Universitatea POLITEHNICA din Bucureşti
Facultatea de Inginerie în Limbi Străine
Departamentul de Inginerie în Limbi Străine

- anul I -

Disciplina:

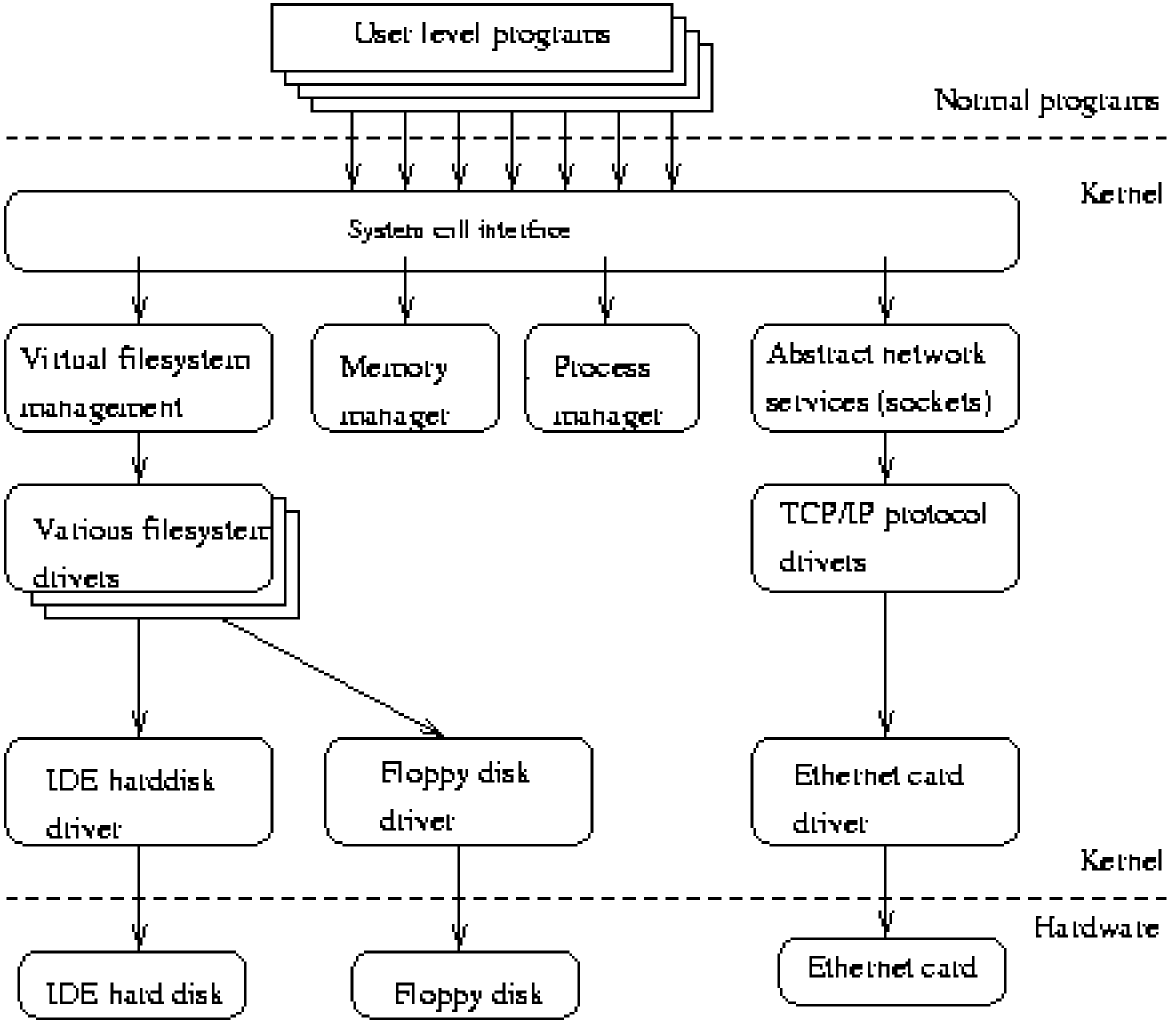
Sisteme de Operare (partea 3)

