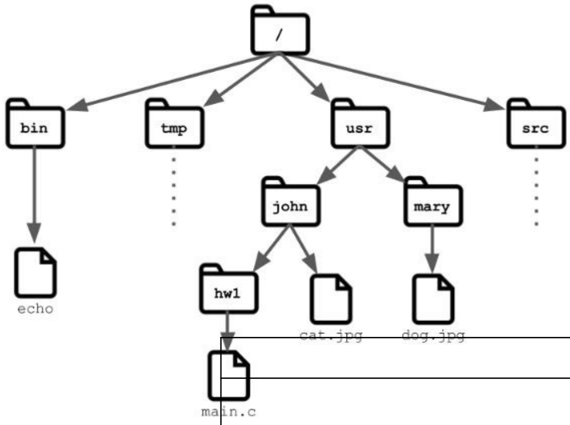


wslUNIX R1.04

TP4

N. Delomez – J.F. Kamp - X. Roirand

Prénom Nom : Carré Lucien	Date : 02/12/24
Groupe : 1D2	

REMARQUES : comme pour le TP3, à chaque question vous ferez une copie d'écran du contenu de la fenêtre de votre terminal ou un copier/coller des lignes apparaissant dans votre terminal, c'est à dire de la ou les commandes saisies avec les réponses générées par l'interpréteur de commandes.

Ce TP4 a un énoncé en anglais, vous pouvez y répondre en anglais ou en francais. Tout ce qui sera sur fond bleu signifiera qu'il faut répondre à une question. Le reste est de l'information ou des élements qui vont vous permettre de répondre aux questions suivantes..

TP 04 : The command line in Linux

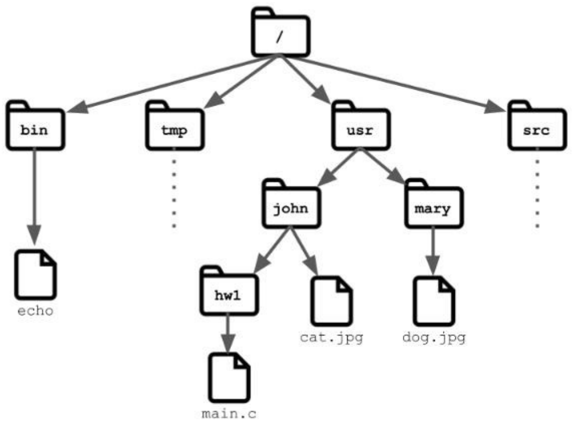
Introduction: This is the first of a two-part introduction to the GNU/Linux operating system installed on the machines of the Technoloy University Institute department. It gives you some basic knowledge that you will use throughout your courses. All notions discussed here will be considered as acquired and mastered for all future courses regardless of the module.
This third TP invites you to (re)discover the command line (shell).
Connecting to your session gives you access to your work environment. You will need to open a shell (Terminal) to enter the instructions of this TP.

1 First steps

1.1 File System

In Linux everything is a file, where directories are files that contain names of other files. In order to manage these files in an orderly fashion, we like to imagine them as an ordered tree. Each vertex is a file and folders(which are also files) are vertices that can have other children. The root of the file system is the "root" folder, denoted by "/". Under it you will find all the rest of the files. Let's go over few of the important(for this TP) sub-directories of the root :

— **/home** : Where you will find your users' personal directories.
— **/home/"username"** : Your private folder. In most shells(explained soon) the character "~" is interpreted as the path to your privet directory.
— **/bin :** Here you will find most of the commands(applications) that we'll use during this TP.



1.2 Shell and Terminal

A Terminal is a type of "text input/output environment", which means it can read your textual input and output some text on the screen. In it we'll input our commands to the computer, and get results. To open the terminal on you computer(one of the ways), you press "Ctrl+Alt+T".
The terminal doesn't perform your commands, but sends them to a Shell. The shell is a "command line interpreter", meaning it receives your textual commands and tries to interpret and preform your commands.

1
You can think of it as the steering wheel of you computer. There are multiple types of shells(Sh,Zsh,Ksh. . .), we are going to use Bash.

