1)- Software installation :

The source codes of our project are stored on 3 different files : **fonctions.c**, **programme.c**, and finally **head.h** .

fonctions.c : is a C file which contains the implementation of all the functions and procedures used by our project.

programme.c : the “menu” of our project, a C program that simulates a shell interface.

head.h : the header of our the project, contains the global variables, the functions declaration, and the structure definition.

Compilation :

To compile the project, go into the project directory, then use the **‘make’** command. This will generate an executable file named ‘programme’.

Execution :

To run the project, go on the project directory shell and write the following line :
‘./programme partition’ , partition is the linux file used as a partition for our project.

2)-User guide :

The overall interface of our project is very similar to the linux shell interface. It simulates a console interface where you can execute commands.

There are 13 commands available which are :

1)- ls
2)- cd
3)- dossier
4)- fichier
5)- copier
6)- couper
7)- coller
8)- contenu
9)- renommer
10)- supprimer
11)- clear
12)- reboot
13)- exit

We will now discuss about each of the commands, its syntax and its work.

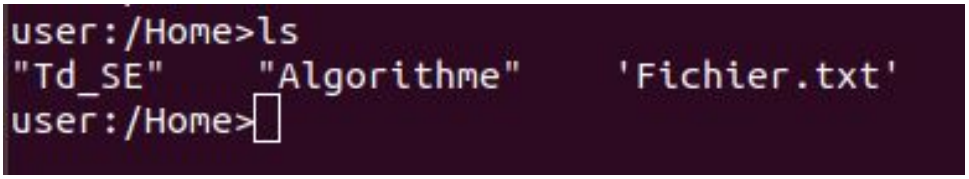
1)-The “ls” command :

ls lists the files and subdirectories present in the current working directory. Files are surrounded by ‘ ‘ while directories are surrounded by “ “.

Syntaxe :

ls

Example :



2)-The “cd” command :

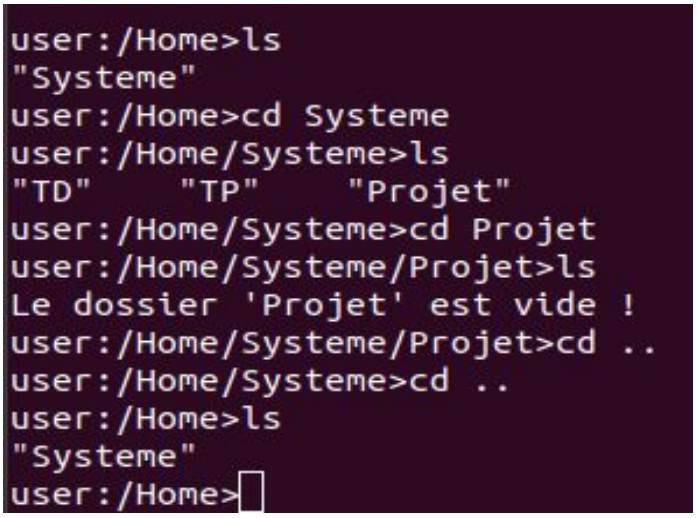
cd is used to change the current working directory.The upper level we can go to is the “home” directory, which represents the root of our partition.

Syntaxe :

cd <directory_name> : Go into the specified directory.

cd .. : Go into the father of the actual working directory.

Example :



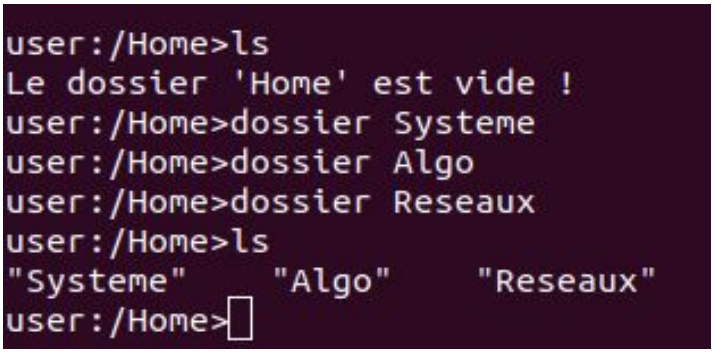
3)-The “dossier” command :

dossier is used to create a new directory.

Syntaxe :

dossier <directory_name> : Create a directory with the specified name inside the actual working directory.