Operating Systems

Systèmes d’exploitation

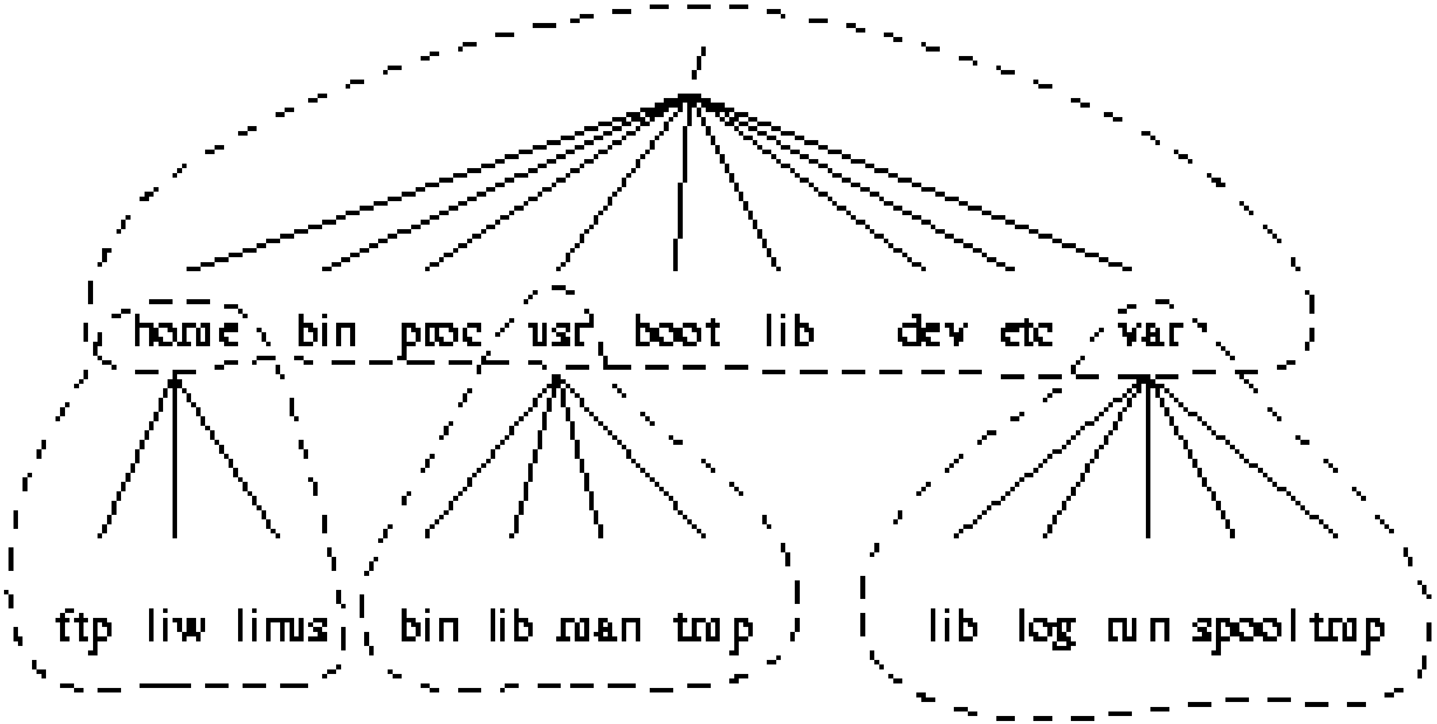
Constantin Viorel MARIAN, Eng. PhD

Linux system

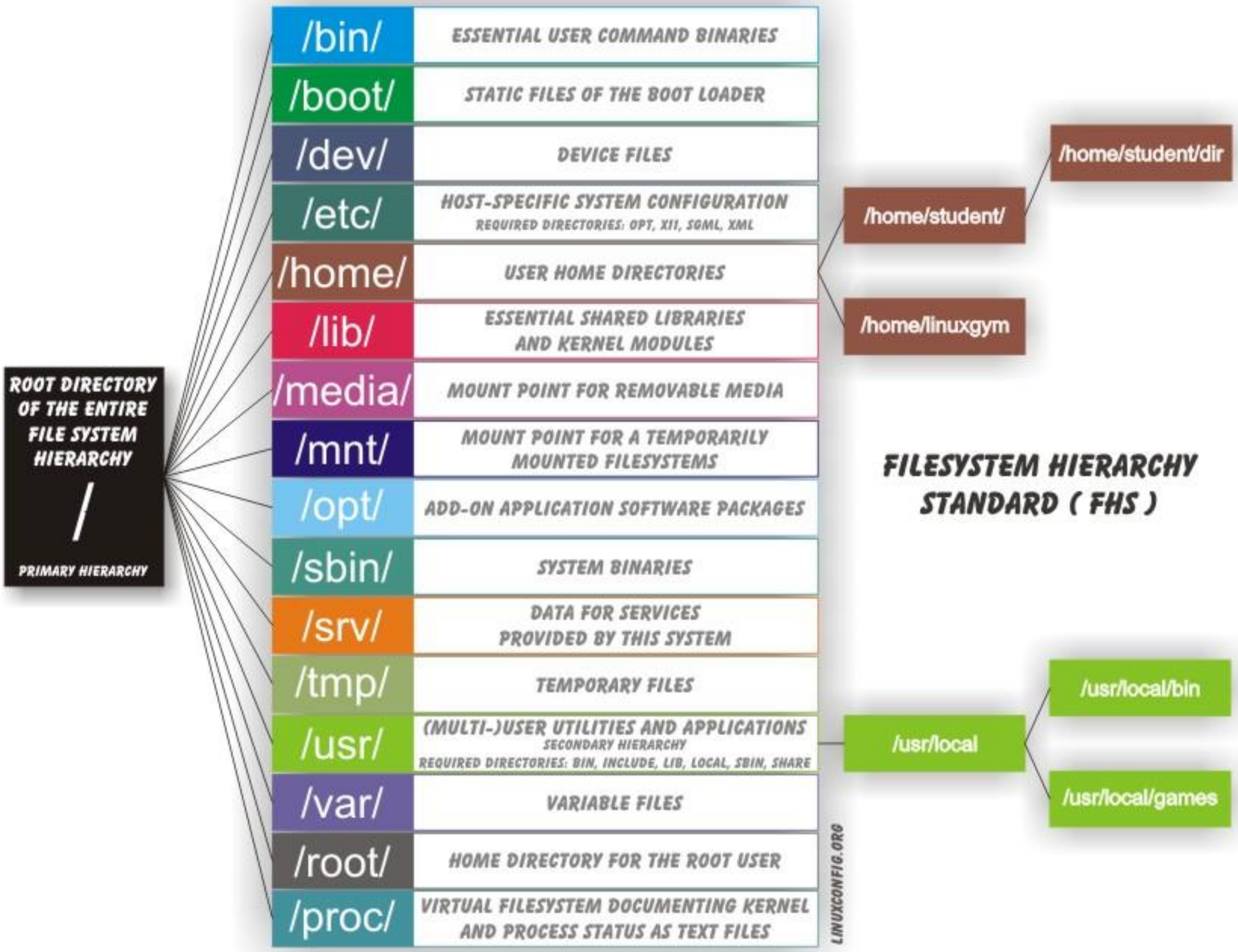


credits to / *d'après*
Alex Weeks
" The Linux System
Administrator's Guide "

Linux file system / *système de fichiers* (Filesystem Hierarchy Standard 3.0)



credits to / *d'après*
Alex Weeks
" The Linux System
Administrator's Guide "



FILESYSTEM HIERARCHY STANDARD (FHS)

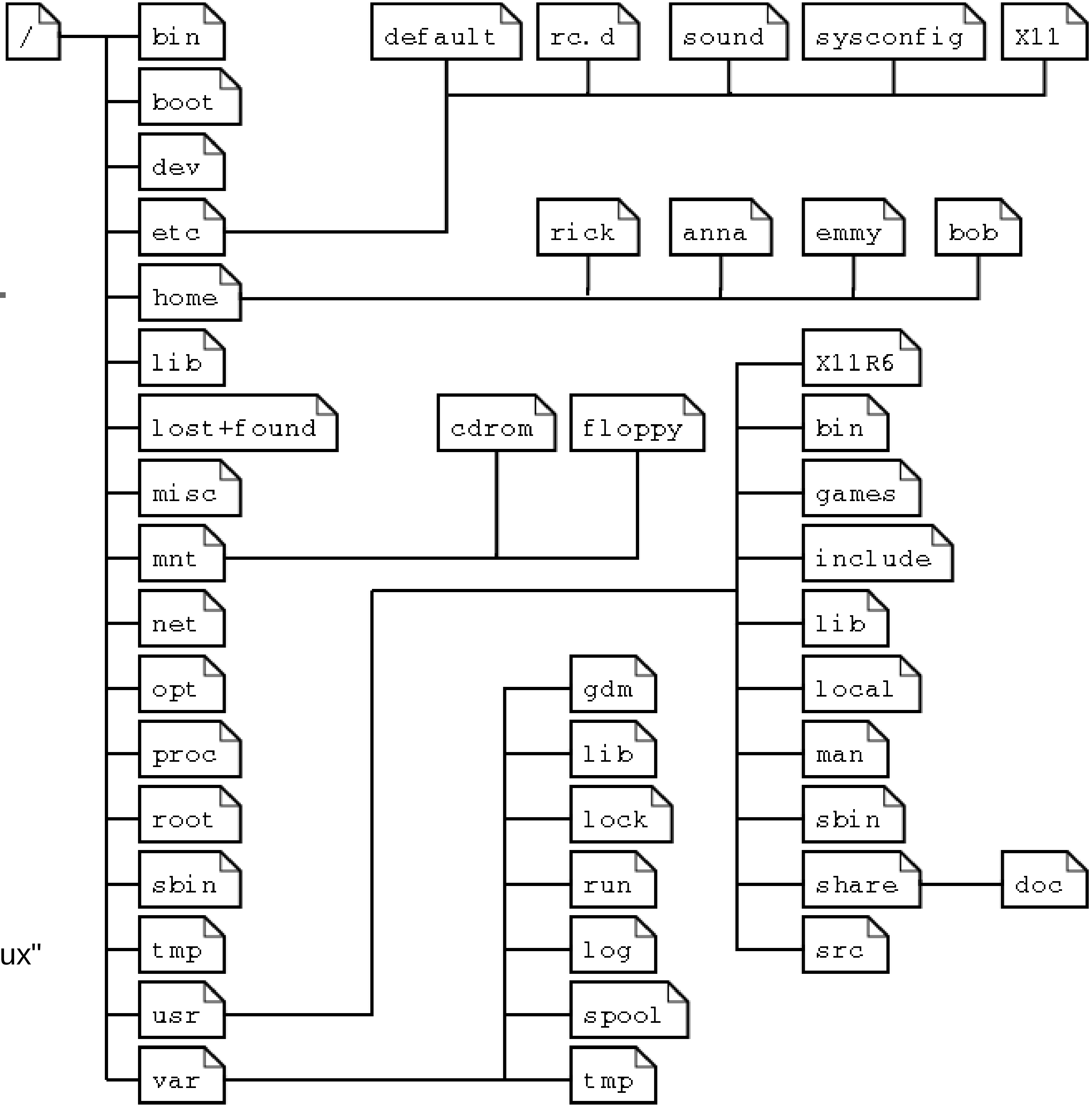
ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY

/

PRIMARY HIERARCHY

/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

Linux



- /
  - bin
  - boot
  - dev
  - etc
    - default
    - rc.d
    - sound
    - sysconfig
    - X11
  - home
    - rick
    - anna
    - emmy
    - bob
  - lib
  - lost+found
  - misc
  - mnt
    - cdrom
    - floppy
  - net
  - opt
  - proc
  - root
  - sbin
  - tmp
  - usr
    - X11R6
    - bin
    - games
    - include
    - lib
    - local
    - man
    - sbin
    - share
      - doc
    - src
  - var
    - gdm
    - lib
    - lock
    - run
    - log
    - spool
    - tmp

credits to / *d'après*
Machtelt Garrels
"Introduction to Linux"

HELP commands / *commandes d'AIDE*

- **man *command***

  Format and display online (in the operating system) the manual pages for **command** / *Formater et afficher en ligne (dans le système d'exploitation) les pages de manuel pour la* ***command***

- ***command* - - help**

Practice Platform & Documentation / *Plateforme pratique et documentation* LAB

- Debian documentation and manuals /

  documentation Debian et manuals

  www.debian.org/doc/#manuals

  www.debian.org/doc/index.fr.html#manuals

and more specific the "Debian Reference" manual /

et plus spécifique le manuel "Référence Debian"

www.debian.org/doc/manuals/debian-reference/index.en.html

www.debian.org/doc/manuals/debian-reference/index.fr.html

Note: For details about a specific command, use the “man-pages” (embedded in the operating system or on the internet) /

Remarque: Pour plus d'informations sur une commande spécifique, utilisez les "man-pages" (intégrés dans le système d'exploitation ou sur Internet)

http://manpages.debian.org/cgi-bin/man.cgi

Linux: Command Format / *La forme d'une commande*

- **command [arg 1]…[arg m] [-opt 1]…[-opt n]**
- **arg** argument
- **opt** option

Examples / *Exemples* :

- `ls`
- `ls -R`
- `ls -l`
- `ls -a`
- `ls -alR`

Linux and Windows commands / *commandes Linux et Windows*

- **Homework for the next lab /** ***Devoir pour le labo suivant***

- Linux and Windows have the same background, check in Annex 1 some similar commands. / *Linux et Windows ont le même fond, vérifiez dans l'Annexe 1 des commandes similaires.*

“Quickstart” commands / *commandes*

- **passwd** Change the password for the current user
- **sh** Open a standard shell
- **exit** or **logout** Leave this session; exit the shell
- **shutdown** Bring the system down
- **who** Show who is logged on
- **whoami** Print effective user ID
- **which** Shows the full path of (shell) commands
- **newgrp** Log in to another (new) group
- **date** Print or set system date and time

“Filesystem Hierarchy Standard FHS” commands / *commandes*

- **pwd** Display present (or current) working directory
- **ls** Displays a list of files in the current working directory on screen (**dir** command in DOS)
- **file *examplefile*** Display file type of the file with name examplefile
- **cd *directory*** Change directory (see . and ..)
- **df *file*** Reports on used disk space on the partition containing file (report file system disk usage)
- **mount** Mount a file system or display information about mounted file systems
- **umount** Unmount a file system

Commands / *commandes* - 1

- **touch** ***file*** Change file timestamps or a file that does not exist is created empty
- **cat** ***textfile*** Concatenate files and send content of textfile to standard output (screen); throws content of textfile on the screen
- **cp** ***sourcefile targetfile*** Copy files and directories
- **mv** ***oldfile newfile*** Rename or move oldfile
- **head** ***file*** Send the first part of file to standard output
- **tail** ***file*** Print the last part of file to standard output

Commands / *commandes* - 2

- **mkdir *newdir*** Make a new empty directory
- **rm *file*** Removes files and directories
- **rmdir *file*** Removes directories (empty)
- **find *path expression*** Find files in the file system hierarchy

Commands / *commandes* - 3

- **chmod *mode file*** Change file access permissions
- **chmod *mode directory***
- **chown *newowner[:[newgroup]] file*** Change file owner and group ownership
- **chgrp** Change group ownership
- **crontab** Maintain crontab files for users (-e, -l)
- **more** Filter for better viewing on screen
- **less** Opposite of more

Commands / *commandes* - 4

- **ps** Report process status (ps -ef)
- **pstree** Display a tree of processes
- **jobs** List backgrounded tasks
- **kill** Terminate process(es)
- **time** Time a simple command or give resource usage
- **top** Display top CPU processes
- **uptime** Display system uptime and average load

Commands / *commandes* - 5

vi	start the vi editor (to edit a file)
vim	edit a file
nano	start a simple text editor

Commands / *commandes* - 6

- Commands to be used to edit a file using vi editor / *Commandes pour l'editeur vi*
- **a**
- **i**
- **x**
- **dd**
- **<ESC> :w**
- **<ESC> :q**
- **<ESC> :wq!**
- **<ESC> :q!**

Commands / *commandes* - 7

how to link a command to another one / *comment deux commandes peuvent être reliés*

- standard input **(stdin)** device = **keyboard**
- standard output **(stdout)** device = **display, printer**

  standard input: 0

  standard output: 1

  standard error: 2

- **>** to put output of a command in a file
- **|** send the standard output of one command to another command as standard input
- **<** a file to be the input for a command
- **>>** append text to an existing file

Commands / *commandes* - 8

- **tar** Saves many files together into a single tape or disk archive, and restore individual files from the archive
- **gzip** Create a compressed file (reduces the size of the named files using Lempel-Ziv coding)
- **echo** ***string*** Display a line of text (string)
- **ln** ***targetfile linkname*** Make links between files (make a link with the name linkname to targetfile)
- **grep** ***PATTERN file*** Print lines in file containing the search pattern
- **umask** Set user file creation mask