A few tricks that can be useful while using the shell :

— **Tab** : While writing you command you can click the Tab button, and the shell will try to auto-complete your command. If it has a multiple options for completion it will do nothing. But, clicking it twice will print out all of the options it found.
— **Ctrl+C & Ctrl+V** : You'll find out that these two shortcuts don't work 'normally'. This is partially since "Ctrl+C" is designated for stop running process. Instead you can use "Ctrl+Shift+C" and "Ctrl+Shift+V".
— **Special folder shortcuts :** As described earlier, "~" is understood as the address of your home directory. Other shortcuts are "." (current folder), ".." (previous folder).

2 Command line : first steps

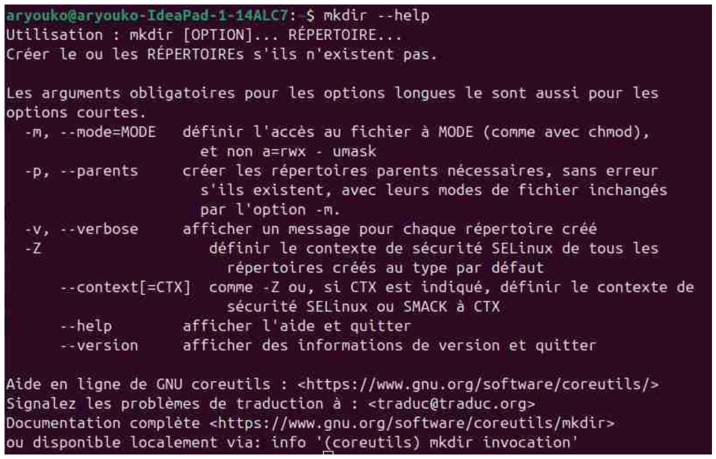
[…]

2.1.4 Other ways of getting help

There are other option to find help :

1. Adding the option - -help at the end of a command, gives you a short explanation about this com-mand, e.g. mkdir - -help.

Q1) Type “mkdir –help” then copy in the output here



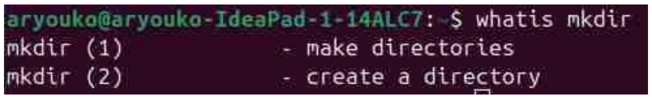
```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ mkdir --help
Utilisation : mkdir [OPTION]... RÉPERTOIRE...
Créer le ou les RÉPERTOIREs s'ils n'existent pas.

Les arguments obligatoires pour les options longues le sont aussi pour les
options courtes.
  -m, --mode=MODE   définir l'accès au fichier à MODE (comme avec chmod),
                      et non a=rwx - umask
  -p, --parents     créer les répertoires parents nécessaires, sans erreur
                      s'ils existent, avec leurs modes de fichier inchangés
                      par l'option -m.
  -v, --verbose     afficher un message pour chaque répertoire créé
  -Z                   définir le contexte de sécurité SELinux de tous les
                         répertoires créés au type par défaut
      --context[=CTX]  comme -Z ou, si CTX est indiqué, définir le contexte de
                         sécurité SELinux ou SMACK à CTX
      --help        afficher l'aide et quitter
      --version     afficher des informations de version et quitter

Aide en ligne de GNU coreutils : <https://www.gnu.org/software/coreutils/>
Signalez les problèmes de traduction à : <traduc@traduc.org>
Documentation complète <https://www.gnu.org/software/coreutils/mkdir>
ou disponible localement via: info '(coreutils) mkdir invocation'
```

