Solution of TP n°4 : Unix filters (1)

Exercise n° 1

1) Write the command to replace "a" with "A", "," with ";" and "/" with "_" in *file1.txt* and redirect the result of this command to *file2.txt*.

```
tr 'a,/' 'A;_' <file1.txt  >file2.txt
```

2) Write the command to replace $ with F, () with {} and <tab> with <space> in *file1.txt* and redirect the result of this command to *file2.txt* (<tab> is 011 in octal).

```
tr '$()\011' 'F{} ' <file1.txt  >file2.txt
```

3) Write the command that converts uppercase letters to lowercase letters of the file *file1.txt* and redirect the result of this command to *file2.txt*.

```
tr [A-Z] [a-z] <file1.txt  >file2.txt
```

4) Write the command to replace several consecutive spaces with a single space in *file1.txt* and redirect the result to *file2.txt*.

```
tr -s ' ' <file1.txt >file2.txt
```

5) Write the command that replaces several consecutive <new line> with a single one in *file1.txt* and redirects the result to *file2.txt*.

```
tr -s '\012' <file1.txt >file2.txt
```

```
\012 is the ASCII code of LF, line feed (Saut de ligne)
```

6) Write the command that replaces several consecutive spaces or <new line> or <tab> with a single one in the file *file1.txt* and redirects the result to the file *file2.txt*.

```
tr -s '\011\012' <file1.txt >file2.txt
```

```
\011 is the ASCII code of TAB
```

7) Write the appropriate command to build an alphabetical dictionary of all the words in the file *file1.txt* and sort them alphabetically, eliminating duplicates (the generating dictionary is called *file_dic*).

```
tr -cs 'a zA Z' '\012' < file1.txt | sort -u > file_dic
```

- The option '-c' is used to *remove all characters except letters.*
- The option '-s' is used to remove repeated instances of characters of the last SET specified.
- '\012' is used to generate words
- -u in the command sort is used to duplicates.

Exercise n° 2

1) Write the commands that split the file *file1.txt* into files with 5 lines and the prefix "*Ing*", and check the number of lines in each of the files created.

```
split -n 5 file1.txt  Ing
wc -l Ing*
```

2) Write the commands that divide the file *file1.txt* into 5 files with a suffix whose length is 3, and return the number of lines in each of the files created.

```
split -n 5 -a 3 file1.txt
wc -l xaa*
```

3) Write the command that splits a file into 5 files with a numeric suffix, avoiding the creation of files of zero size.

```
split -n5 -e -d file1.txt
wc -l xaa*
```

4) Write the commands that divide a file into files each containing 8 bytes and check the number of characters in each of the files created. What do you notice?

```
split -b8 file1.txt or split -b 8 file1.txt
wc -c x*
LF, line feed is counted as one character (1 byte)
```

Exercise n° 3

1) Display the number of lines, characters and words in the "/etc/passwd" file.

```
wc /etc/passwd
```

To display only :
a) Lines: `wc -l /etc/passwd`
b) Words: `wc -w /etc/passwd`
c) Characters: `wc -c /etc/passwd`

2) Display the lines in the "/etc/passwd" file that meet each of the following criteria:

a) the first 10 lines,

head –n -10 /etc/passwd or *head -10 /etc/passwd*

b) the last 10 lines,

tail –n -10 /etc/passwd ou *tail -10 /etc/passwd*

c) all lines from the tenth line onwards,

tail –n -10 /etc/passwd

d) lines between 10 to 25.

head -25 /etc/passwd | tail -16

3) Retrieve lines 4 to 10 from a file of 15 lines using pipes (this operation can be performed in two ways).

head -10 fichier | tail -7

or

tail +4 fichier | head -7

or

tail -12 fichier | head -7

4) How do you display the tenth line of a file?

head -10 fichier | tail -1

5) Create the file "test.txt" after executing the following commands:

cat > test.txt

first tp

second tp

third tp

fourth tp

fifth tp

6) Display the contents of the file "test.txt" sorted in reverse alphabetical order.

sort –r ~/test.txt