Commands / *commandes* - 9

- **alias**
- **unalias**
- **help**
- **echo *string*** display a line of text (string)
- **let** performs arithmetic on shell variables
- **printf**
- **read** to read one line from standard input
- **readarray**

Commands / *commandes* - 10 (configure the network / *configurer le réseau*)

- **ip command**
- **ifconfig**
- **netstat** Display network connections, routing tables, interface statistics, masquerade, connections and multicast memberships / *Pour afficher les connexions réseau, les tables de routage, les statistiques d'interface, le masquerade, les connexions et les appartenances multicast*

configuration file / *fichiers de configuration* :

- **/etc/hosts**
- **/etc/resolv.conf**
- **/etc/nsswitch.conf**

Commands / *commandes* - 11 (check the network / *vérifier le réseau*)

- ping
- traceroute
- whois

Homework for the next lab / *Devoir pour le labo suivant*

- Using man (or help) verify the purpose of each command in Annex 2 / *En utilisant man (ou help) vérifiez le but de chaque commande dans l'Annexe 2*

Security - Files permissions / *Sécurité - autorisations des fichiers*

```
marian:~> ls -l TestFile123
-rw-rw-r-- 1 marian users 7 Oct 18 11:39 TestFile123
```

r = read access
w = write permission
x = execute permission

first triad	**rwx**	**what the file owner can do**
second triad	**rwx**	**what the group members can do**
third triad	**rwx**	**what other users can do**

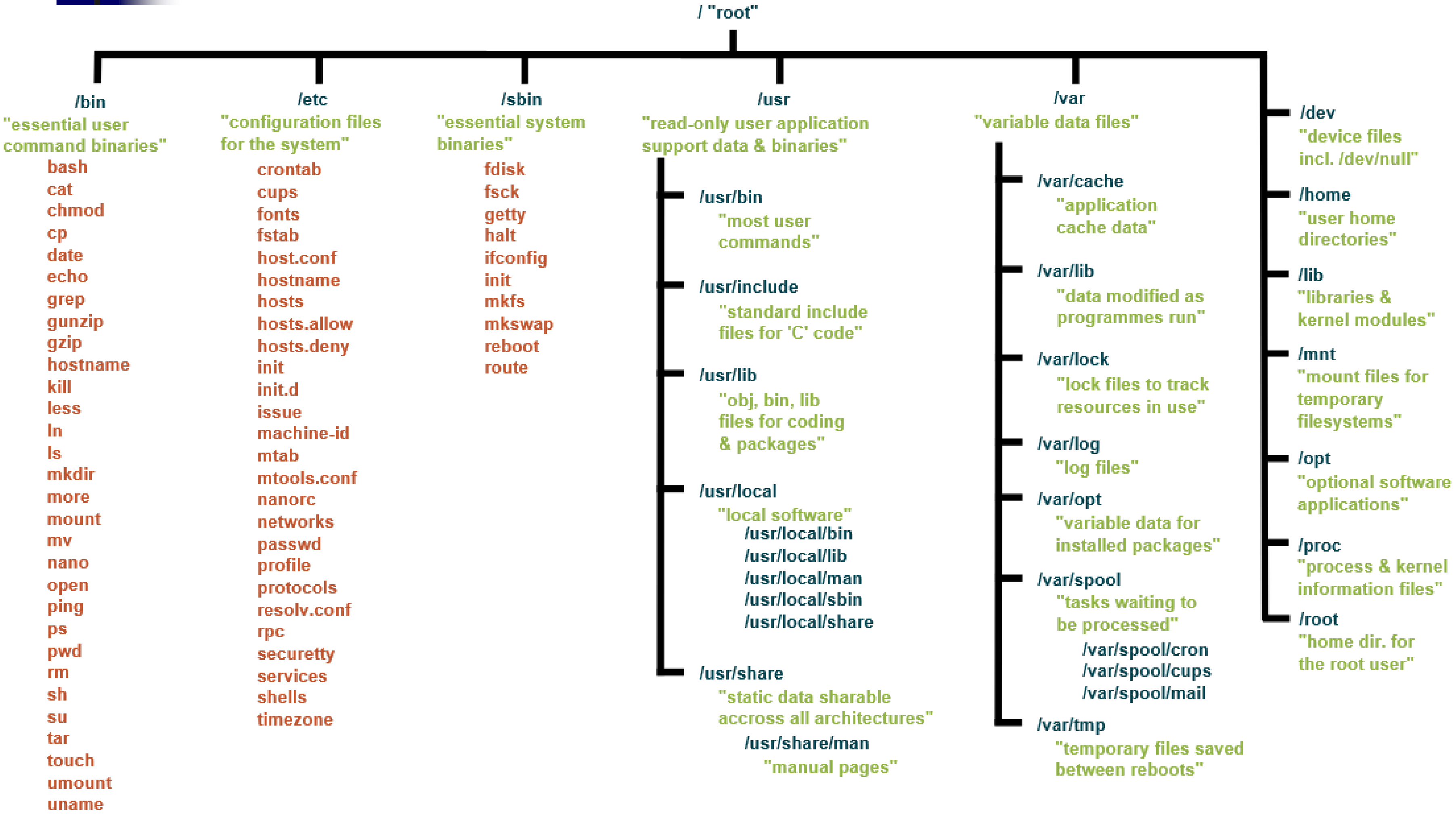
Security - Files permissions / *Sécurité - autorisations des fichiers*

```
marian:~> ls -l TestFile123
-rw-rw-r-- 1 marian users 7 Oct 18 12:39 TestFile123
```

0 or - The access right that is supposed to be on this place is not granted.
4 or r read access is granted to the user category defined in this place
2 or w write permission is granted to the user category defined in this place
1 or x execute permission is granted to the user category defined in this place

Who	**r(ead)**	**w(rite)**	**(e)x(ecute)**
u(ser)	4	2	1
g(roup)	4	2	1
o(ther)	4	2	1

UNIX file system / *Système de fichiers* (www.wikimedia.org)



/ "root"

/bin
"essential user command binaries"
bash
cat
chmod
cp
date
echo
grep
gunzip
gzip
hostname
kill
less
ln
ls
mkdir
more
mount
mv
nano
open
ping
ps
pwd
rm
sh
su
tar
touch
umount
uname

/etc
"configuration files for the system"
crontab
cups
fonts
fstab
host.conf
hostname
hosts
hosts.allow
hosts.deny
init
init.d
issue
machine-id
mtab
mtools.conf
nanorc
networks
passwd
profile
protocols
resolv.conf
rpc
securetty
services
shells
timezone

/sbin
"essential system binaries"
fdisk
fsck
getty
halt
ifconfig
init
mkfs
mkswap
reboot
route

/usr
"read-only user application support data & binaries"

/usr/bin
"most user commands"

/usr/include
"standard include files for 'C' code"

/usr/lib
"obj, bin, lib files for coding & packages"

/usr/local
"local software"
/usr/local/bin
/usr/local/lib
/usr/local/man
/usr/local/sbin
/usr/local/share

/usr/share
"static data sharable accross all architectures"
/usr/share/man
"manual pages"

/var
"variable data files"

/var/cache
"application cache data"

/var/lib
"data modified as programmes run"

/var/lock
"lock files to track resources in use"

/var/log
"log files"

/var/opt
"variable data for installed packages"

/var/spool
"tasks waiting to be processed"
/var/spool/cron
/var/spool/cups
/var/spool/mail

/var/tmp
"temporary files saved between reboots"

/dev
"device files incl. /dev/null"

/home
"user home directories"

/lib
"libraries & kernel modules"

/mnt
"mount files for temporary filesystems"

/opt
"optional software applications"

/proc
"process & kernel information files"

/root
"home dir. for the root user"

Script

A text file (e.g. script1) with the syntax / *Un fichier texte (ex. script1) avec la syntaxe*

```
#! /bin/bash
 . . . command 1
 . . . command 2
 . . . command 3
```

Note: !!! Each command can be executed by itself in the terminal

Remarque: !!! Chaque commande peut être exécutée par elle-même dans le terminal.