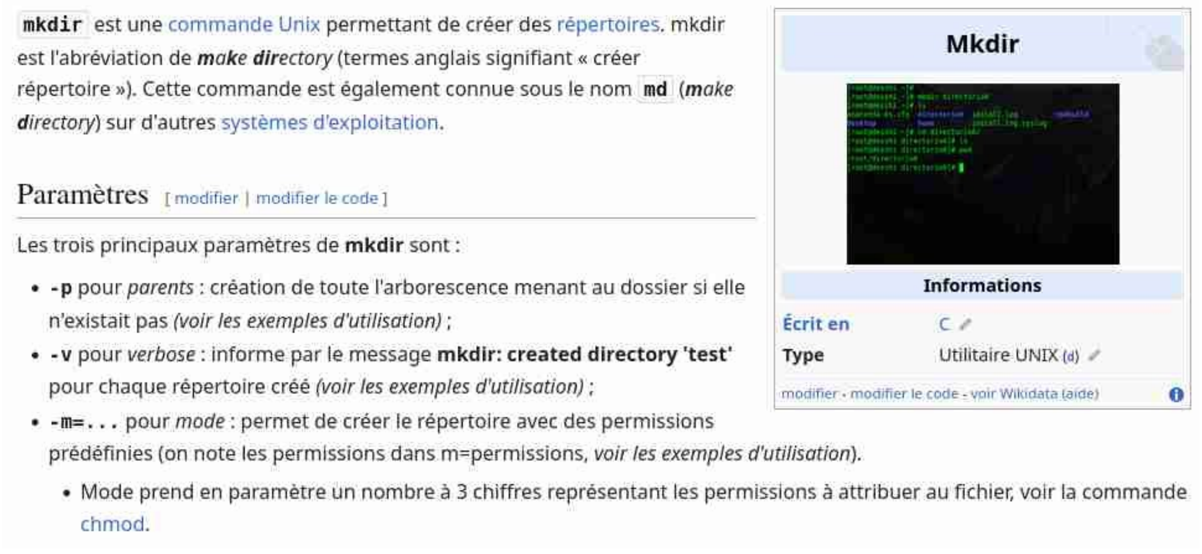
1. The command "whatis" displays a one-line description of the command following it, e.g. whatis ls.

Q2) Type “whatis mkdir” then copy the output here:



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ whatis mkdir
mkdir (1)            - make directories
mkdir (2)            - create a directory
```

1. Q3) Using your favorite search engine on the internet, e.g. www.duckduckgo.com, www.bing.com, www.google.com, type “mkdir linux”, then copy the beginning (first ten lines) of the first result in the web page here:



`mkdir` est une commande Unix permettant de créer des répertoires. mkdir est l'abréviation de ***make directory*** (termes anglais signifiant « créer répertoire »). Cette commande est également connue sous le nom `md` (*make directory*) sur d'autres systèmes d'exploitation.

Paramètres [modifier | modifier le code]

Les trois principaux paramètres de **mkdir** sont :

- **-p** pour *parents* : création de toute l'arborescence menant au dossier si elle n'existait pas *(voir les exemples d'utilisation)* ;
- **-v** pour *verbose* : informe par le message **mkdir: created directory 'test'** pour chaque répertoire créé *(voir les exemples d'utilisation)* ;
- **-m=...** pour *mode* : permet de créer le répertoire avec des permissions prédéfinies (on note les permissions dans m=permissions, *voir les exemples d'utilisation*).
  - Mode prend en paramètre un nombre à 3 chiffres représentant les permissions à attribuer au fichier, voir la commande chmod.

Mkdir

Informations

Écrit en	C
Type	Utilitaire UNIX (d)

modifier - modifier le code - voir Wikidata (aide)

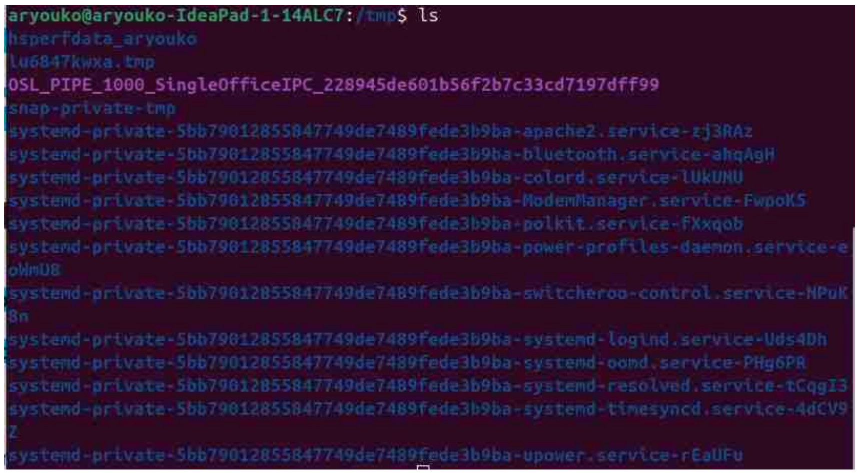
2. Asking a fellow human.

2.1.5 The command cd

— Q4) In few wors, find out what this command is for:

to change directory

— Go to the **tmp** directory at the root of the system and list its contents



