

1. Powłoki, konfiguracja.

- Do komunikacji użytkownika z jądrem systemu operacyjnego służy *powłoka systemu (shell)*, w linuxie jest dostępnych kilka powłok;
 - The C shell (`/bin/csh`, często link do `/bin/tcsh`);
 - The TC shell (`/bin/tcsh`);
 - The Korn shell (`/bin/ksh`);
 - The Bourne shell (`/bin/sh`, często link do `/bin/bash`);
 - The Bourne again shell (`/bin/bash`);
- Różne powłoki można uruchamiać w trakcie pracy z systemem;

```
suse:~ # env | grep SHELL
SHELL=/bin/bash
suse:~ # csh
user has logged on :0 from local.
user has logged on pts/0 from :0.0.
user has logged on pts/1 from :0.0.
suse /root# pwd
/root
suse /root# exit
exit
```

- Powłoka uruchamiana zaraz po zalogowaniu użytkownika nosi nazwę **login shell**;
- Pliki konfiguracyjne powłoki **login shell**;
 - `/etc/profile` (plik dla całego systemu);
 - `~/.profile` (dodatkowa konfiguracja dla każdego użytkownika);
 - `/etc/bash.bashrc` (konfiguracja powłoki dla całego systemu, konfiguracja polecenia `ls` i aliasy);
 - `~/.bashrc` (konfiguracja użytkownika);
- W przypadku powłok uruchamianych z linii poleceń (**non-login shell**) konfiguracja jest pobierana z plików `/etc/bash.bashrc` i `~/.bashrc`;
- Aby załadować konfigurację po zmianach należy uruchomić pliki konfiguracyjne;

```
suse:~ # source ~/.bashrc
```

lub

```
suse:~ # . ~/.bashrc
```

- Klawisz [TAB] można używać do dokończania poleceń lub nazw plików;
- W pliku `~/.bash_history` jest zapamiętana historia wpisywanych poleceń, można ją wyświetlić poleceniem `history` lub wyświetlać kolejne polecenia kursorami;

2. Zmiana użytkownika.

- Za pomocą polecenia `su` można przełączyć się na innego użytkownika, np. `root`, opcja `-` (myślnik, minus) lub `-l` powoduje uruchomienie powłoki jako **login shell**;

```
user@suse:~> whoami
user
user@suse:~> id
uid=1000(user) gid=100(users) groups=100(users),16(dialout),33(video)
user@suse:~> su -
Password:
suse:~ # whoami
root
suse:~ # id
uid=0(root) gid=0(root) groups=0(root),105(sfcb)
suse:~ # su - user
Directory: /home/user
Mon Mar 25 12:43:03 CET 2013
user@suse:~> whoami
user
user@suse:~> id
uid=1000(user) gid=100(users) groups=100(users),16(dialout),33(video)
user@suse:~> exit
logout
suse:~ # whoami
root
suse:~ # exit
logout
user@suse:~> whoami
user
user@suse:~>
```

3. Zmienne.

- Powłoki pozwalają na definiowanie zmiennych, są one potrzebne do poprawnego działania samej powłoki jak i innych programów, zdefiniowane zmienne można wyświetlić poleceniem `env`;

```
suse:~ # env
LESSKEY=/etc/lesskey.bin
NNTPSERVER=news
INFODIR=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/share/man:/usr/local/man
HOSTNAME=suse
XKEYSYMDB=/usr/share/X11/XKeysymDB
HOST=suse
SHELL=/bin/bash
TERM=xterm
PROFILEREAD=true
HISTSIZE=1000
MORE=-s1
USER=root
...
```

- Ważniejsze zmienne;

PATH	ścieżka przeszukiwania poleceń
HOME	katalog domowy użytkownika
USER	nazwa aktualnego użytkownika
HOSTNAME	nazwa komputera
SHELL	powłoka systemu

- Pojedyncze zmienne można wyświetlać poleceniem **echo**;

```
suse:~ # echo $HOME
/root
suse:~ # echo $SHELL
/bin/bash
```

- Zmienne można definiować przypisaniem do nich wartości, w przypadku znaków specjalnych należy użyć cudzysłowów;

```
suse:~ # VAR=value
suse:~ # echo $VAR
value
suse:~ # NUMVAR=69
suse:~ # echo $NUMVAR
69
suse:~ # TEXTVAR="text line"
suse:~ # echo $TEXTVAR
text line
```

4. Aliasy (skrót).

- Aliasy pozwalają na definiowanie skrótów często wykorzystywanych poleceń, w systemie prawdopodobnie będą już zdefiniowane jakieś skrót;

```
suse:~ # alias md
alias md='mkdir -p'
suse:~ # alias dir
alias dir='ls -l'
suse:~ # type -a ls
ls is aliased to `ls $LS_OPTIONS'
ls is /bin/ls
```

- Aliasy są definiowane poleceniem **alias** (zwykle w plikach konfiguracyjnych powłoki), i mogą być usuwane w trakcie pracy systemu za pomocą polecenia **unalias**;

```
suse:~ # alias dir
alias dir='ls -l'
suse:~ # dir
total 92
-rw----- 1 root root 103 Mar 14 17:25 .Xauthority
-rw----- 1 root root 9325 Mar 25 12:43 .bash_history
drwxr-xr-x 3 root root 4096 Mar 14 17:26 .config
...
suse:~ # unalias dir
suse:~ # dir
bin dir1 dir3      file1  file3
dir dir2 file.txt file2  inst-sys
suse:~ # alias dir="ls -l"
```

```
suse:~ # dir
total 92
-rw----- 1 root root  103 Mar 14 17:25 .Xauthority
-rw----- 1 root root 9325 Mar 25 12:43 .bash_history
drwxr-xr-x 3 root root 4096 Mar 14 17:26 .config
...
```

