# LINUX: Exercises about files manipulation

Student: Marta Muñoz San Román

## Part1

1. **Add a new line, containing CHRISTOPHER#NOLAN#INCEPTION#DIRECTOR, at the end of the file. You should complete this exercise using commands instead of file editors.**

To do this, I have to use > operator and » operator to add the new line:

cat > awards.txt

(I paste the text)

echo CHRISTOPHER#NOLAN#INCEPTION#DIRECTOR >> awards.txt

And finally, I use cat awards.txt to see if it has been updated succefully:

```
marta@marta-VirtualBox:~$ cat > awards.txt
JESSE#EISENBERG#THE SOCIAL NETWORK#ACTOR
COLIN#FIRTH#THE KING'S SPEECH#ACTOR
HALLE#BERRY#FRANKIE AND ALICE#ACTRESS
NATALIE#PORTMAN#BLACK SWAN#ACTRESS
DARREN#FINCHER#THE SOCIAL NETWORK#DIRECTOR
ANNETTE#BENING#THE KIDS ARE ALRIGHT#ACTRESS
^C
marta@marta-VirtualBox:~$ echo CHRISTOPHER#NOLAN#INCEPTION#DIRECTOR >> awards.txt
marta@marta-VirtualBox:~$ cat awards.txt
JESSE#EISENBERG#THE SOCIAL NETWORK#ACTOR
COLIN#FIRTH#THE KING'S SPEECH#ACTOR
HALLE#BERRY#FRANKIE AND ALICE#ACTRESS
NATALIE#PORTMAN#BLACK SWAN#ACTRESS
DARREN#FINCHER#THE SOCIAL NETWORK#DIRECTOR
ANNETTE#BENING#THE KIDS ARE ALRIGHT#ACTRESS
CHRISTOPHER#NOLAN#INCEPTION#DIRECTOR
```

2. **Create a file called "actors" in which you only include ACTORS.**

To do this operation, I have to filter with grep, and then "create" a file with the ">" parameter to redirect that information. So:

Filter for ACTOR → grep "ACTOR" awards.txt

```
marta@marta-VirtualBox:~$ grep "ACTOR" awards.txt
JESSE#EISENBERG#THE SOCIAL NETWORK#ACTOR
COLIN#FIRTH#THE KING'S SPEECH#ACTOR
```

Redirect to a file automatically created (by copy and pasting the selected printed information)→ cat > actors.txt

```
marta@marta-VirtualBox:~$ cat > actors.txt
JESSE#EISENBERG#THE SOCIAL NETWORK#ACTOR
COLIN#FIRTH#THE KING'S SPEECH#ACTOR
^C
marta@marta-VirtualBox:~$ cat actors.txt
JESSE#EISENBERG#THE SOCIAL NETWORK#ACTOR
COLIN#FIRTH#THE KING'S SPEECH#ACTOR
```

3. **Display the files which begin with a in your home directory.**

With ls, and by using the pipe operator plus the grep command and "^a", It´s possible to list and search for the files which begin with "a". So:

Ls | grep ^a

```
marta@marta-VirtualBox:~$ ls | grep ^a
actors.txt
awards.txt
marta@marta-VirtualBox:~$
```

4. **Create a file called “directors” in which you only include DIRECTORS.**

To do this operation, I have to filter with grep, and then, with the “>” parameter, I redirect that information. So:

Filter for DIRECTOR → grep “DIRECTOR” awards.txt

```
marta@marta-VirtualBox:~$ grep "DIRECTOR" awards.txt
DARREN#FINCHER#THE SOCIAL NETWORK#DIRECTOR
CHRISTOPHER#NOLAN#INCEPTION#DIRECTOR
marta@marta-VirtualBox:~$
```

Redirect to a file automatically created (by copy and pasting the selected printed information)→ cat > directors.txt

```
marta@marta-VirtualBox:~$ cat > directors.txt
DARREN#FINCHER#THE SOCIAL NETWORK#DIRECTOR
CHRISTOPHER#NOLAN#INCEPTION#DIRECTOR
^C
marta@marta-VirtualBox:~$ cat directors.txt
DARREN#FINCHER#THE SOCIAL NETWORK#DIRECTOR
CHRISTOPHER#NOLAN#INCEPTION#DIRECTOR
```

