

Tlemcen University
Department of computer science
1st Year Engineer
"Introduction to operating system 1"

Session 5: Unix directories and files

Academic year: 2023-2024

□ Unix directories and files

■ *Types of files*

■ *Reach a file*

- *Naming a file*

- *The access path*

■ *Metacharacters*

■ *Access rights*

- *User identification*

- *User rights definition*

- *Case 1: a standard file*

- *Case 2: a directory*

- *Related commands*

- *Change rights: chmod*

- *Change default rights: umask*

- *Change owner and group*

Types of files

❑ In LINUX, every object in the tree is considered to be a file.

❑ There are three types of file:

■ **Data files:** *a set of data*

- *directory (d)*
- *regular or ordinary: programs, data.... (-)*
- *symbolic links (l)*

■ **Exchange files:** *for data communication*

- *FIFO files or pipes (p): for communication between processes*
- *Socket (s)*
 - *A socket is the combination of an IP address and a port, and is used to create a connection between two computers.*

■ **Hardware file:** *peripheral resource*

- *Block files (b): hard disk, CD-ROM, etc.*
- *Character files (c): screen, keyboard, mouse, etc. handle one transmitted character at a time.*

❑ File naming rules

- *Older unix systems were limited to 14 characters, but nowadays long file names are supported from 1 to 255 characters.*
- *The slash (/) is forbidden because it is the directory delimiter in the tree structure, and it represents the root.*
- *Files whose name begins with a dot (.) are hidden or masked files, they do not appear by default with the "ls" command without the "-a" option.*
- *The colon (..) identifies the parent directory and the dot (.) identifies the current or working directory.*
 - *These two files exist in all directories.*
 - *It is therefore not possible to name a file with a single dot or with a colon, as the pointers already exist.*
 - *It is not possible to have two files with the same name in the same directory,*
 - *it is not possible to remove the pointer to the current directory or to the parent directory.*

❑ The metacharacters

- *Wildcards ("?", "*", "[]") can replace all or part of a path, directory or filename.*
 - *To replace any string: **
 - *To include all names starting with a capital letter from A to X: [A-X]**
 - *To replace two characters: ??*

❑ The access path

- *The tree structure of Linux starts with the root directory.*
- *The first directories just below the root are divided up by function.*
- *Depending on the distribution and/or installation of LINUX, the files in the tree are not the same and are not stored in the same place.*

- */ (The root)*
- */bin (The commands)*
- */boot*
- */dev (contains entries for physical devices that may be present on your system)*
- */etc (Configuration: contains several sub-folders, each with a specific purpose, example /etc/passwd)*
- */home (User home directories)*
- */lib (Routine libraries, the kernel module, "library")*
- */var (System logs: contains files which are likely to change frequently, such as system activity logs, print queues, etc.)*
- */root (Superuser directory)*
- */sbin (Binary system files: contains the essential system binaries (for example the adduser command))*
- */tmp (The temporary directory)*

■ *The types of paths*

- ***Absolute path** of a file always begins with a slash to identify the root.*
- ***Relative path** refers to the current or working directory.*

Access permissions

- ❑ File or directory access rights are used to protect them from unauthorized reading, writing or execution operations.
- ❑ There are three kinds of file permissions in Linux:
 - *r (read): It is possible to view or copy the file and, in the case of a directory, to list its contents.*
 - *w (write): This access right allows you to modify, remove or rename a file and, in the case of a directory, it is possible to add or remove files in the directory.*
 - *x (execute): These are the rights assigned to files that can be executed, such as programs. In the case of a directory, this gives you access to all its sub-directories.*

❑ There are three categories of users for whom access rights can be defined rights :

- *Owner: the user*

- *Group: other people belonging to the same group as the user. Groups are set up and updated by the system administrator.*