Example :



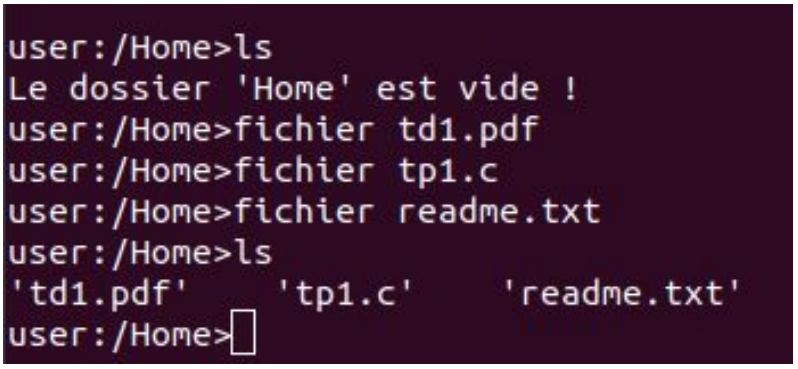
4)-The “fichier” command :

fichier is used to create a new file. The content of the file is filled with the content of the **‘EcritureFichierTexte.txt’** file content, this file is located on the same directory of the project. So, if the user want to create a file with a specific content, he just has to write the content of the text file **‘EcritureFichierTexte.txt’**, and then create a file with a specific name. It will result in the creation of a file with specific name and with the same content written on the text file.

Syntax :

fichier <file_name> : Create a file with the specified name inside the actual working directory.

Example :



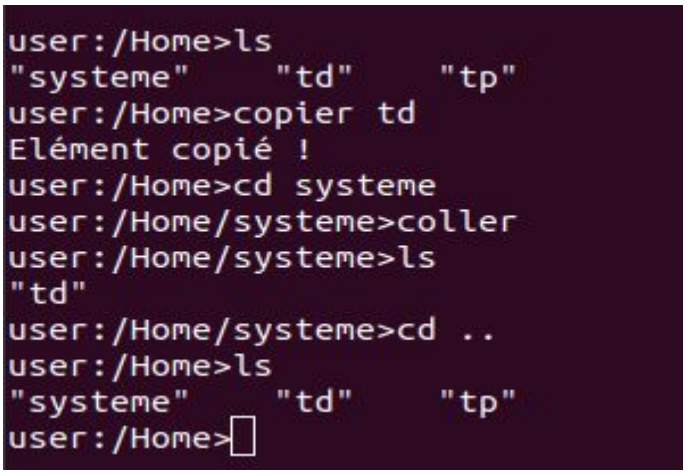
5)-The “copier” command :

The copier command allows the user to copy a file or a directory. The copied element is saved in memory, and that will be the pasted element if the user calls the coller command. (Unless the user calls the copier or couper command on another element).

Syntax :

copier <elment_name> : Save in memory that the specified element is the one to be past if the user calls the coller command.

Example :



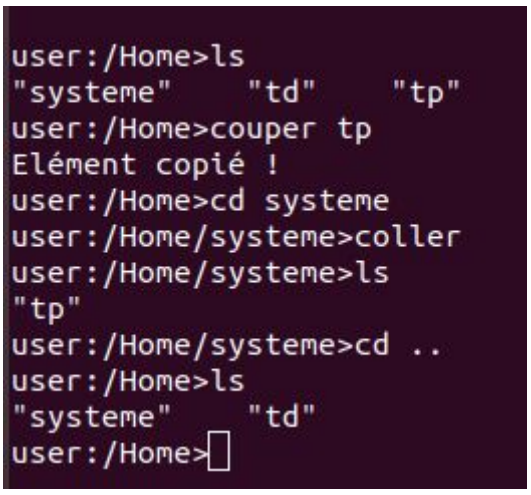
6)-The “couper” command :

The couper command allows the user to cut a file or a directory. The cut element is saved in memory, and if the user calls the coller command, then it will paste the cut element on the actual working directory AND delete the original element from its original directory. (In brief it performs a classic cut/paste operation)

Syntax :

couper<element_name> : Save in memory that the specified element is the one to be past if the user calls the coller command. Also delete the original element from its original directory.

Example :



7)-The “coller” command :

The coller command is used to paste on actual working directory the last copied or cut element.

Syntax :

coller : Paste the last saved element by the copier or couper commands.

Example :

The screens used in copier and couper can serve as examples for the work of the coller command.

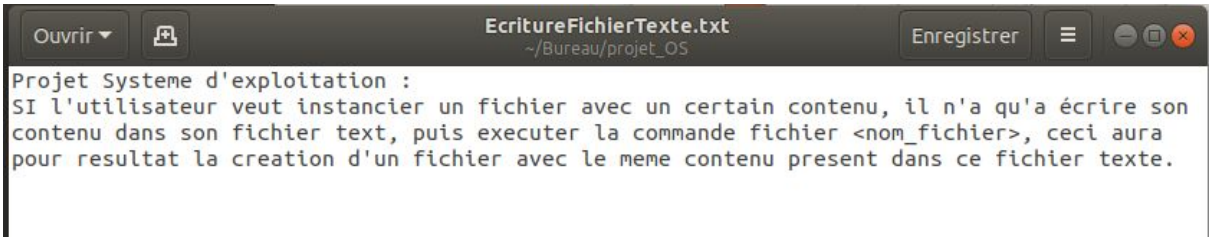
8)-The “contenu” command :