How to create and run a script / *Comment créer et exécuter un script*

Create a script / *Créer un script*

```
marian:~$ touch script123
```

Edit the script / *Editer le script*

```
marian:~$ nano script123
.. .. .. .. .. .. .. .. .. #!/bin/bash
.. .. .. .. .. .. .. .. .. echo “This is the OS course”
```

Add the correct permissions / *Ajouter les permissions correctes*

```
marian:~$ chmod 755 script123
```

Run the script / *Rouler le script*

```
marian:~$ /home/user/script123
This is the OS course
```

Double Quotes / *Citations doubles*

- **“........” The double quote protects everything enclosed between two double quote marks with exception of $ and ' and " and \**

  ***La double citation protège tout ce qui se trouve entre deux doubles guillemets à l'exception de $ et de ' et de " et de ***

- **Use them only for variables and command substitution. / *Utilisez-les uniquement pour la substitution de variables et de commandes .***

```
echo "$PATH"
/usr/local/bin:/usr/bin:/bin
```

Single Quotes / *Citations simples*

- **'……..' The single quote protects everything enclosed / *La citation unique protège tout ce qui est inclus***

- **Use them to turn off all special meaning of all characters. / *Utilisez-les pour désactiver toute signification particulière de tous les caractères.***

```
user@linux# echo '$PATH'
user@linux# $PATH
```

Variables definition / *Définition des variables*

- In bash for a variable there is no data type / *Dans bash, pour une variable il n'y a pas de type de données.*

- To set a variable / *Pour définir une variable*

```
variablename=value
```

- To print a variable / *Pour imprimer une variable*

```
echo $variablename
```

Variables / *Variables*

- **All bash variables are string, so bash does not distinguish between the number 1 and the character string “1” in assignments./ *Toutes les variables bash sont des chaînes, donc bash ne fait pas de distinction entre le nombre 1 et la chaîne de caractères "1".***

- **Example / *Exemple***

```
x=1
y=2
z=$x+$y
```

Variables / *Variables* (cont)

- **+ as string literal**
- ***+ chaîne littérale***

```
z=$x+$y
echo $z
1+2
```

+ as arithmetic addition

+ addition arithmétique

```
z=$(($x+$y))
echo $z
3
```

- **For numeric evaluation, enclose an expression in $((…)). After evaluation, $d is still receiving the string “3” / *Pour évaluation numérique, placez une expression dans $ ((…)). Après l'évaluation, $d reçoit toujours la chaîne "3"***

Variables / *Variables*
homework / *devoir*

```
#! /bin/bash
  VARIAB=“some_characters_to_display”
  echo $VARIAB
```

Homework 1:

```
#! /bin/bash
echo ls
```

Homework 2:

```
#! /bin/bash
echo $(ls)
```

Arguments / *Argumentes*

To pass arguments to a Bash script they are stored in variables with a number (in the order of the arguments, always starting with 1). / *Pour passer des arguments à un script Bash, ils sont stockés dans des variables avec un nombre (dans l'ordre des arguments, en commençant toujours par 1).*

```
command:      /home/user/script  arg1  arg2  arg3
```

In variable $1 we have arg1, in variable $2 we have arg2, in variable $3 we have arg3. / *Dans la variable $ 1 nous avons la valeur du arg1, dans la variable $ 2 nous avons la valeur du arg2, dans la variable $ 3 nous avons la valeur du arg3*.

Arguments / *Argumentes*

Example (arguments without spaces) / *Exemple*

```
#!/bin/bash
echo "Is $2 older than $1 ?"
echo "...this is your homework!"
```

command: **/home/user/script Windows Linux**

output: **Is Linux older than Windows ?**
...this is your homework!

Arguments / *Argumentes*

Example (arguments with spaces) / *Exemple*

```
#!/bin/bash
echo "Is $2 older than $1 ?"
echo "...this is your homework!"
```

```
command:    /home/user/script  'Win XP'  'Linux kernel'
output:     Is Linux kernel older than Win XP ?
            ...this is your homework!
```

Multiples arguments / *Argumentes multiples*

The "$@" variable - to pass a variable number of arguments to a script (in this case the variable is an array that contains all input parameters). /** ***La variable "$@" - pour transmettre un nombre variable d'arguments à un script (dans ce cas, la variable est un tableau contenant tous les paramètres d'entrée).*

```
#!/bin/bash
for i in "$@"
do
echo "there_is_another_brick_in_the_wall"
done
```

Arithmetic (Integer) Comparison / *Comparaison arithmétique*

Arithmetic (integer) comparisons / ***Comparaisons arithmétiques (entières)*** **:**

-lt	**-le**	**-eq**
-gt	**-ge**	**-ne**

Note: all the conditional expressions should be placed inside square braces with spaces around them

[$a <= $b] **is correct**

[$a<=$b] **is incorrect**

Files Comparison / *Comparaison de fichiers*

Files comparison / *Comparaison de fichiers* :

-d *file*	file exists and is a directory
-e *file*	file exists
-f *file*	file exists and is a regular file
-s *file*	file exists and is not empty
-w *file*	You have write permission on file
-r *file*	You have read permission on file
file1 **-nt** *file2*	file1 is newer than file2
file1 **-ot** *file2*	file1 is older than file2

Note: there are Korn Shell Operators and C Shell Operators

IF … THEN

if *test_expression* **; then** *statement* **; fi**

where 'statement' is only executed if ‘test_expression' evaluates to true / *où 'statement' n'est exécuté que si 'test_expression' est vrai*

if *test_expression*

then *statement*

fi

Note: one command per line or ;

une commande par ligne ou ;

IF … THEN … ELSE

if *expression* **; then** *statement1* **; else** *statement2* **; fi**

'statement1' is executed if 'expression' is true, otherwise 'statement2' is executed / *'statement1' est exécuté si 'expression' est vrai, sinon 'statement2' est exécuté*

IF … THEN … ELIF … ELSE

```
if test_expression1 ;
    then statement1 ;
        elif test_expression2 ;
        then statement2 ;
    else statement3 ;
fi
```