```
aryouko@aryouko-IdeaPad-1-14ALC7:/tmp$ ls
hsperfdata_aryouko
lu6847kwxa.tmp
OSL_PIPE_1000_SingleOfficeIPC_228945de601b56f2b7c33cd7197dff99
snap-private-tmp
systemd-private-5bb79012855847749de7489fede3b9ba-apache2.service-zj3RAz
systemd-private-5bb79012855847749de7489fede3b9ba-bluetooth.service-ahqAgH
systemd-private-5bb79012855847749de7489fede3b9ba-colord.service-lUkUNU
systemd-private-5bb79012855847749de7489fede3b9ba-ModemManager.service-FwpoKS
systemd-private-5bb79012855847749de7489fede3b9ba-polkit.service-fXxqob
systemd-private-5bb79012855847749de7489fede3b9ba-power-profiles-daemon.service-e
oWmU8
systemd-private-5bb79012855847749de7489fede3b9ba-switcheroo-control.service-NPuK
8n
systemd-private-5bb79012855847749de7489fede3b9ba-systemd-logind.service-Uds4Dh
systemd-private-5bb79012855847749de7489fede3b9ba-systemd-oomd.service-PHg6PR
systemd-private-5bb79012855847749de7489fede3b9ba-systemd-resolved.service-tCqgI3
systemd-private-5bb79012855847749de7489fede3b9ba-systemd-timesyncd.service-4dCV9
Z
systemd-private-5bb79012855847749de7489fede3b9ba-upower.service-rEaUFu
```

— Q5) Return to your home directory, how do you do it ? Find two other ways to do it:

With cd or cd $HOME

— Q6) What does **cd -** do ?

Affiche le répertoire courant (le HOME)

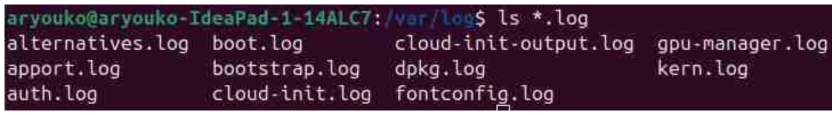
2.1.6 Wildcards

A **wildcard** is a symbol that takes the place of an unknown character or set of characters. Some of the heavily used wildcards are :

— The asterisk "*", represents any number of unknown characters. For example "*cake" can stand for "carrotcake, chocolatcake,cake,..." and "cake*" can stand for "cakeParty, cakeWorld,cake,...".

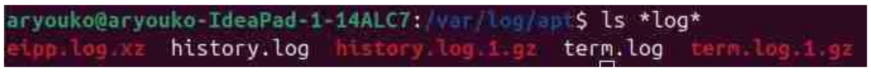
— The question mark " ?", represents only one character. For example "??ke" can stand for "cake,bake,take,. . ." and "ke??" can stand for "keen,kept,keep,. . .".

Q7) Use the "wildcard" "*" associated with ls command to list all the files ending with ".log" in "/var/log"



```
aryouko@aryouko-IdeaPad-1-14ALC7:/var/log$ ls *.log
alternatives.log  boot.log        cloud-init-output.log  gpu-manager.log
apport.log        bootstrap.log   dpkg.log               kern.log
auth.log          cloud-init.log  fontconfig.log
```

Q8) Use the "wildcard" "*" associated with ls command to list all the files containing the work “log” in their name under directory "/var/log/apt"



```
aryouko@aryouko-IdeaPad-1-14ALC7:/var/log/apt$ ls *log*
eipp.log.xz  history.log  history.log.1.gz  term.log  term.log.1.gz
```

2.1.7 Manuel of the command mkdir

Consider that we want to create at the root of your account the following path :

```
ArchiSys/tp/01
```

Launch the following command and observe what is happening :

```
cd;mkdir ArchiSys/tp/01
```

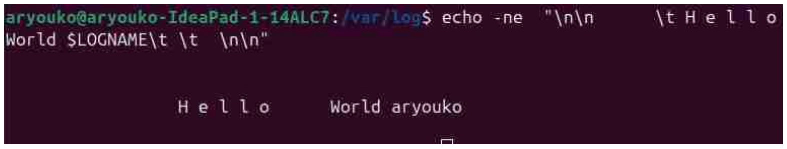
Look in the "mkdir" man page for the option to create the entire directory hierarchy in one command.

Q9) What is the exact command to use ?
Pour créer une architecture entière il faut utiliser le -p donc écrire mkdir -p ArchiSys/tp/01

2.2 The command echo

Q10) What is the command "echo" used for ?
La commande est utiliser pour afficher du texte
Q11) What does the following command do ?

