przejście do katalogu mucha znajdującego się w katalogu głównym

Najważniejsze polecenia

find – wyszukiwanie pliku

```
find /wkatalogu -name szukana_nazwa_pliku
```

cat – wyświetla zawartość pliku o wyspecyfikowanej nazwie pliku

```
cat nazwa_pliku
```

more – filtr umożliwiający wyświetlanie zawartości pliku tekstowego na ekranie terminala (najczęściej monitorze komputera z którego łączymy się z serwerem). Umożliwia wstrzymanie wyświetlania po każdym zapełnieniu ekranu.

```
more nazwa_pliku
```

less – podobnie jak powyżej, umożliwia także przeglądanie wstecz.

```
less nazwa_pliku
```

wc – zliczanie liczby linii, słów lub znaków w pliku.

```
wc [opcje] nazwa_pliku
```

opcje:

-c – zliczanie znaków,

-l – zliczanie linii,

-w – zliczanie słów.

Najważniejsze polecenia

mkdir – polecenie to (skrót od „make directory”) pozwala na utworzenie nowego katalogu.

```
mkdir nazwa_katalogu
```

np.:

```
mkdir mucha - tworzy katalog o nazwie mucha w katalogu bieżącym
```

rm – polecenie rm (skrót od „remove”) pozwala na usunięcie zbioru(ów) lub drzewa katalogów.

```
rm [opcje] nazwa_zbioru
```

Najczęściej używane opcje to:

- i system żąda potwierdzenia usunięcia każdego zbioru
- f system nie żąda potwierdzenia usunięcia zbiorów które mają zabroniony zapis
- v w czasie usuwania są listowane usuwane zbiory
- R usuwanie całego drzewa katalogów łącznie z katalogami

np.:

```
rm MojZbior - usuwa zbiór o nazwie MojZbior z bieżącego katalogu jeżeli jest pusty  
rm -v katalog/* - usuwa wszystkie zbiory w katalogu katalog i listuje je
```

Najważniejsze polecenia

mv – polecenie mv (od ang. „move”) służy do przenoszenia lub zmiany nazwy zbiorów.

```
mv [opcje] zbiór_źródłowy miejsce_docelowe
```

np.:

```
mv jeden.txt katalog - kopiuje zbiór jeden.txt do katalogu katalog.
```

cp - kopiuj (ang. copy) - polecenie to służy do kopiowania zbioru lub drzewa katalogów.

```
cp [opcje] zbiór_zródłowy zbiór_docelowy
```

Najczęściej używane opcje to:

-v w trakcie kopiowania są listowane zbiory kopiowane.

-R kopiowanie drzewa katalogów. Parametr ten musi być użyty jeśli zbiorem źródłowym jest katalog.

-i podczas kopiowania system zażąda potwierdzenia, jeżeli nastąpi zapisanie kopiowanego zbioru na już istniejącym.

np:

```
cp jeden.txt dwa.txt -kopiuje zbiór jeden.txt na zbiór dwa.txt
```

```
cp -vRc * ../backup/ -kopiuje wszystkie zbiory i katalogi z katalogu bieżącego do katalogu backup znajdującego się o jeden poziom wyżej w stosunku do bieżącego i listuje nazwy kopiowanych zbiorów.
```

Najważniejsze polecenia

`ln` – tworzenie linku (dowiązania) do pliku. Nowy plik może znajdować się w zupełnie innym

katalogu niż plik macierzysty. Takie połączenie nazw z plikiem nosi nazwę linku, co można

rozpoznać przy wyświetleniu zawartości katalogu po oznaczeniu `l` (ang. link) znajdującym się na pierwszym miejscu opisu typu pliku.

