Tlemcen University
Faculty of Science
Computer Science Department
1st Year Engineer

Academic year: 2023-2024
Introduction to operating systems 1

Solution of TPn° 2

Exercise n° 1

1) Create a directory called ***Rep1*** in your home directory.

```
mkdir  Rep1
```

2) Create two empty files ***"t1.txt"*** and ***"t2.txt"*** in the directory ***Rep1*** using a single line of commands.

```
touch   Rep1/t1.txt  Rep1/t2.txt
```

3) Use the command ***echo*** to add the message "TP 2" in the file "***t1.txt"***.

```
echo “TP 2”  > t1.txt      because t1.txt is empty

           or

echo “TP 2”  >> t1.txt
```

4) Add the message "Redirection and pipe" in the file ***"t1.txt"*** without deleting the first message.

```
echo “Redirection and pipe”  >> t1.txt
```

5) What happens if you use a greater than sign (>) in an output redirection operation?

If you use a greater than sign (>) t1.txt be cleared before adding “Redirection and pipe”.

6) What shell option can be used to avoid clearing a file in an output redirection operation? And how can this option be activated?

If you want to do not overwrite the content of an existing file you will use noclobber shell option.

You activate this shell option as follows:

```
set -o noclobber
```

7) How can this option be overruled (ignored) even if it is activated?

You can overrule the effect of this shell option even if it is activated by adding the >| symbol instead of using > alone.

Example :

```
echo “Redirection and pipe”  >|  t1.txt
```

8) How do to disactivate this option?

You disactivated this shell option as follows:

```
set +o noclobber
```

9) What is the standard file used for the stderr error stream?

```
/dev/null
```

*The file **/dev/null** is a special file. This file has a unique property: it is always empty. All data sent to /dev/null is* *removed. This function is useful when you are running a program or command that generates output that you want to ignore.*

10) Give an example of commands that can be used to redirect stdout to the "***file1.txt***" file and stderr to the standard file for errors?

```
find /  > file1.txt   2>/dev/null      (or other commands)
```

11) Give an example of commands that can be used to redirect stdout and stderr to the same file?

```
find /  > file1.txt   2>&1
```

where 2> represents a stderr redirection and 1> represents a stdout redirection

Exercise n° 2

Continuation of exercise 1

1) List the contents of the directory ***Rep1*** with details of each item.

```
ls -l Rep1/
```

2) Modify the access rights for the file ***"t1.txt"*** by granting only the owner read and write permissions.

```
chmod u+rw,g-rwx,o-rwx Rep1/t1.txt
```

3) Remove all access permissions to the file ***"t1.txt"*** and try to display its contents afterwards. What do you notice?

```
chmod a-rwx Rep1/t1.txt
```

We can not display the content of this file.

4) Remove execution rights for the directory ***Rep1*** and try to access it. What do you notice?

```
chmod u-x Rep1
```

you can not execute the command cd

5) Remove the read permission for the directory ***Rep1*** and try to list its contents. What do you notice?

```
chmod u-r Rep1
```

you can not execute the command ls

Exercise n° 3

In your home directory, create a directory called ***Rep2***, by default this directory has 755 as access rights, what are the commands (in symbolic and numerical notation) to give it the following rights (it is assumed that after each command the directory is reset to 755):

	Owner			Group			Other		
	Read	Write	Access	Read	Write	Access	Read	Write	Access
Command 1	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes
Command 2	Yes	No	Yes	No	Yes	No	No	No	Yes

```
Command 1 :      chmod 751 Rep2
                       or
                  chmod o-w Rep2

Command 2 :      chmod 521 Rep2
                    or
                 chmod u-w,g-rx,g+w,o-r Rep2
```