Very common condition examples / *Exemples pour conditions très communes*

! EXPRESSION	The EXPRESSION is false
-n STRING	The length of STRING is greater than zero
-z STRING	The lengh of STRING is zero (it is empty)
STRING1 = STRING2	STRING1 is equal to STRING2
STRING1 != STRING2	STRING1 is not equal to STRING2
INTEGER1 -eq INTEGER2	INTEGER1 is numerically equal to INTEGER2
INTEGER1 -gt INTEGER2	INTEGER1 is numerically greater than INTEGER2
INTEGER1 -lt INTEGER2	INTEGER1 is numerically less than INTEGER2
-d FILE	FILE exists and is a directory
-e FILE	FILE exists
-r FILE	FILE exists and the read permission is granted
-s FILE	FILE exists with size greater than zero (not empty)
-w FILE	FILE exists and the write permission is granted
-x FILE	FILE exists and the execute permission is granted

SELECT

```
select expression in list
do
    statement
done
```

CASE

```
case expression  in
    case1) statement1 ;;
    case2) statement2 ;;
     …
    casen) statementn ;;
esac
```

EXIT

The command "`exit`" terminates script execution if some error occurred (e.g. wrong statement, incorrect user input). / ***La commande "`exit`" met fin à l'exécution du script si une erreur s'est produite (par exemple une mauvaise instruction, une entrée incorrecte de la part d'utilisateur).***

FOR

for *variable* **in** *[list]* **; do** *statement* **; done**

For each items (value of a variable) in a given list it will perform some commands. / *Pour chaque élément (valeur d'une variable) dans une liste donnée, il exécutera certaines commandes.*

for *expression*

do *statement*

done

Note: one command per line or ;

une commande par ligne ou ;

FOR

Example / *Exemple*

```
#! /bin/bash
for VAR in 1 2 3 4 5; do
  echo The value of variable VAR is:
  echo $VAR
done
```

WHILE

while *expression* **; do** *statement* **; done**

The loop executes some commands if the expression is true and stops when it is false. /

La boucle exécute certaines commandes si l'expression est vraie et s'arrête lorsqu'elle est fausse.

UNTIL

until *expression* **; do** *statement* **; done**

The loop executes some commands while the expression evaluates to false. /

La boucle exécute certaines commandes pendant que l'expression est évaluée à false.

FUNCTION
1 - to group some commands / *regrouper certaines commandes*

- Declaring a function / *Déclaration d'une fonction*:

  function *name* **{** *code_and_commands* **}**

```
function my_quit {
  exit
}



function my_hello {
  echo Hello!!!!
}
```

FUNCTION
2 - to use the function / *utiliser la fonction*

- Calling a function: write function name /

Appel d'une fonction: écrire le nom de la fonction

```
#!/bin/bash
my_hello
my_quit
echo test_on_screen
```

FUNCTION and PARAMETERS / *FONCTION et PARAMETRES*

- Function with one parameter / *Fonction avec un paramètre*

```
function fx {
      echo $1
}
```

- Calling a function / *Appel d'une fonction*

```
#!/bin/bash
fx Linux
fx and
fx Windows
echo second_test_on_screen
```

FUNCTION and VARIABLES / *FONCTION et VARIABLES*

- Local variables are declared with the keyword **local**

Les variables locales sont déclarées avec le mot clé ***local***

- Homework / *Devoir* :

```
#!/bin/bash
var=first_text_aaa
function fx {
  local var=second_text_bbb
  echo $var
  }
echo $var
fx
echo $var
```

Redirection - 1

- **Standard streams are input and output data transfer channels between a computer operating system (program) and its environment. /**

  Les flux standards sont des canaux de transfert de données d'entrée et de sortie entre un système d'exploitation d'ordinateur (programme) et son environnement.

Redirection - 2

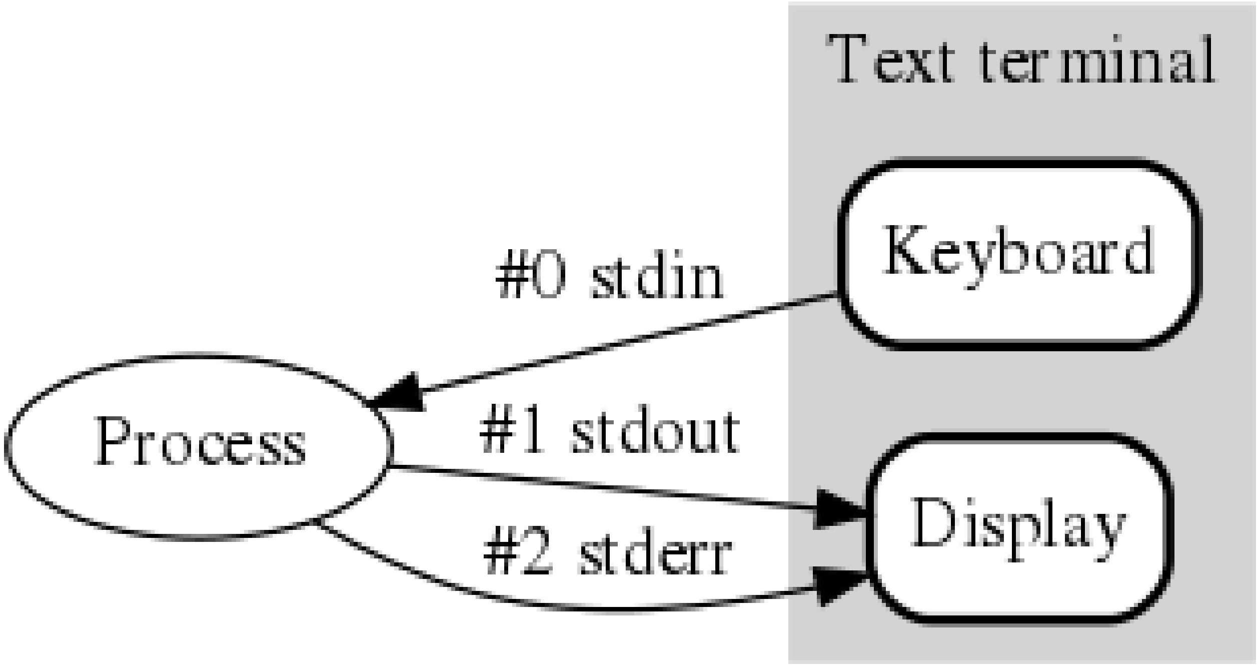
- There are three I/O connections :

  standard input (0 or stdin),
  standard output (1 or stdout),
  standard error (2 or stderr)

- *Il y a trois connexions I/O* :

  entrée standard (0 ou stdin),
  sortie standard (1 ou stdout),
  erreur standard (2 ou stderr)

Redirection - 3



credits to *www.wikipedia.org*

Redirection - 4

- **redirect stdout and stderr to a file /** ***rediriger stdout et stderr vers un fichier***

```
ls -al > file.txt
```

(is the same as: `ls -al 1> file.txt`**)**

```
./script_with_errors 2> another_file.txt
```

- **redirect stdout to stderr /** ***rediriger stdout vers stderr***

Redirection - 5

- **redirect stderr to stdout /** ***rediriger stderr vers stdout***

- **append the output to a file /** ***ajouter la sortie dans un fichier***

```
ls >> file.txt
```