5. **Display the lines of “awards” containing actresses. Show the line number.**

To display lines, it’s needed to use the parameter “-n”. So, in addition to this, I have to execute also the word “actress” between double quotes:

grep “ACTRESS” -n awards.txt

```
marta@marta-VirtualBox:~$ grep "ACTRESS" -n awards.txt
3:HALLE#BERRY#FRANKIE AND ALICE#ACTRESS
4:NATALIE#PORTMAN#BLACK SWAN#ACTRESS
6:ANNETTE#BENING#THE KIDS ARE ALRIGHT#ACTRESS
marta@marta-VirtualBox:~$
```

As it appears above, these words are in lines 3, 4 and 6.

6. **Display the lines in “awards” which are not directors. Show the line number.**

This could be done using the parameter “-v”, which don´t include lines including a certain word or letter. To show the line number, I have to use a combination of “-n” and “-v”. So:

grep “DIRECTOR” -nv awards.txt

```
marta@marta-VirtualBox:~$ grep "DIRECTOR" -nv awards.txt
1:JESSE#EISENBERG#THE SOCIAL NETWORK#ACTOR
2:COLIN#FIRTH#THE KING’S SPEECH#ACTOR
3:HALLE#BERRY#FRANKIE AND ALICE#ACTRESS
4:NATALIE#PORTMAN#BLACK SWAN#ACTRESS
6:ANNETTE#BENING#THE KIDS ARE ALRIGHT#ACTRESS
```

cl

## Part 2

File “people.txt” creation:

```
marta@marta-VirtualBox:~$ cat > people.txt
Ana;Perrrrrrrrales;1000
Pedro;Soria;1600
Jacinto;Manzano;500
Claudia;Pastor;600
Xavier;Granados;1000
Soraya;Annnnnnnneja;3000
Xavier;Guerra;4500
Sara;Siria;675
Pablo;Serrano;1000
Sonia;Mono;1600
^C
```

7. **Create a file called “people2.txt”, containing those people whose name begins with J or S.**

To do this operation, I have to filter with grep, and then, with the “>” parameter, I redirect that information. So:

Filter for the letters, using the brackets to select both → grep “^[JS]” people.txt

```
marta@marta-VirtualBox:~$ grep "^[JS]" people.txt
Jacinto;Manzano;500
Soraya;Annnnnnnneja;3000
Sara;Siria;675
Sonia;Mono;1600
marta@marta-VirtualBox:~$
```

Redirect to a file automatically created (by copy and pasting the selected printed information)→ cat > people2.txt

```
marta@marta-VirtualBox:~$ cat > people2.txt
Jacinto;Manzano;500
Soraya;Annnnnnnneja;3000
Sara;Siria;675
Sonia;Mono;1600
^C
```

8. **Create a file called “people3.txt”, containing those people whose names do not begin with S**

To do this operation, I have to filter with grep and use the parameter “-v” (that don´t include the lines that contains certain characters or words), and then, with the “>” parameter, I redirect that information. So:

Filter for the people whose name don´t begin with S → grep “^S” -v people.txt

```
marta@marta-VirtualBox:~$ grep "^S" -v people.txt
Ana;Perrrrrrrales;1000
Pedro;Soria;1600
Jacinto;Manzano;500
Claudia;Pastor;600
Xavier;Granados;1000
Xavier;Guerra;4500
Pablo;Serrano;1000
```

Redirect to a file automatically created (by copy and pasting the selected printed information)→ cat > people3.txt

```
marta@marta-VirtualBox:~$ cat > people3.txt
Ana;Perrrrrrrales;1000
Pedro;Soria;1600
Jacinto;Manzano;500
Claudia;Pastor;600
Xavier;Granados;1000
Xavier;Guerra;4500
Pablo;Serrano;1000
^C
```

9. **Display people whose name begin with S and redirect to file to “people_s.txt”**

To do this operation, I have to filter again with grep, and then, with the “>” parameter, I redirect that information to the file “people_s.txt”. So:

Filter for the people that starts with “S” → grep “^S” people.txt

```
marta@marta-VirtualBox:~$ grep "^S" people.txt
Soraya;Annnnnnnneja;3000
Sara;Siria;675
Sonia;Mono;1600
```

Redirect to a file automatically created (by copy and pasting the selected printed information)→ cat > people_s.txt

```
marta@marta-VirtualBox:~$ cat > people_s.txt
Soraya;Annnnnnnneja;3000
Sara;Siria;675
Sonia;Mono;1600
^C
```

**10. Display the number of people whose name begins with A (case insensitive).**

To display the number of people whose name begins with “A”, I need to do the same as in previous exercise, with the exception that the “A” must be case insensitive. To do this, I need to use the condition “-i”:

grep “^A” -i people.txt

```
marta@marta-VirtualBox:~$ grep "^A" -i people.txt
Ana;Perrrrrrrales;1000
marta@marta-VirtualBox:~$ grep "^a" -i people.txt
Ana;Perrrrrrrales;1000
```

As it´s shown above, as it´s case insensitive, I can do it with “a” in capital letters or in minus.grep

**11. Display how many people earn 1000 and concatenate the result in people2.txt**

To concatenate, I need to use the pipe operator and make the following commands be executed at the same time:

cat > people2.txt | grep "1000" people.txt

letter -c

```
marta@marta-VirtualBox:~$ cat > people2.txt | grep "1000" people.txt
Ana;Perrrrrrrales;1000
Xavier;Granados;1000
Pablo;Serrano;1000
^C
```

## Part 3

12. **Create a file called list, including the contents <u>from the current directory</u>**

To do this, I first list all the contents of my current directory (which is “home”) by using the command “ls”, and then, use the “cat” parameter to create the new file:

ls

cat > list.txt

```
marta@marta-VirtualBox:~$ ls
actors.txt  Desktop        Downloads  people2.txt   people.txt  snap       Videos
awards.txt  directors.txt  exercises  people3.txt   Pictures    systems
config      Documents      Music      people_s.txt  Public      Templates
marta@marta-VirtualBox:~$ cat > list.txt
actors.txt  Desktop        Downloads  people2.txt   people.txt  snap       Videos
awards.txt  directors.txt  exercises  people3.txt   Pictures    systems
config      Documents      Music      people_s.txt  Public      Templates
^C
```

**13. Open the file to check if the content is right using cat, more and less and observe the differences**

Checking the content with “cat”:

```
marta@marta-VirtualBox:~$ cat list.txt
actors.txt  Desktop        Downloads  people2.txt   people.txt  snap       Videos
awards.txt  directors.txt  exercises  people3.txt   Pictures    systems
config      Documents      Music      people_s.txt  Public      Templates
```

Checking the content with “more”:

```
marta@marta-VirtualBox:~$ more list.txt
actors.txt  Desktop        Downloads  people2.txt   people.txt  snap       Videos
awards.txt  directors.txt  exercises  people3.txt   Pictures    systems
config      Documents      Music      people_s.txt  Public      Templates
```

In more,

Checking the content with “less”, which is in fact more:

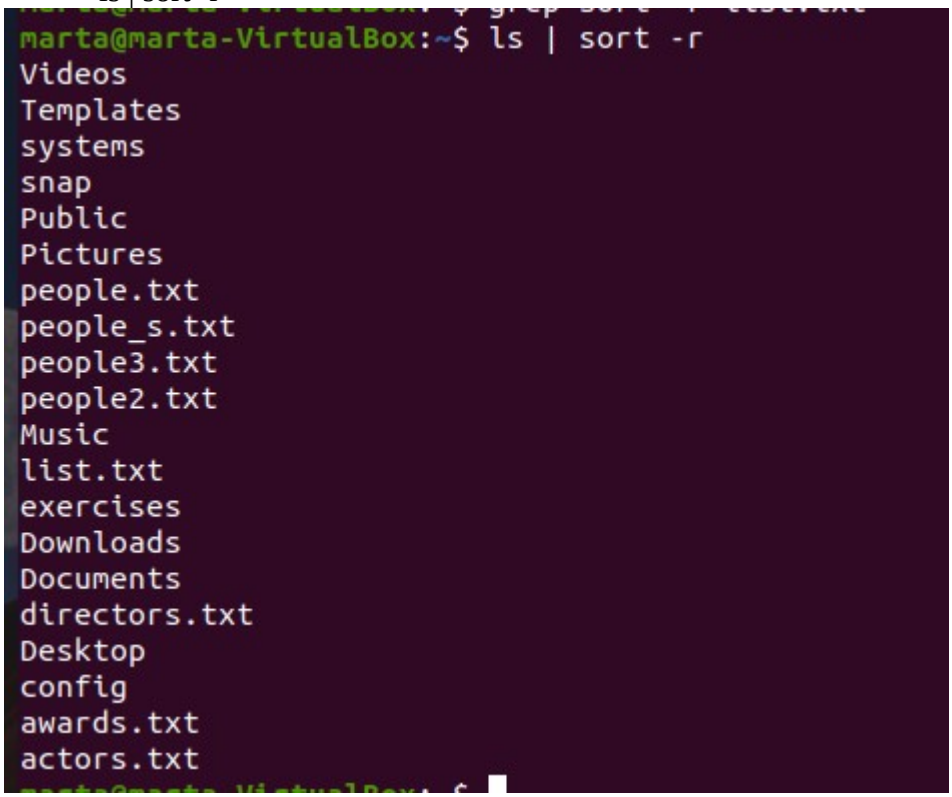
```
actors.txt  Desktop        Downloads  people2.txt   people.txt  snap       Videos
awards.txt  directors.txt  exercises  people3.txt   Pictures    systems
config      Documents      Music      people_s.txt  Public      Templates
list.txt (END)
```

In less, there are more possibilities: it can be used “b”, to go back a page Enter Key, or use the Arrow keys to scroll horizontally the file. To exit from this display, I have to press “q”.

**14. Print the contents of the current directory in reverse alphabetical order.**

To print the content of the current directory in reverse alphabetical order, I need to use sort -r and ls:

ls | sort -r

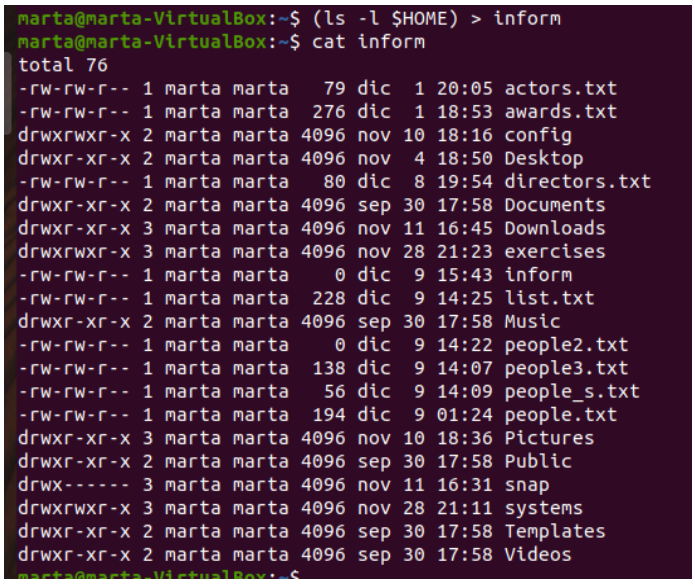



**15. Create a file called inform, containing the long format list of the files and directories in your home directory**

To create this file, I need to use the “>” parameter to redirect and automatically create a file with a defined name. Firstly, between brackets, I write “(ls -l $HOME)”, in order to list all the contents of my home directory:

(ls -l $HOME) > inform

cat inform




And I have use cat to show the result.

16. **Find the word FILE in each file of your home directory, ignoring case and showing the line number (create files containing this word if you want any match).**

To do this operation, I need to use the grep parameter with the combination of “-iwn”, where -w option force the exact “FILE” word, and the -i to ignore case sensitive and -n to show the line number. Finally, with 2> I redirect the errors to another route (/dev/null):

```
grep "FILE" -iwn $HOME/* 2> /dev/null
```