- Aliasy zdefiniowane podczas pracy są aktualne dla sesji;

```
suse:~ # ps
  PID TTY          TIME CMD
 4127 pts/0    00:00:00 dbus-launch
11496 pts/0    00:00:00 su
11499 pts/0    00:00:00 bash
11595 pts/0    00:00:00 ps
suse:~ # alias ps="echo Hello\!"
suse:~ # ps
Hello!
suse:~ # bash
suse:~ # ps
  PID TTY          TIME CMD
 4127 pts/0    00:00:00 dbus-launch
11496 pts/0    00:00:00 su
11499 pts/0    00:00:00 bash
11596 pts/0    00:00:00 bash
11607 pts/0    00:00:00 ps
suse:~ # exit
exit
suse:~ # ps
Hello!
suse:~ # unalias ps
suse:~ # ps
  PID TTY          TIME CMD
 4127 pts/0    00:00:00 dbus-launch
11496 pts/0    00:00:00 su
11499 pts/0    00:00:00 bash
11608 pts/0    00:00:00 ps
```

- Aby były dostępne dla nowych sesji powinno się je definiować w plikach konfiguracyjnych powłoki, np. `~/bash.rc`;

5. Strumienie i przekierowanie.

- Strumienie są kanałami, przez które aplikacja kontaktuje się z użytkownikiem i środowiskiem w jakim pracuje;
 - **standard input (stdin, 0)** - standardowe wejście, zwykle klawiatura;
 - **standard output (stdout, 1)** - standardowe wyjście, zwykle monitor;
 - **standard error (stderr, 2)** - kanał błędów, zwykle monitor;
- Każdy strumień może być przekierowany;

<	przekierowanie standardowego wejścia
>, 1>	przekierowanie z nadpisaniem standardowego wyjścia
>>, 1>>	przekierowanie wyjścia (z dołączaniem)
2>, 2>>	przekierowanie błędów

```
suse:~ # ls /opt /dummy
ls: cannot access /dummy: No such file or directory
/opt:
VBoxGuestAdditions-4.2.6  kde3
suse:~ # ls /opt /dummy 2> /dev/null
/opt:
VBoxGuestAdditions-4.2.6  kde3
```

- Strumień może być przekierowany do pliku (nadpisany lub dołączony);

```
suse:~ # ls /opt /dummy > file 2> /dev/null
suse:~ # cat file
/opt:
VBoxGuestAdditions-4.2.6
kde3
suse:~ # ls /opt /dummy >> file 2> /dev/null
suse:~ # cat file
/opt:
VBoxGuestAdditions-4.2.6
kde3
/opt:
VBoxGuestAdditions-4.2.6
kde3
```

- Można skierować standardowe wejście do pliku;

```
suse:~ # echo "Hello,
>
> how are you?" > greetings
suse:~ # ls -l greetings
-rw-r--r-- 1 root root 21 Mar 25 14:18 greetings
suse:~ # cat greetings
Hello,
how are you?
```

- Standardowe wyjście polecenia może być wejściem innego polecenia, służy do tego symbol | (**pipe**);

```
suse:~ # ls -l / | wc -l
22
suse:~ # ls -l /etc | less
```

- Wyjście z wielu poleceń można łączyć za pomocą nawiasów;

```
suse:~ # (id; ls -l /var) > output
suse:~ # cat output
uid=0(root) gid=0(root) groups=0(root),105(sfcb)
total 52
drwxr-xr-x  2 root root 4096 May  5  2010 X11R6
drwxr-xr-x 11 root root 4096 Mar 12 13:37 adm
drwxr-xr-x 15 root root 4096 Mar 12 16:48 cache
drwxr-xr-x  2 root root 4096 May  5  2010 crash
drwxr-xr-x  2 root root 4096 May  5  2010 games
drwxr-xr-x 43 root root 4096 Mar 24 20:10 lib
drwxrwxr-t  6 root uucp 4096 Mar 25 10:16 lock
drwxr-xr-x 12 root root 4096 Mar 18 16:23 log
lrwxrwxrwx  1 root root   10 Mar 12 16:01 mail -> spool/mail
drwxr-xr-x  2 root root 4096 May  5  2010 opt
```

```
drwxr-xr-x 18 root root 4096 Mar 25 13:57 run
drwxr-xr-x 12 root root 4096 Mar 12 16:38 spool
drwxrwxrwt 3 root root 4096 Mar 25 13:57 tmp
drwxr-xr-x 3 root root 4096 Mar 12 16:40 yp
```

- Każde polecenie zwraca kod mówiący stanie po wykonaniu, wartość 0 oznacza sukces, wartości większe oznaczają błąd, wartość ta jest przechowywany w zmiennej ?;

```
suse:~ # ls /
app boot etc      home lost+found mnt proc sbin      srv tmp var
bin dev  export lib  media          opt root selinux sys  usr
suse:~ # echo $?
0
suse:~ # ls dummy
ls: cannot access dummy: No such file or directory
suse:~ # echo $?
2
```

- Zwracany kod może być użyty do kontrolowania wykonania kolejnych poleceń;

<code>command1 && command2</code>	polecenie <i>command2</i> zostanie uruchomione jeśli polecenie <i>command1</i> wykona się bez błędów
<code>command1 command2</code>	polecenie <i>command2</i> zostanie uruchomione jeśli polecenie <i>command1</i> wykona się z błędami

```
suse:~ # ls dummy && ls /
ls: cannot access dummy: No such file or directory
suse:~ # ls dummy || ls /
ls: cannot access dummy: No such file or directory
app boot etc      home lost+found mnt proc sbin      srv tmp var
bin dev  export lib  media          opt root selinux sys  usr
```