```
echo    -ne  "\n\n      \t H e l l o  World $LOGNAME\t \t  \n\n"
```



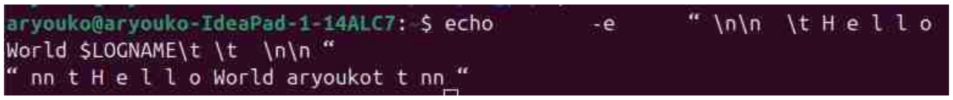
```
aryouko@aryouko-IdeaPad-1-14ALC7:/var/log$ echo -ne  "\n\n      \t H e l l o
World $LOGNAME\t \t  \n\n"


                H e l l o      World aryouko
```

Elle affiche avec des sauts de lignes et des alinéas

Q12) Try the following commands, what is the difference ? Why ?

```
echo    -e  “ \n\n      \t H e l l o  World $LOGNAME\t \t  \n\n “
```



```
aryouko@aryouko-IdeaPad-1-14ALC7: $ echo        -e      “ \n\n  \t H e l l o
World $LOGNAME\t \t  \n\n “
“ nn t H e l l o World aryoukot t nn “
```

L’option n’est pas la même sur les deux le -n sert à considérer les sauts de lignes,etc

2.3 Other commands

In the following, we will use the command "ps" which lists the programs running on the machine and "cat" which reads an input and prints it on the standard output. See the manual pages for these commands if they are not familiar to you.

3 Redirections

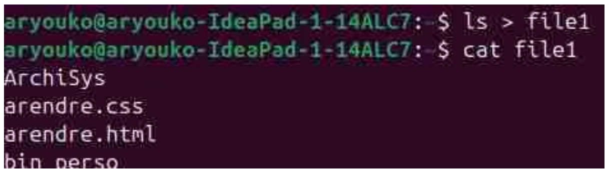
The commands that we have seen so far send their outputs to the so-called **standard output** (stdout). By default, that is the terminal. However, it is possible to change the destination of the standard output for a command and for example print the result of a command into a file (this practice is called redirection). Note that programs may also produce output in places other than standard output.

In general, most programs work as follows :
— if one or more files are specified, the program works on these files ;
— if no file is specified, standard input is used.

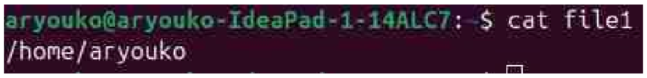
3.1 First steps in redirection

Q13) Execute each of the following commands(in order) in your terminal, examine the results and deduce what happens (it will be necessary to look at the contents of the files generated after each command) :

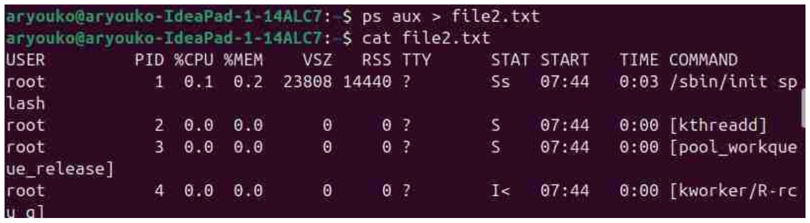
```
ls
ls > file1
pwd > file1
ps aux > file2.txt
cd > file3
cat file1 file2.txt > file4
cat file1 >> file1
pwd >> file1
cat file1 > /dev/null
```



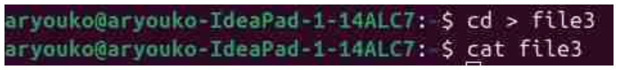
```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ls > file1
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file1
ArchiSys
arendre.css
arendre.html
bin_perso
```



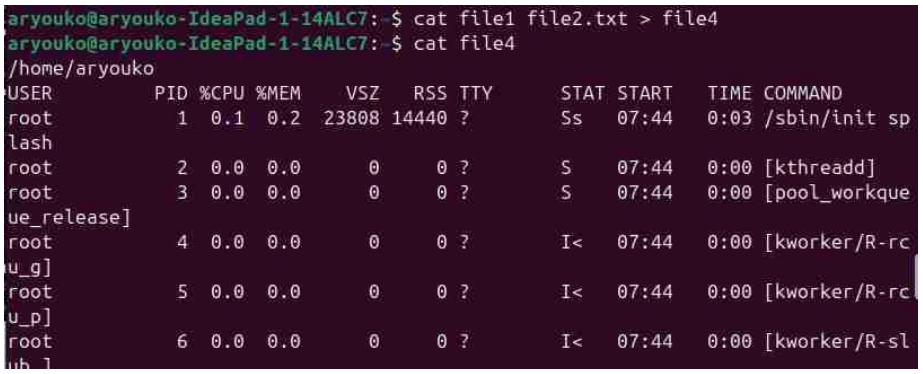
```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file1
/home/aryouko
```



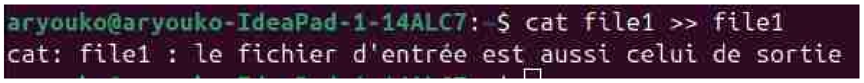
```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps aux > file2.txt
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file2.txt
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.2  23808 14440 ?        Ss   07:44   0:03 /sbin/init sp
lash
root           2  0.0  0.0      0     0 ?        S    07:44   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    07:44   0:00 [pool_workque
ue_release]
root           4  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-rc
u_g]
```



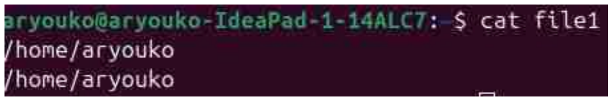
```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cd > file3
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file3
```



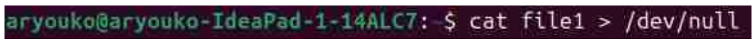
```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file1 file2.txt > file4
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file4
/home/aryouko
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.2  23808 14440 ?        Ss   07:44   0:03 /sbin/init sp
lash
root           2  0.0  0.0      0     0 ?        S    07:44   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    07:44   0:00 [pool_workque
ue_release]
root           4  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-rc
u_g]
root           5  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-rc
u_p]
root           6  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-sl
ub_]
```



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file1 >> file1
cat: file1 : le fichier d'entrée est aussi celui de sortie
```



