UNIX and Linux – An Introduction

CHARMM – CERM/PROTEO workshop
Condordia University, March 2010

Olivier Fisette
<olivier.fisette.1@ulaval.ca>

Département de biochimie
et de microbiologie

Université Laval

UNIX – An operating system

- The original UNIX is an operating system developed in 1969 at the AT&T Bell Labs
- It implemented revolutionnary ideas
  - Concurent users and tasks
  - Remote access and networking
  - File orientation
- UNIX philosophy
  - Many small tools combine to perform a task

UNIX - An operating system family

- Many vendors developed derivatives and clones
  - HP UNIX
  - Sun Solaris
  - Mac OS X
- Academic and free projects did the same
  - BSD (University of Berkeley, California)
  - Minix
  - Linux
- A family of UNIX-like operating systems

UNIX - A standardised OS

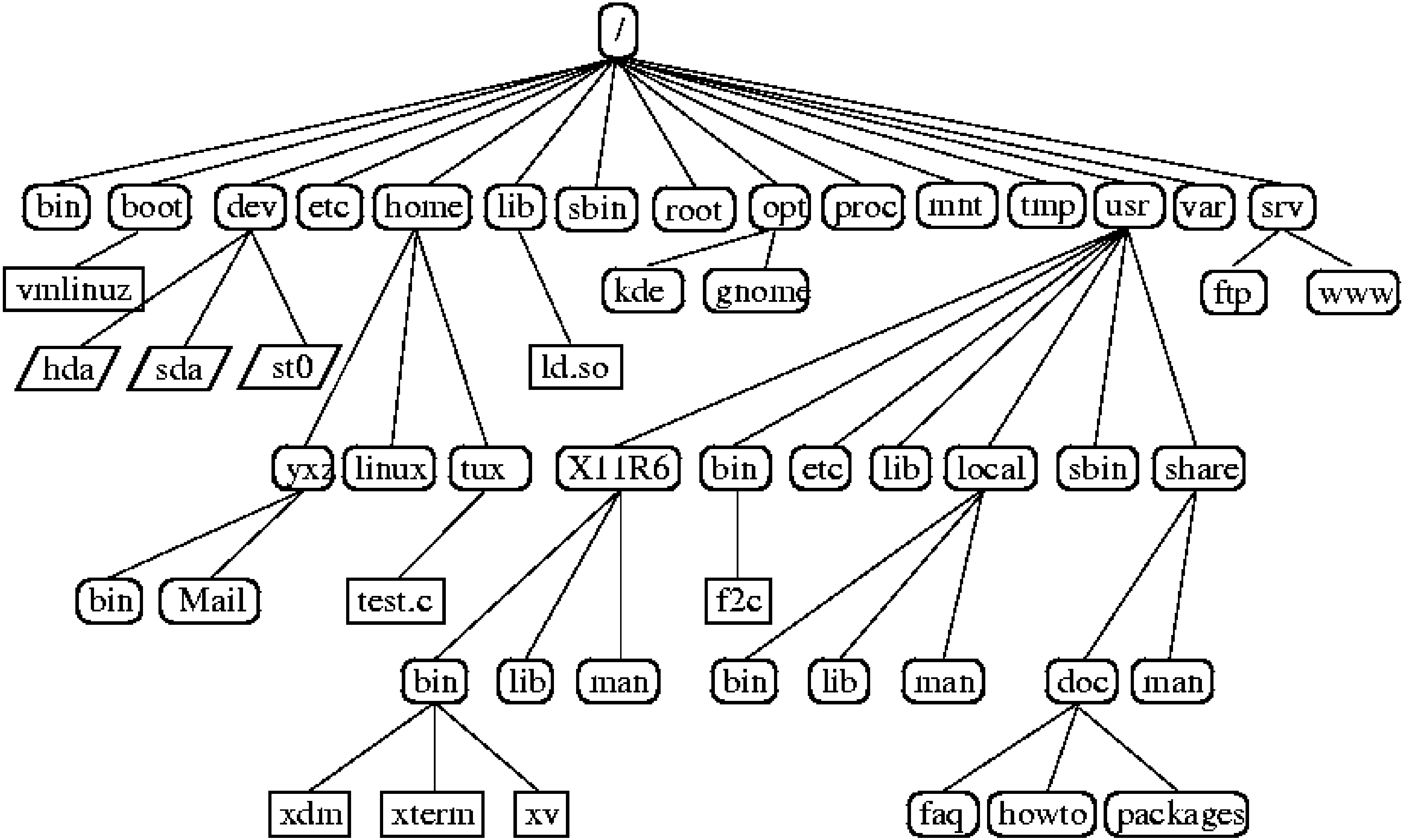
- The POSIX standard by the Open Group defines what is UNIX today
- UNIX and UNIX-like are generic terms for any operating system that is reasonably compatible with the standard
- What you learn on Linux or another UNIX is transferable