Pipes / *Tubes*

- A `pipe` creates a link between the output of a program and the input of another one. / *Un tube crée un lien entre la sortie d'un programme et l'entrée d'un autre.*

```
ls -al /etc/ | less
```

```
tar -ztvf compressed_file.tar.gz | less
```

Operators – in decreasing order of equal-precedence - 1

id++ id--	variable post-increment and post-decrement
++id --id	variable pre-increment and pre-decrement
- +	unary minus and plus
! ~	logical and bitwise negation
******	exponentiation
*** / %**	multiplication, division, remainder
+ -	addition, subtraction
<< >>	left and right bitwise shifts
<= >= < >	comparison

Operators – in decreasing order of equal-precedence - 2

<table>
<tr><th></th><th></th></tr>
<tr><td><b>== !=</b></td><td>equality and inequality</td></tr>
<tr><td><b>&</b></td><td>bitwise AND</td></tr>
<tr><td><b>^</b></td><td>bitwise exclusive OR</td></tr>
<tr><td><b>|</b></td><td>bitwise OR</td></tr>
<tr><td><b>&&</b></td><td>logical AND</td></tr>
<tr><td><b>||</b></td><td>logical OR</td></tr>
<tr><td>string<b>:</b>regexp</td><td>match of regexp in string</td></tr>
<tr><td><b>= *= /= %= += -= <<= >>= &= ^= |=</b></td><td>assignment</td></tr>
<tr><td>expr1 <b>,</b> expr2</td><td>comma</td></tr>
</table>

Operators / *Operateurs*

- **Homework for the next lab / *Devoir pour le labo suivant***

- Create 18 commands or scripts to use the previous operators. / *Créez 18 commandes ou scripts pour utiliser les opérateurs précédents.*

Regular expressions (regexp) / *Expressions régulières*

- Regular expressions are patterns to be matched in a search operation: very useful in programming languages (Java, Perl), text processing (grep, sed, vim), etc. / *Les expressions régulières sont des modèles à rechercher dans une opération de recherche: très utiles dans les langages de programmation (Java, Perl) et le traitement de texte (grep, sed, vim), etc.*

Regular expressions (regexp) / *Expressions régulières*

- **metacharacter** = special character (or sequences of characters) that has a special meaning and is used to represent something else. / ***métacaractère*** *= caractère spécial (ou séquence de caractères) ayant une signification spéciale et utilisé pour représenter autre chose.*
- expl: In regular expression the metacharacter **^** means **"not"**. So "m" means "match m" and **"^m"** means **"do not match m"** / *Dans l'expression régulière, le métacaractère* ***^*** *signifie* ***"pas"****. Donc "m" signifie "match m" et* ***"^m"*** *signifie* ***"ne correspond pas m"****.*

Regular expressions (expl.) / *Expressions régulières (expl.)*

- **\n** , **\t** , **\r** are interpreted as **newline** , **tab** , **carriage return**

- \ search literal character, so **\\$5** means search for $5

- **[***characters***]** is matching for literal characters, so **[xyz]** means search for x or y or z or xy or yz or xyz

- **[^***characters***]** is NOT matching for literal characters, so **[^xyz]** means search anything but xyz (such as abcd)

Arithmetic expansion / *Expansion arithmétique*

- **Arithmetic expansion and evaluation: placing an integer expression in the following format. / *Expansion arithmétique et évaluation: placer une expression entière dans le format suivant.***

$((expression))

```
echo $(( variab1 + variab2 ))
echo $(( variab1 - variab2 ))
echo $(( variab1 * variab2 ))
echo $(( variab1 / variab2 ))
```

Note: $[expression] is deprecated / *est obsolète*

Arithmetic evaluation / *Évaluation arithmétique*

- **+** plus / *plus*
- **-** minus / *moins*
- ***** multiplication / *multiplication*
- **/** division / *division*
- ****** exponentiation / *exponentiel*
- **%** modulo (returns the remainder of an integer division) / *modulo (donne le reste d'une opération de division)*
- **=** assignment / *assignation*
- **==** equality / *égalité*
- **!=** not equality / *non égalité*

Arithmetic evaluation / *Évaluation arithmétique*

- Note: all the conditional expressions should be placed inside square braces with spaces around them. / *Toutes les expressions conditionnelles doivent être placées entre des accolades carrées avec des espaces autour d'eux.*

  [$a == $b] is **correct**

  [$a==$b]   is **incorrect**

Arithmetic evaluation / *Évaluation arithmétique*

- !! **bash** only uses integers to respond (display) !!

***bash** n'utilise que des entiers pour répondre (affichage)*

- `echo $((1+1))` **has 2 as a result**
- `echo $[1+1]` **has 2 as a result**
- `echo $[3/4]` **has 0 as a result**
- `echo 'scale=2; 3/4' | bc`

  has 0.75 as a result

String comparisons / *Comparaisons de chaînes*

<	
<=	
>	
>=	
=	(or **==**)
!=	
-z	(null)
-n	(not null)

Bash debugging

```
#!/bin/bash  -x
```

Other commands / *commandes*

- **bc (bash calculator)** - from command line or other command (using |)
- **wc** (it counts newline, word, byte in a file)
- **grep, egrep, fgrep** - print lines matching a pattern

```
grep "text_to_find" /home/user/script
```

Other commands / *commandes* (homework)

- **sort** (to sort lines in text files)
- **awk** (pattern scanning and processing language; text retrieval)
- **sed** (stream editor for filtering and transforming text)

```
sed -e '5,9d' /home/user | more
sed 's/old_text/new_text/g' /home/user/testfile
```

Scheduling / *programmation des tâches*

- **CRON automatically runs tasks at regular intervals (in the background) / CRON *exécute automatiquement les tâches à intervalles réguliers (en arrière-plan)***
- **The command is executed when the minute, hour, and month fields match the current time; AND at least one of the two day fields (day of month, or day of week) match the current day. / *La commande est exécutées lorsque les champs des minutes, des heures et des mois correspondent à l'heure actuelle; ET au moins l'un des champs de deux jours (jour du mois ou jour de la semaine) correspond au jour actuel.***

Scheduling / *programmation des tâches* (cont.)