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file1
/home/aryouko
/home/aryouko
```



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat file1 > /dev/null
```

Q14) What are the key words «>» and «>>» used in the previous examples ?

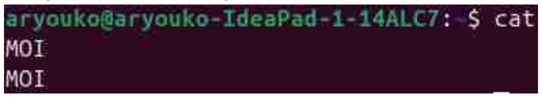
> sert à enregistrer la valeur vers le fichier en écrasant les données tant dis que >> additionne les données au lieu de les écraser.

3.2 The pipe

The pipe is also a form of redirection, but this time instead of redirecting the standard output to a file as before, we will redirect it to a command. Note that, just like for standard output, any program has a **"standard input"**, **"stdin"**, which may or may not be used. In the terminal, the standard input is often the keyboard.

3.2.1 Cat

Run the cat command in a terminal. Write a few words on the screen then press "Enter". The line is duplicated. Q15) Why ? (to finish press Ctrl + D)



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat
MOI
MOI
```

Ca l'écrit deux fois car il y a moi qui écrit ma phrase et le terminal qui l'affiche après

3.2.2 Examples of pipe

Q16) Execute each of the following commands(in order) on your device, examine the contents of the mani-pulated files and deduce what is happening :

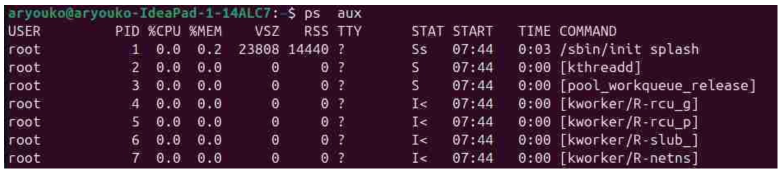
ls -R > file1 **Cela met la valeur de la commande ls -R dans file1, le -R permet d'afficher aussi les fichier et répertoires des répertoires (c'est récursif)**

less file1 **Cela permet d'afficher file1 de la manière d'une page**

cat file1 **Cela permet d'afficher le contenu de file1 dans le terminal**

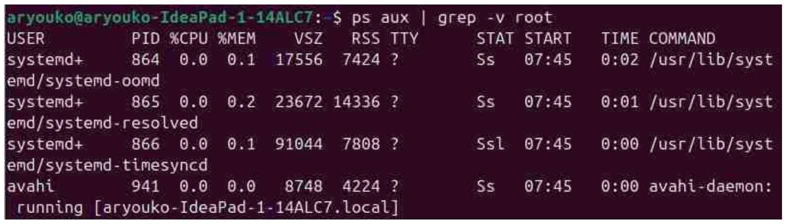
cat file1 | less **Cela permet d'afficher file1 de la manière d'une page mais dans le terminal**

ps aux **Cela permet d'afficher tout les processus en cours dans le système**



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps  aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2  23808 14440 ?        Ss   07:44   0:03 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    07:44   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    07:44   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-netns]
```