First steps with UNIX

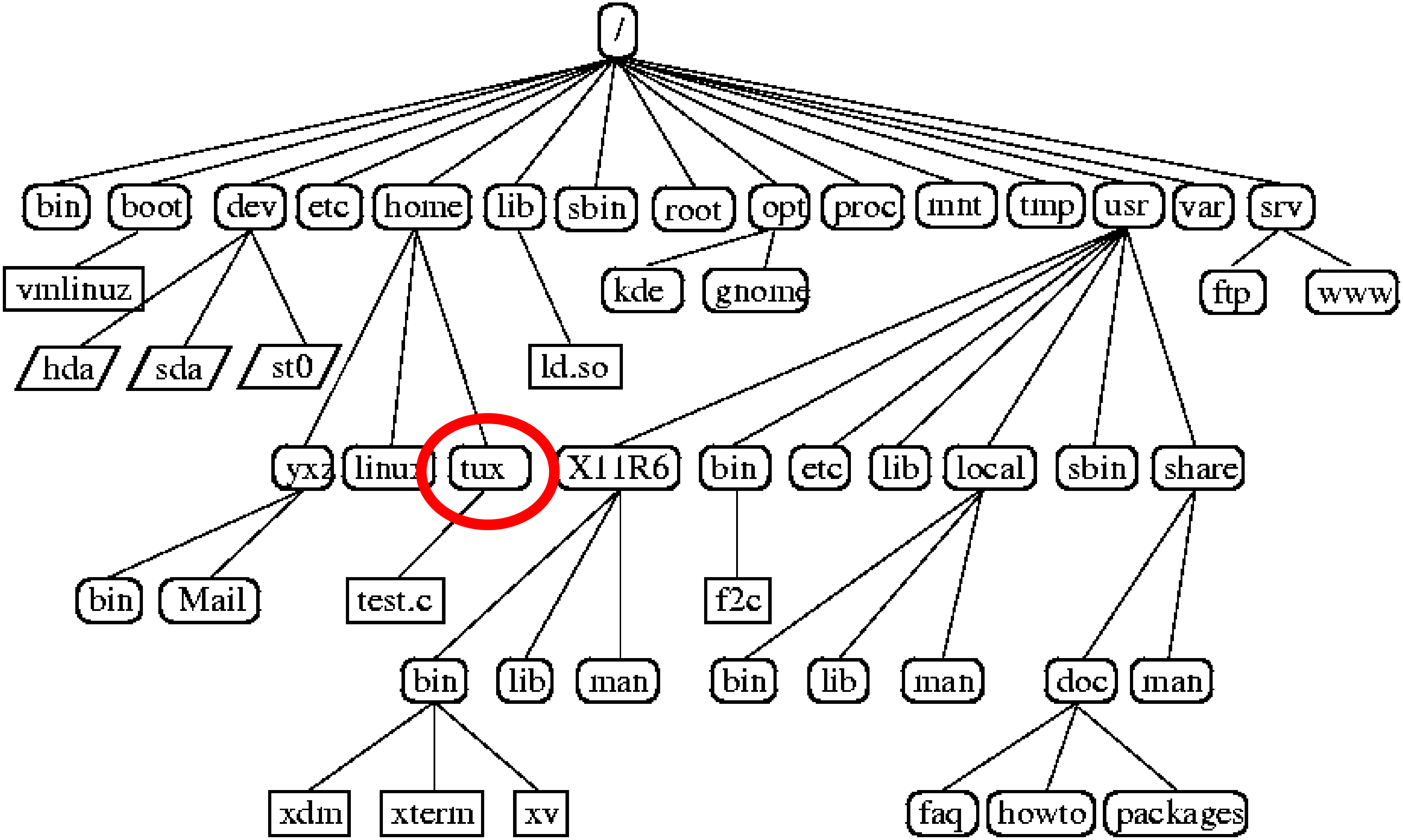
Ex. 1

- Open a terminal application
- Meet the shell (command-line interpreter)
  - UNIX is a text-based operating system
  - The Linux shell is bash
  - A graphical interface is built on top of the basic OS
- Type a few commands
  - `date, whoami, who, ps, ls, pwd`