Rozróżnia się linki twarde i linki symboliczne. Link twardy to nadanie innej nazwy istniejącemu plikowi (przez to link oraz oryginał stają się przez to nierozróżnialne), natomiast link symboliczny jest specjalnym typem pliku wskazującym na plik oryginalny poprzez jego nazwę (jest to wpis w katalogu wskazującym na nazwę oryginalnego pliku).

np.:

```
ln plik_jeden plik_dwa – stworzenie linku twardego
```

```
ln -s plik_jeden plik_dwa – stworzenie linku symbolicznego
```

Ochrona i prawa dostępu do pliku

Ze względu na to, że Unix jest systemem wielodostępnym, dostęp do każdego pliku w systemie jest chroniony poprzez system własności plików i praw dostępu. Każdy plik ma swojego właściciela i jest przypisany do grupy użytkowników. Prawa dostępu podzielone są na trzy grupy: dostęp dla właściciela, grupy użytkowników i pozostałych, a przydzielane prawa to:

r - (*read*) prawo czytania,
w - (*write*) prawo zapisu,
x - (*execute*) prawo wykonywania pliku.

W przypadku katalogu prawo zapisu oznacza, iż zawarte w nim pliki można usuwać i zmieniać.

Przykładowy układ praw dostępu do katalogu:

d | rwx | r-x | --x
typ pliku ~ właściciel ~ grupa ~ wszyscy

d - oznacza że dany zbiór jest katalogiem (ang. Directory)

l - oznacza że dany zbiór jest linkiem symbolicznym

- - oznacza zwykły plik

Ochrona i prawa dostępu do pliku

Do zmiany uprawnień użytkowników w stosunku do pliku czy katalogu służy polecenie **chmod**

(*change mode* - zmień tryb). Składnia polecenia jest następująca:

```
chmod ugo nazwa_pliku
```

gdzie: u określa właściciela (ang. user), g- grupę w ramach której pracuje dany użytkownik (ang. group), natomiast o –wszystkich innych (ang. other)

Prawa dostępu można podawać w dwojaki sposób:

- w postaci liczbowej z zastosowaniem zapisu ósemkowego, przy założeniu, że prawu odczytu r odpowiada wartość 4, prawu zapisu w - 2, a prawu wykonywania x – 1, brak danego prawa równoważny jest 0.

- w postaci specyfikacji: u [+ | -] [rwx] , g [+ | -] [rwx] , o [+ | -] [rwx] (dodajemy (znak +), lub odbieramy (znak -) dane prawo).

Oznacza to iż komendy:

```
chmod u=rwx,g=rx,o=r plik  
chmod 754 plik
```


Okresowe wykonywanie zadań - crontab

```
# Run hourly cron jobs at 47 minutes after the hour:  
47 * * * * /usr/bin/run-parts /etc/cron.hourly 1> /dev/null
```

```
# Run daily cron jobs at 3:40 every day:  
40 3 * * * /usr/bin/run-parts /etc/cron.daily
```

```
# Run weekly cron jobs at 4:30 on the first day of the week:  
30 4 * * 0 /usr/bin/run-parts /etc/cron.weekly
```

```
# Run monthly cron jobs at 4:20 on the first day of the month:  
20 4 1 * * /usr/bin/run-parts /etc/cron.monthly
```

zakresy liczbowe

1-3 - czyli wartości 1,2,3

0-10/2 - czyli wartość 0,2,4,6,8 i 10 (co druga wartość ze zbioru od 0 do 10)

1,2,5 - czyli wartości kolejno 1,2,5

*/2 - co 2 dozwolona wartość (np. w pierwszej kolumnie będzie to 0,2,4,6...56,58)

1-3,5,6 - czyli 1,2,3 oraz 5 i 6

Edytor vi / vim

Edytor vi jest powszechnie dostępny we wszystkich odmianach systemu UNIX. Wykorzystywany jest najczęściej do pisania skryptów i programów. Aby utworzyć lub edytować plik należy użyć wywołania **vi** z nazwą pliku jako argumentem.

vi nazwa_pliku

podstawowe funkcje edytora:

i - przejście w tryb edytowania (pisanie tekstu; przejście do następnego wiersza [Enter];poprawki tylko w trybie rozkazów)

<Esc> - przejście do trybu rozkazów

x - kasowanie znaku znajdującego się przy kursorze,
dw - kasowanie wyrazy w którym znajduje się kursor,
dd - kasowanie wiersza w którym znajduje się kursor,
:w – zapis pliku na dysk,
:q! - wyjście bez zapisu,
:x - wyjście z zapisaniem pliku,
:q – wyjście z edytora, jeśli plik nie był modyfikowany.

cd. w dodatkowej dokumentacji