- **Commands / *Commandes***

`crontab -e`	Edit or create a crontab file / *Modifier (créer) un fichier crontab*
`crontab -l`	Display crontab file / *Afficher le fichier crontab*
`crontab -r`	Remove crontab file / *Supprimer le fichier crontab*

- **File structure / *Le format du fichier***

```
*       *       *    *      *           command_to_be_executed
```

Scheduling / *programmation des tâches* **(cont.)**

- *** means first to last value for a specific position / *signifie première à dernière valeur pour une position spécifique***

```
*     *     *   *    *        command_to_be_executed
|     |     |   |    |
|     |     |   |    +----- day of week (0 - 7)
|     |     |   +------- month (1 - 12)
|     |     +--------- day of month (1 - 31)
|     +----------- hour (0 - 23)
+------------- min (0 - 59)
```

Note: for day of week, Sunday is 0 or 7 (or use names) /
pour le jour de la semaine, le dimanche est 0 ou 7 (ou utiliser des noms)

Script scheduling / *programmation des tâches (un script)*

- **Example 1 / *Exemple 1***

at 5.00 PM every Monday to Wednesday but only in July / *à 17h00 tous les lundi au mercredi, mais seulement en Juillet*

```
0     17    *   7   1-3       /home/user/script1
-     -     -   -    -
|     |     |   |    |
|     |     |   |    +----- day of week (0 - 6) (Sunday=0)
|     |     |   +-------- month (1 - 12)
|     |     +--------- day of month (1 - 31)
|     +----------- hour (0 - 23)
+------------- min (0 - 59)
```

Script scheduling / *programmation des tâches (un script)*

- **Example 2 / *Exemple 2***

It means the command will run at 8:20am on the 10th and 20th of each month; AND every Thursday. / *Cela signifie que la commande se déroulera à 8h20 le 10 et le 20 de chaque mois; ET tous les jeudis.*

```
20 8    10,20   *       4       /home/user/script2
-  -    -       -       -
|  |    |       |       |
|  |    |       |       +----- day of week (0 - 6)
|  |    |       +------- month (1 - 12)
|  |    +--------- day of month (1 - 31)
|  +----------- hour (0 - 23)
+------------- min (0 - 59)
```

Annex 1: Windows MS-DOS vs. Linux and Unix / *Windows MS-DOS contre Linux et Unix*

attrib	chmod
backup	tar
dir	ls
cls	clear
copy	cp

Annex 1: Windows MS-DOS vs. Linux and Unix / *Windows MS-DOS contre Linux et Unix* (cont.)

del	rm
deltree	rm -R , rmdir
edit	vi , nano , pico
format	fdformat , mount , umount , fdisk , parted

Annex 1: Windows MS-DOS vs. Linux and Unix / *Windows MS-DOS contre Linux et Unix* (cont.)

move , rename	mv
type	less <file>
cd	cd , chdir
more < file	more <file>
md	mkdir
win	startx

Annex 2: Linux (and Unix) commands / *commandes Linux (et Unix)*

- Not all commands work with every distribution. / *Toutes les commandes ne fonctionnent pas avec toutes les distributions.*

- Not all commands work with any user (root vs. other users) because privileges. / *Toutes les commandes ne fonctionnent pas avec tous utilisateurs (root ou autre utilisateur), à cause des privilèges.*

Annex 2: Linux (and Unix) commands / *commandes Linux (et Unix)* (cont.)

- **A**: a2p | ac | addgroup | adduser | alias | agrep | apropos | apt-cache | apt-get | aptitude | ar | arch | arp | as | aspell | at | atq | atrm | awk
- **B**: basename | bash | batch | bc | bdiff | bfs | bg | biff | break | bs | bye
- **C**: cal | calendar | cancel | cat | cc | cd | cfdisk | chdir | checkeq | checknr | chfn | chgrp | chkey | chmod | chown | chroot | chsh | cksum | clear | cls | cmp | col | comm | compress | continue | copy | cp | cpio | crontab | csh | csplit | ctags | cu | curl | cut
- **D**: date | dc | dd | delgroup | deluser | depmod | deroff | df | dhclient | diff | dig | dir | dircmp | dirname | dmesg | dos2unix | dpkg | dpost | du

Annex 2: Linux (and Unix) commands / *commandes Linux (et Unix)* (cont.)

- **E**: echo | ed | edit | egrep | eject | elm | emacs | enable | env | eqn | ex | exit | expand | expr
- **F**: fc | fdisk | fg | fgrep | file | find | findsmb | finger | fmt | fold | for | foreach | free | fromdos | fsck | ftp | fuser
- **G**: gawk | getfacl | gpasswd | gprof | grep | groupadd | groupdel | groupmod | gunzip | gview | gvim | gzip
- **H**: halt | hash | hashstat | head | help | history | host | hostid | hostname
- **I**: id | ifconfig | ifdown | ifup | info | init | insmod | iostat | ip | isalist | iwconfig
- **J**: jobs | join
- **K**: keylogin | kill | killall | ksh

Annex 2: Linux (and Unix) commands / *commandes Linux (et Unix)* (cont.)

- **L**: last | ld | ldd | less | lex | link | ln | lo | locate | login | logname | logout | losetup | lp | lpadmin | lpc | lpq | lpr | lprm | lpstat | ls | lsof | lsmod| lzcat | lzma
- **M**: mach | mail | mailcompat | mailx | make | man | md5sum | merge | mesg | mii-tool | mkdir | mkfs | mkswap | modinfo | modprobe | more | mount | move | mt | mv | myisamchk | mysql | mysqldump
- **N**: nc | neqn | netstat | newalias | newform | newgrp | nice | niscat | nischmod | nischown | nischttl | nisdefaults | nisgrep | nismatch | nispasswd | nistbladm | nl | nmap | nohup | nroff | nslookup
- **O**: od | on | onintr | optisa

Annex 2: Linux (and Unix) commands / *commandes Linux (et Unix)* (cont.)

- **P**: pack | pagesize | parted | partprobe | passwd | paste | pax | pcat | perl | pg | pgrep | pico | pine | ping | pkill | poweroff | pr | printenv | priocntl | printf | ps | pstree | pvs | pwd | python
- **Q**: quit
- **R**: rcp | readlink | reboot | red | rehash | remsh | rename | renice | repeat | replace | rgview | rgvim | rlogin | rm | rmail | rmdir | rmmod | rn | route | rpcinfo | rsh | rsync | rview | rvim
- **S**: s2p | sag | sar | scp | screen | script | sdiff | sed | sendmail | service | set | setenv | setfacl | sfdisk | sftp | sh | sha224sum | sha256sum | sha384sum | sha512sum | shred | shutdown | signals | sleep | slogin | smbclient | sort | spell | split | stat | stop | startx | strftime | strip | stty | su | sudo | swapoff | swapon | sysklogd

Annex 2: Linux (and Unix) commands / *commandes Linux (et Unix)* (cont.)

- **T**: tabs | tac | tail | talk | tar | tbl | tcopy | tcpdump | tcsh | tee | telinit | telnet | test | time | timex | todos | top | touch | tput | tr | traceroute | tree | troff | tty
- **U**: ul | umask | unalias | uname | uncompress | unhash | uniq | unlink | unlzma | unmount | unpack | untar | until | unxz | unzip | uptime | useradd | userdel | usermod
- **V**: vacation | vedit | vgrind | vi | view | vim | vipw | visudo | vmstat
- **W**: w | wait | wall | wc | wget | whereis | whatis | which | while| who | whoami | whois | write
- **X**: X | xargs | xfd | xhost | xinit | xlsfonts | xorg | xset | xterm | xrdb | xz | xzcat
- **Y**: yacc | yes | yppasswd | yum
- **Z**: zcat | zip | zipcloak | zipinfo | zipnote | zipsplit

Annex 3: Linux and Unix top commands / *Commandes principales Linux et Unix*

- man , help
- cd , chdir
- ls
- cp
- mv
- rm
- chmod
- nano , vi
- pwd
- tar
- Find
- grep
- ip , ifconfig
- date
- kill