The UNIX filesystem



The UNIX file system



Files and paths

- Everything in UNIX is a file
  - Even directories are specialised files
  - Files are organised in a hierarchical fashion
- A path is the address of a file
- Paths can be absolute...
  - `/home/tux/test.c`
- ... or relative (to /home/tux in these examples)
  - `test.c`
  - `../../etc/`

File management in UNIX

Ex. 2

- A variety of commands are used to create, delete, show and manipulate files in UNIX
- Download the exercice archive

```
– wget www.proteo.ca/ws-2010-03.tar.bz2
```

- Unpack the archive

```
– tar xjf ws-2010-03.tar.bz2
```

- Check the newly uncompressed files

```
– ls
```

- Files are in the `ws-2010-03` directory

File management in UNIX

Ex. 2

- Basic file management commands:
  - `pwd` : Show the working directory
  - `cd` : Change the working directory
  - `ls [dir]` : Show directory contents
  - `cat <file>` : Show file contents
  - `touch <file>` : Create an empty file
  - `mkdir <dir>` : Create an empty directory
  - `rm <file>` : Delete a file (except a directory)
  - `rmdir <dir>` : Delete an empty directory

File management in UNIX

Ex. 3

- More basic file management
  - `cp [file1] [file2]` : Copy a file
  - `cp -r [file1] [file2]` : Copy a directory
  - `mv [file1] [file2]` : Move a file
  - `mv [file] [directory]` : Move a file inside a directory
  - `rm -r [file]` : Remove a directory
  - `ls -l -a -h` (`ls -lah`) : Detailed, complete, readable file listing
  - `less [file]` : Show a file, one page at a time
  - `nano [file]` : Edit a file

Path expansion in bash

Ex. 4

- Several characters can be used to match one or more paths in bash
  - ? Matches any single character
  - * Matches any string of characters (or nothing)
  - ~ Expanded to your home directory
- Also remember the standard UNIX characters for special directories
  - .. Parent directory
  - . The current directory

Getting help with UNIX and bash

Ex. 5

- Manual pages are available for UNIX commands.
  - `man [program]`
- Help is available for bash built-in commands
  - `help [command]`
- Unknown files can be identified
  - `file [file]`

Working efficiently with bash

Ex. 6

- Key bindings for command edition
  - Ctrl+A, Ctrl+E : Go to start or end of line
  - Ctrl+B, Ctrl+F : Move one character right or left
  - Ctrl+P, Ctrl+N : Previous or next command in history
- Using completion
  - Tab : Attempt to complete the current word
  - Tab+Tab : Attempt to complete the current word, then show possible choices
- Other key bindings
  - Ctrl+D : Exit shell (like the exit command)

Text manipulation

Ex. 7

- UNIX is very much text oriented; a wide array of commands is available to analyse and manipulate text
    - `head [file]` : Show the first lines of a text
    - `tail [file]` : Show the last lines of a text
    - `grep [pattern] [file]` : Search for matching lines in a text
        - grep supports regular expressions
        - ^ and $ match beginning and end of lines
        - Enclose RE and expressions with whitespace characters in 'quotes'
    - `sort [file]` : Sort lines of text alphabetically
- These are UNIX filter commands

Communication channels

Ex. 8

- There are three standard communication channels in UNIX
  - Standard input (stdin) : What you type
  - Standard output (stdout) : Program or command output
  - Standard error (stderr) : Warnings and error messages from commands
- The last two are both shown on the terminal
  - They can be separated
- Filter commands operate on files or on stdin

Communication channels

- Input and output redirection
- Output redirection sends the output of a command to a file
  - `[command] > [file]`
  - Standard error is still shown on screen (but not with >&)
  - File is overwritten (but not with >>)
- Input redirection sends the contents of a file to the input of a command
  - `[command] < [file]`
  - Seldom used, since UNIX filters accept files as arguments

Communication channels

Ex. 10

- UNIX pipes send the output of a command to the input of another command
  - `[command1] | [command2]`
  - They can be chained and combined with redirections
- Using only filters, redirections and pipes, one can achieve powerful text manipulation

Process management

- UNIX allows many programs to run simultanously
- Commands are available to list and manage processes (running programs)
  - `ps` : List user processes in the current session
  - `ps -ef` : List all processes in the system
- Each process has an identifier (PID) and a parent
- Bash keeps a list of processes started within the current shell (jobs) for convenient process management