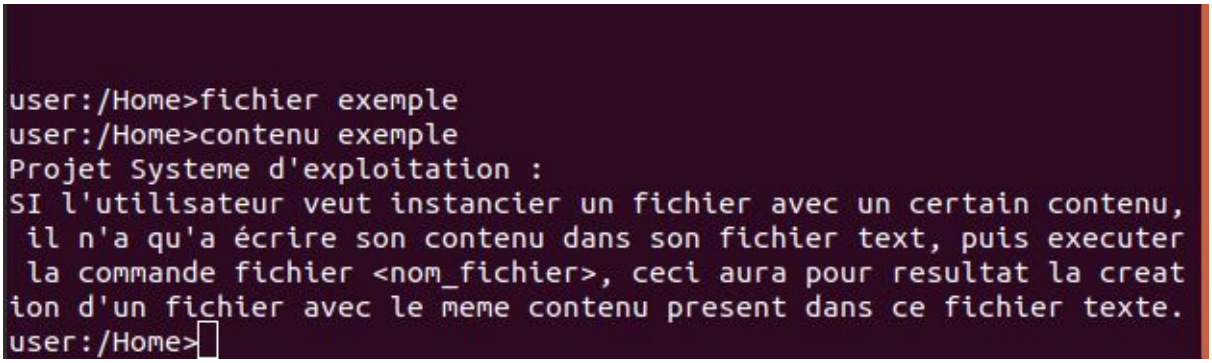
Show the content of a file.

Syntax :

content <file_name> : Prints on the terminal the content of the specified file.

Example :





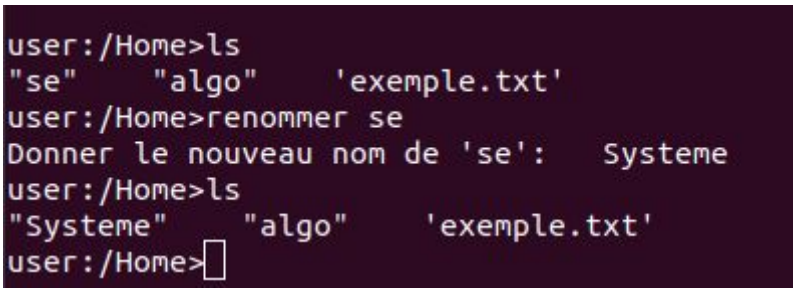
9)-The “renommer” command :

The renommer command allows the user to rename a file or a directory.

Syntax :

renommer <name_element>

Example :



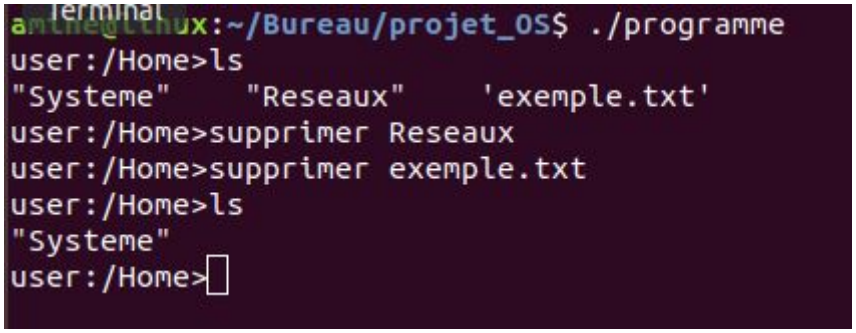
10)-The “supprimer” command :

supprimer allows the user to delete a file or a directory.

Syntax :

supprimer <name_element>

Example :



11)-The “clear” command :

The clear command clears the terminal screen.

Syntax :

clear

Example :

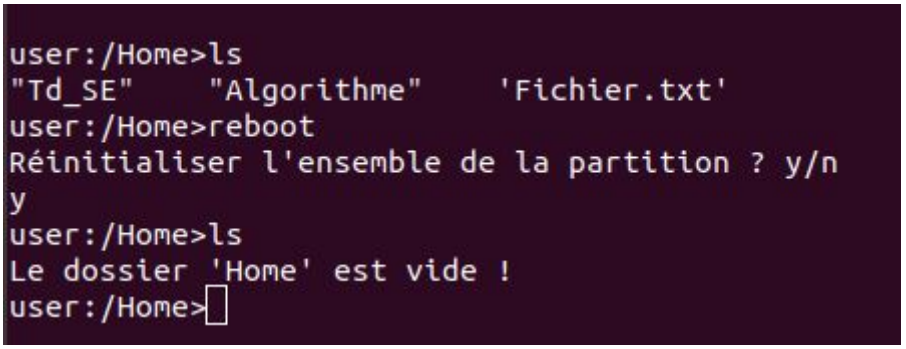
12)-The “reboot” command :

The reboot command performs a reboot on the partition, as it delete all the files and directories saved in it. The result of the reboot is a partition with only the root directory ‘home’ in it.

Syntax :

reboot

Example :



13)-The “exit” command :

The command exit allows the user to exit the program.

Syntax :

exit

Example :