Estonian Information Technology College

User environment and processes

Operating systems I800

Edmund Laugasson
edmund.laugasson@itcollege.ee

There has been used materials from Margus Ernits, Katrin Loodus when creating current slides.

User environment settings

- When user will enter into system there will be run scripts that initialize user environment (there will be user session created)
  - environment variables will be set
  - shell aliases will be set, e.g. in ~/.bashrc file:
    - alias ls='ls –color=auto'
    - alias grep='grep --color=auto'
    - alias fgrep='fgrep --color=auto'
    - alias egrep='egrep –color=auto'
    - alias ll='ls -alF'
    - alias la='ls -A'
    - alias l='ls -CF'
  - functions
- there will be initialized the file /etc/profile meant for all users
- also user personal preferences in home folder will be initialized
  - ~/.profile
  - ~/.bash_profile
  - ~/.bashrc
- user can change personal settings

in ~/.bashrc there is written:

```
if [ –f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
```

... so the correct file for bash aliases would be the mentioned ~/.bash_aliases – this could be copied also to /etc/skel/ in order to make it available for all new users

Bash shell configuration files

- **.bash_profile** user environment individual settings. There you can change default settings and add new ones. Will be run when user log in.
- **.bash_login** will be run only when user log in. When *.bash_profile* do not exist, this file will be read
- **.bashrc** will be run e.g. by opening a terminal window (interactive shell)
  .bash_aliases – will be run e.g. by opening a terminal window, short commands file.
- **.bash_history** here are the history of entered commands up to a values defined in ~/.bashrc with parameters HISTSIZE=1000 and HISTFILESIZE=2000 (default values). Check also the utility *history (man history)*.
- **.bash_logout** contains a commands entered while logging out
- **/etc/profile** similar to the file *.bash_profile*, but applies globally
- **/etc/profile.d** files in that folder will be treated similarly with **/etc/profile** file. When you need define your functions then /etc/profile.d/ would be the good place.
- Why there are often *.d folders used?
  http://unix.stackexchange.com/questions/4029/what-does-the-d-stand-for-in-directory-names
- More information
  - https://en.wikipedia.org/wiki/Bash_(Unix_shell)
  - https://help.ubuntu.com/community/Beginners/BashScripting
  - http://tldp.org/LDP/abs/html/ - Advanced Bash Scripting Guide
  - in Estonian https://wiki.itcollege.ee/index.php/BASH_shell

Running session scripts

- The *~/.bash_aliases* and *~/.bashrc* are treated as session scripts and can be run by using command *source* or just dot with space, e.g.
  - **`source ~/.bash_aliases`**
  - **`. ~/.bash_aliases`**
- ~/.bashrc will be read every time you open new terminal (shell)
- /etc/profile and .profile will be run every time the user will enter into system
- when you change the content of *.profile*, then in order to apply changes you need either relogin or run the session script: *source ~/.profile* (. ~/.profile)

Alias – short command

- every user can define short commands, aliases
- permanent aliases are defined in ~/.bash_aliases file, because in ~/.bashrc there is written:

```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
```

- some aliases for ls command
  - alias ls='ls --color=auto'
  - alias ll='ls -l'
  - alias la='ls -A'
  - alias l='ls -CF'
- in English http://tldp.org/LDP/abs/html/aliases.html
- in Estonian https://wiki.itcollege.ee/index.php/Alias_bash_shellis

Environment variables

- **USER** – username
- **PATH** – folder names from which the system will search program files that user can run without referring full directory path
  - e.g. add new folder into current path for user student
    - nano ~/.bashrc
    - export PATH=$PATH:/home/student/bin
    - source ~/.bashrc
- **HOME** – user home folder
- **SHELL** – user shell
- **EDITOR** – text editor used by user
- **HOSTNAME** – a computer name in network stack
- **env** is used to see environment variables
- ***declare*** will show extended list of environment variables

Environment variables 2

- ***export*** command can be used to set up environment variable in Bash shell
  - **export** variable=value
  - **export** variable2=”longer value of this variable”
- in C shell
  - **setenv** variable value
- in MS Windows
  - **set** variable=value

Processes

- creation
- input/output and errors
- redirection
- communication between processes
- signals
- jobs

Processes

- The process is a started program that has separated resources from processor and memory (RAM)
- The process has a PID (*process ID*)
- The process can start other processes
  - The process that started another process is called *parent process*
- Processes will establish a process tree that has in peak the first process, in Linux-like systems **init**

Process table

- Operating system keeps track about processes and resources
- Data will be kept in process table
- The process tree can be displayed (Linux/Unix)

  pstree
- The process table can be displayed (Linux/Unix)

  ps -ef
- more information
  - in English http://www.linfo.org/ps.html
  - in Estonian https://wiki.itcollege.ee/index.php/Ps
  - man ps
- more choices (needs to be installed): htop (more colorful), atop

