Współczesne systemy komputerowe
Powłoka i zmienne

Powłoki, konfiguracja

- Do komunikacji użytkownika z jądrem systemu operacyjnego służy *powłoka systemu* (*shell*), w linuksie 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 (jeśli są zainstalowane)

```
root@debian:~# env | grep SHELL
SHELL=/bin/bash
```

```
root@debian:~# sh
# pwd
/root
# logout
```

- Powłoka uruchomiana 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, 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

```
root@debian:~# source ~/.bashrc
```

lub

```
root@debian:~# . ~/.bashrc
```

- Klawisza **[TAB]** można używać do dokańczania poleceń lub nazw plików
- W pliku **~/.bash_history** jest zapamiętana historia wpisywanych poleceń, można ją wyświetlić poleceniem **history**
- Kursorami można wyświetlać ostatnio wpisane polecenia

Zmiana użytkownika

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

```
foo@debian:~$ whoami
foo
```

```
foo@debian:~$ id
uid=1001(foo) gid=1001(foo) groups=1001(foo)
foo@debian:~$ su -
Password:
```

```
root@debian:~# whoami
root
```

```
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:~# su - foo
```

```
foo@debian:~$ whoami
foo
```

```
foo@debian:~$ id
uid=1001(foo) gid=1001(foo) groups=1001(foo)
```

```
foo@debian:~$ logout
```

```
root@debian:~# whoami
root
```

```
root@debian:~# logout
```

```
foo@debian:~$ whoami
foo
```

Zmienne

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

```
root@debian:~# env
LANG=en_US.UTF-8
DISPLAY=:1
COLORTERM=truecolor
USER=root
PWD=/root
HOME=/root
MAIL=/var/mail/root
SHELL=/bin/bash
TERM=xterm-256color
SHLVL=1
LOGNAME=root
XAUTHORITY=/run/user/1001/gdm/Xauthority
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
_=/usr/bin/env
```

- Ważniejsze zmienne

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

- Pojedyncze zmienne można wyświetlić poleceniem **echo**

```
root@debian:~# echo $HOME
/root
```

```
root@debian:~# echo $SHELL
/bin/bash
```

- Zmienne można definiować przez przypisanie do nich wartości, w przypadku znaków należy użyć cudzysłowów

```
root@debian:~# VAR=value
```

```
root@debian:~# NUMVAR=69
```

```
root@debian:~# TEXTVAR="text line"
```

```
root@debian:~# echo $VAR
value
```

```
root@debian:~# echo $NUMVAR
69
```

```
root@debian:~# echo $TEXTVAR
text line
```

Aliasy

- Aliasy pozwalają na definiowanie skrótów często wykorzystywanych poleceń

```
root@debian:~# alias fulldir="ls -al"
```

```
root@debian:~# type -a fulldir
fulldir is aliased to `ls -al'
```

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

```
root@debian:~# alias fulldir
alias fulldir='ls -al'
```

```
root@debian:~# fulldir
total 1576
drwx------ 11 root root    4096 Mar  5 04:31 .
drwxr-xr-x 22 root root    4096 Feb 28 07:05 ..
-rw-------  1 root root    4154 Mar  5 04:52 .bash_history
...
```

```
root@debian:~# unalias fulldir
```

```
root@debian:~# fulldir
-su: fulldir: command not found
```

- Aliasy zdefiniowane podczas pracy są aktualne dla sesji

```
root@debian:~# ps
  PID TTY          TIME CMD
 6847 pts/0    00:00:00 su
 6848 pts/0    00:00:00 bash
 6879 pts/0    00:00:00 ps
```

```
root@debian:~# alias ps="echo Hello\!"
```

```
root@debian:~# ps
Hello!
```

```
root@debian:~# bash
```

```
root@debian:~# ps
  PID TTY          TIME CMD
 6847 pts/0    00:00:00 su
 6848 pts/0    00:00:00 bash
 6880 pts/0    00:00:00 bash
 6881 pts/0    00:00:00 ps
```

```
root@debian:~# exit
exit
```

```
root@debian:~# ps
Hello!
```

```
root@debian:~# unalias ps
```

```
root@debian:~# ps
  PID TTY          TIME CMD
 6847 pts/0    00:00:00 su
 6848 pts/0    00:00:00 bash
 6882 pts/0    00:00:00 ps
```

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

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

```
root@debian:~# ls /opt /dummy
ls: cannot access '/dummy': No such file or directory
/opt:
VBoxGuestAdditions-4.3.12
```

```
root@debian:~# ls /opt /dummy 2> /dev/null
/opt:
VBoxGuestAdditions-4.3.12
```

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

```
root@debian:~# ls /opt /dummy > file 2> /dev/null
```

```
root@debian:~# cat file
/opt:
VBoxGuestAdditions-4.3.12
```

```
root@debian:~# ls /opt /dummy >> file 2> /dev/null
```

```
root@debian:~# cat file
/opt:
VBoxGuestAdditions-4.3.12
/opt:
VBoxGuestAdditions-4.3.12
```

- Można skierować standardowe wejście do pliku

```
root@debian:~# echo "Hello,
>
> how are you?" > greetings
```

```
root@debian:~# cat greetings
Hello,

how are you?
```

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

```
root@debian:~# ls -l | wc -l
16
```

```
root@debian:~# ls -l /etc | less
```

- Wyjście z wielu poleceń można łączyć za pomocą nawiasów

```
root@debian:~# (id; ls -l /var) > output
```

```
root@debian:~# cat output
uid=0(root) gid=0(root) groups=0(root)
total 36
drwxr-xr-x  2 root root  4096 Mar  5 03:52 backups
drwxr-xr-x 16 root root  4096 Feb 28 12:16 cache
drwxr-xr-x 47 root root  4096 Dec 19 06:24 lib
drwxrwsr-x  2 root staff 4096 Jul 13  2017 local
```

```
lrwxrwxrwx  1 root root     9 Aug 18  2017 lock -> /run/lock
drwxr-xr-x  9 root root  4096 Mar  5 03:53 log
drwxrwsr-x  2 root mail  4096 Aug 18  2017 mail
drwxr-xr-x  2 root root  4096 Aug 18  2017 opt
lrwxrwxrwx  1 root root     4 Aug 18  2017 run -> /run
drwxr-xr-x  7 root root  4096 Aug 18  2017 spool
drwxrwxrwt  9 root root  4096 Mar  5 03:53 tmp
```

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

```
root@debian:~# ls /
bin   etc         initrd.img.old  media  proc  sbin  tmp  vmlinuz
boot  home        lib             mnt    root  srv   usr  vmlinuz.old
dev   initrd.img  lost+found      opt    run   sys   var
```

```
root@debian:~# echo $?
0
```

```
root@debian:~# ls dummy
ls: cannot access 'dummy': No such file or directory
```

```
root@debian:~# echo $?
2
```

- Zwracany kod może być użyty do kontrolowania wykonania kolejnych poleceń

command1 && *command2*	polecenie *command2* zostanie uruchomione jeśli polecenie *command1* wykona się bez błędów
command1 || *command2*	polecenie *command2* zostanie uruchomione jeśli polecenie *command1* wykona się z błędami

```
root@debian:~# ls dummy && ls /
ls: cannot access 'dummy': No such file or directory
```

```
root@debian:~# ls dummy || ls /
ls: cannot access 'dummy': No such file or directory
bin   etc         initrd.img.old  media  proc  sbin  tmp  vmlinuz
boot  home        lib             mnt    root  srv   usr  vmlinuz.old
dev   initrd.img  lost+found      opt    run   sys   var
```