- *Other: any other person*

❑ Files are protected by a 9-bit binary protection code

Owner (u)	Group (g)	Other (o)
3 bits	3 bits	3 bits
rwX	rwX	rwX

□ Displaying permission settings

- *Permissions of a file/directory can be displayed with the “ls -l” (files and directories).*
- *Example:*

```
$ pwd
/home/hostname/user2
$ ls -l
total 8
drwxr-xr-x  2 user2      1024 Feb  9 14:22 directory1
-rw-r--r--  1 user2         0 Feb 10 10:20 emptyfile
-rw-r--r--  1 user2    104357 Feb  5 08:20 large-file
drwxr-xr-x  3 user2      1024 Feb 10 11:13 veggies2ind
```

Droits d'accès Liens Propriétaires Taille Date Heure Fichier ou répertoire

- *The first character = ‘-’, which means it’s a file*
‘d’, which means it’s a directory.
- *The next nine characters = (rw-r-r-) show the access permissions for owner, group and other.*
 - *the first three for the owner,*
 - *the next three for the group,*
 - *the last three for the others.*

---	---	---
<i>rwX</i>	<i>rwX</i>	<i>rwX</i>
<i>user</i>	<i>group</i>	<i>other</i>

- *The next column shows the owner of the file. (Here it is ‘user2’)*

❑ Changing permissions (chmod)

- *The command **chmod** is used to change access rights to a file or directory.*
- *This command is run by the owner of a file or directory, or the superuser.*
- *The command **chmod** looks like this:*

chmod access_rights namefile

- *"access_rights" refers to the access rights to be modified,*
- *"namefile" is the name of the file or directory,*

□ Access permissions can be indicated in different ways. The following is one of the forms used:

- *Use one or more letters to indicate the type of user to which the access rights apply:*
 - *u (for user)*
 - *g (for group)*
 - *o (for other users)*
 - *a (for all categories u, g and o)*
- *Indicate whether access rights should be added (+), deleted (-) or assigned (=).*
- *Use one or more letters to indicate access rights.*
 - *r (to read)*
 - *w (to write)*
 - *x (to execute)*

❑ Octal notation for permissions

- *There is another, simpler way of indicating permissions in each category, using octal numbering:*
 - *4 for read,*
 - *2 for write,*
 - *1 for execute.*
- *Correspondence of permissions in binary/octal and their meanings :*

string representation	numerical representation	single number representation
---	000	0
--X	001	1
-W-	020	2
-WX	021	3
r--	400	4
r-X	401	5
rw-	420	6
rwX	421	7

□ Examples

```
$ ls -l Rep1
```

```
drwxr-xr-x 2 user1 1024 Feb 10 11:15 Rep1
```

Do not grant execution rights for the group and others

```
$ chmod g-x,o-x Rep1 or $ chmod 744 Rep1
```

```
$ ls -l Rep1
```

```
drwxr--r-- 2 user1 1024 Feb 10 11:15 Rep1
```



```
chmod ug+rw,o-x file.mp4
```

This command adds read(r) and write(w) permission to both user(u) and group(g) and revoke execute(x) permission from others(o) for the file file.mp4.

```
chmod ug=rx,o+r abc.c
```

This command assigns read(r) and execute(x) permission to both user(u) and group(g) and add read permission to others for the file file.c.

❑ The commands **chown** and **chgrp**

- *You can change the owner of a file or directory using the **chown** command, or change the owner group of a file or directory using the **chgrp** command.*

- *The syntax of the command **chown** is as follows:*

chown [OPTIONS] USER[:GROUP] FILE(S)

- *It is possible to change the owner and group of a file (the user name and group are separated by the : character) using the following syntax:*

chown USER:FILE GROUP

- *To change the owner of a directory recursively, use the -R option (all files and sub-directories in the tree will be modified):*

chown -R USER:DIRECTORY GROUP

❑ Examples

- *To change the owner of the file “Student” with the “School” user, simply use the following command:*

chown School Student

- *You can also change the owner of a file and a directory, by specifying both.*

chown School Student Folder1

- *To change the group of a file, use the separator : followed by the group name using the following syntax:*

chown :FILE GROUP *or*

chgrp FILE GROUP

- *To change the owner of a directory recursively, use the -R option*

chown -R USER:DIRECTORY GROUP