Process management

- Terminate process : `kill [pid]`
  - Or with a bash task: `kill %[job]`
- Interactive process management: `top`
  - Use q and k to quit and terminate process, respectively
- Run command with a different priority
  - `nice - n [i] [command]`
  - Where -20 is the lowest priority and 19 the highest
- Change process priority
  - `renice -n [i] [pid]`

Process management

- Run process in background
  - `[command] &`
- Job management key-bindings
  - Ctrl+C : Terminate foreground process
  - Ctrl+Z : Pause foreground process and send to background
- Other job management commands
  - `bg %[job]` : Restart process in background
  - `fg %[job]` : Restart process in foreground

Shell scripts

- A bash shell script is a file containing bash commands
- The commands are excuted when the script is run
- Anything one can type in an interactive shell can also be put in a bash script
- Bash script have many usages:
  - Gluing together individual UNIX tools
  - Reusing complex commands
  - Automating tasks

My First Shell Script™

```
#!/bin/bash

date
whoami

echo 'This is my first shell script!'
```

Running a shell script

- In UNIX, each file has an owner and permissions
  - This makes the system more secure
  - It is sometimes confusing for users coming from Windows
- Shell scripts must be executable
  - `chmod +x [file]`
- Shell scripts not in the PATH environment variable must be called explicitly
  - `./script.sh`
- The shebang line specifies the interpreter

Environment variables

Ex. 12

- Environment variables give UNIX programs information that is specific to the current session
  - `USER` : User name
  - `HOME` : Home directory
  - `PATH` : List of program directories
- Print environment with `env` and `echo`
- Set an environment variable
  - `export VAR=value`
- The file `.bashrc` in the home directory contains instructions for bash to execute on startup

Variables

- Variables in bash can be assigned with =
  - `var=value`
  - No spaces!
  - Variables are untyped
- Variable substitution with `${var}`
  - `echo ${var}`
- Do not confuse bash and environment variables
- Variables are especially useful in scripts

Calculations

- let command
  - `let 'result = (5 +3) * (4 - 2) + 2**3'`
  - `let 'result *= 4'`
- Arithmetic expansion
  - `$((expression))`
  - `echo $(( (5 +3) * (4 - 2) + 2**3 ))`

Quoting and chain literals

- Bash parses each word as a separate entity; Whitespaces separate words
- Quotes are used to group words that should belong to the same entity
- There are two types of quotes in bash
    - 'single quotes' group words and prevent all substitution and expansion
    - "double quotes" group words but allow some limited substitution and expansion (such as variable expansion)

Control flow

- With bash, you can do tests, conditions and loops
- Combined with variables and UNIX programs, this makes bash a full-fledged programming language

```
#!/bin/bash

nwarn=$(( grep WARNING "${1}" | wc -l ))
if [[ nwarn == 0 ]]; then
        echo "You are ready to graduate!"
else
    echo "${nwarn} warnings; better double-check, mate"
fi
```

Remote access with SSH

- The `ssh` command allows you to connect to a remote host and work inside its UNIX environment.
- Files can be transfered using the `scp` command.
- SSH and SFTP clients exist for most operating systems, including Windows.

What is Linux and GNU?

- In 1983, Richard M. Stallman, an MIT worker, announces the GNU project.
- He was frustrated by the then recent trend of UNIX vendors to close their source code.
- GNU is a free software UNIX clone, which everyone is free to use, modify and redistribute

What is Linux and GNU?

- Linus Torvalds, a student at Helsinki University, Finland, started working on Linux in 1991.
- Linux is a free software operating system kernel, initially for PC computers.
- Together with GNU, Linux makes it possible to run a free UNIX OS on pretty much any computer.

What is Linux and GNU?

- Since Linux and GNU are free software, anyone is allowed to package it and redistribute it.
- There are therefore many flavours of Linux, called Linux distributions. These include the kernel, the GNU projects, desktop environments, etc.
- Ubuntu is a Linux distribution founded by Mark Shuttleworth in 1999.
  - One of its key objectives is to make Linux easy to use for beginners.
  - Ubuntu can optionally be installed inside an

UNIX/Linux and bash resources

- A concise Linux/UNIX quick reference is available is distributed in the exercice archive (you also have a hand-out)
  - http://fosswire.com/post/2007/8/unixlinux-command-cheat-sheet/
- A copy of the slides is in the exercise archive
  - http://www.proteo.ca/ws-2010-03.tar.bz2
- BASH Programming - Introduction HOW-TO
  - http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html
- Advanced bash scripting guide