Processes

- Sharing resources between processes is done by operating system kernel
- Process can be in following states
  - *created*
  - *running*
  - *waiting*
    - also swapped and waiting
  - *blocked*
    - also swapped and waiting
  - *terminated*
  - *zombie – process without parent process*

Communication between processes

- processes can exchange data beween each other
  - using shared files
  - using shared memory
  - using shared sockets
  - by sending signals
  - using semafors (flags)
  - using *pipe*

Processes

- Processes in Linux-like systems
  - standard input ***STDIN***
  - standard output ***STDOUT***
  - error output ***STDERR***
- Process output can be redirected into another process input using pipe – vertical line |
  - ***ps -ef | less***
    - the *ps* output will be redirected into less input
  - ***ps -ef | grep bash | wc***
    - wc will show accordingly: number of lines, words, bytes
- when searching help then redirecting long outputs into appropriate web service would be useful
  - https://help.ubuntu.com/community/Pastebinit

Redirecting a file

- Process input can be taken from file using redirection sign **<**
  - ***cat < /dev/urandom***
    - the program cat input will be taken from random number generator
- process output can be redirected into file using **>** or **>>**
  - ***cat < /dev/urandom > random-numbers.dat***
    - the program cat output will be written into file random-numbers.dat
    - .... and *random-numbers.dat* will be overwritten
  - cat < /dev/urandom >> random-numbers.dat
    - >> will add data to the end of file

Redirecting error output

- when there is required that a program will not write into standard output then we can redirect output e.g. into device /dev/null
  - cat </dev/zero > /dev/null
- error output will be not redirected and for that there can be used 2>&1 in the end of the command(s)
- ./do-not-want-to-know > /dev/null 2>&1
  - error output will be redirected to same place as standard output
- more explanations at https://linuxjourney.com/lesson/stderr-standard-error-redirect

https://en.wikipedia.org/wiki//dev/zero

https://en.wikipedia.org/wiki/Null_device

Signals

- There can be sent signals to the processes
- The process will handle received signals
  - different signals have different influences to the processes
  - process may ignore some signals
- Signals have numerical labels and the short names
- sending a signal can be done by using a ***kill*** command, man kill; in Estonian https://wiki.itcollege.ee/index.php/Kill

Signals 2

- some signals
  - **SIGHUP 1** process freeze or dying, can be used to reload configuration, e.g. reopen log files
  - **SIGABRT 6** Abort, generates a core file to process the data in the memory
  - **SIGKILL 9** force to remove resources from processes, as a last step
  - **SIGPIPE 13** Pipe down (there is no sense to write, because nobody read)
  - **SIGTERM 15** Process polite foreclosure, default, and the safest way to shut down the process
  - **SIGUSR1 30,10,16** The user (programmer) defined by the signal1
  - **SIGUSR2 31,12,17** User-defined signal2
- PID view by application name: **pidof <application>**
  - **ps -ef | grep <application>**

Signals 3

- To send a command signal to the process takes place ***kill***
  - **kill <pid1> <pid2> ….**
  - **kill -9 3242**
    - termination signal -9 (kill) sending to the process 3242
  - **kill -TERM 9588**
    - termination signal -15 (term) sending to the process 9588
- The signals SIGKILL and SIGSTOP can not be ignored or treated by the program itself
- closing with force using precise process name
  - **killall firefox** (by default SIGTERM 15)
  - **killall -15 firefox** (nice closing)
  - **killall -9 firefox** (closing with force)
  - man killall
  - in Estonian https://wiki.itcollege.ee/index.php/Killall

Jobs

- Sometimes we would like to put job into background
  - ./program &
  - when needed to create a file with & in name then use apostrophes or escape sign \
    - e.g. ***touch*** '***file&***' or ***touch file\&***
- overview of programs working in background
  - jobs
- program working in terminal can be sent temporarily to background by using CTRL+Z (SIGSTOP) and terminate withCTRL+C (SIGINT)
- http://superuser.com/questions/262942/whats-different-between-ctrlz-and-ctrlc-in-unix-command-line

Jobs 2

- bring back to front
  - ***fg <job no>***
- to background
  - ***bg <job no>***
- close with force (kill)
  - ***kill %<job no>***
  - ***kill %%*** *(kill last job)*
- see what processes are connected with process:
  - ***pgrep ssh***
  - ***pgrep -u root ssh***

for trying...

- install a program cowsay
- run the following lines and see results (and created files)
  - ***sudo apt install cowsay***
  - ***cowsay mooo***
  - ***cowsay -f sheep maaa > sheep.txt***
  - ***cowsay What sheep >> sheep.txt***
- man cowsay
- in Estonian https://wiki.itcollege.ee/index.php/Cowsay

Questions?

Thank you for your attention!