ps auwx | grep -v root **grep permet de filtrer et -v d'exclure ce qu'il y a après, ici cela exclu ce qui contient root**



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps aux | grep -v root
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
systemd+     864  0.0  0.1  17556  7424 ?        Ss   07:45   0:02 /usr/lib/syst
emd/systemd-oomd
systemd+     865  0.0  0.2  23672 14336 ?        Ss   07:45   0:01 /usr/lib/syst
emd/systemd-resolved
systemd+     866  0.0  0.1  91044  7808 ?        Ssl  07:45   0:00 /usr/lib/syst
emd/systemd-timesyncd
avahi        941  0.0  0.0   8748  4224 ?        Ss   07:45   0:00 avahi-daemon:
 running [aryouko-IdeaPad-1-14ALC7.local]
```

ps aux | grep root **Cela inclu le root**
ps aux > file2 **Cela enregistre le résultat de la commande ps aux dans file2**
cat file2 | grep -v root **Cela affiche que les lignes où il n'y a pas le mot root**

Additional questions :

Q17) What is the purpose of grep ? And its '-v' option ? **Grep sert à filtrer et l'option -v à exclure ce qu'on met après**

Q18) List the processes that do not belong to "root" but contain the root in their names and write them to a file with the name 'processwithroot' .

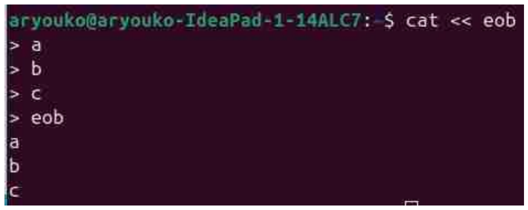
ps aux | grep -v '^root' | grep 'root' > processwithroot (le ^ permet de dire que c'est le propriétaire

3.3 Input redirection

Just as it is possible to redirect the output of a command, it is sometimes interesting to redirect its input.

Q19) Explain the behavior of the following command :

```
c a t <<eob
a
b
c
eob
```



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ cat << eob
> a
> b
> c
> eob
a
b
c
```

Tant qu’on écrit pas eob notre texte ne se fermera pas et donc ne s’affichera pas

Q20) What is the difference between **<**, **<<** and **<<<** ?

< permet de rediriger l’entrée d’un fichier vers une commande

<< permet de fournir un bloc de texte comme entrée à une commande. Ce texte est délimité par le mot spécifié après le <<

<<< permet de fournir une chaine de caractère unique comme entée

Note : Pressing "Ctrl+D" tells the shell that it reached the end of the text.

4 Bonus

This bonus is made of a game where you have level to pass. For passing each level, you must properly use Linux commands and proper understand the given explanation.

NOTICE: Please ensure that you're using Eduroam Wifi for connecting to the bandit.labs.overthewire.org host.

Level 0:

The goal of this level is for you to log into the game using SSH. The host to which you need to connect is **bandit.labs.overthewire.org**, on port 2220. The username is **bandit0** and the password is **bandit0**. Once logged in, go to the Level 1 page to find out how to beat Level 1.

Level 1:

The password for the next level is stored in a file called **readme** located in the home directory. Use this password to log as bandit1 using SSH. Whenever you find a password for a level, use SSH (on port 2220) to log into that level and continue the game.

Level 2:

The password for the next level is stored in a file called **-** located in the home directory

Level 3:

The password for the next level is stored in a file called **spaces in this filename** located in the home directory

Level 4:

The password for the next level is stored in a hidden file in the **inhere** directory.

Level 5:

The password for the next level is stored in the only human-readable file in the **inhere** directory. Tip: if your terminal is messed up, try the "reset" command.

Level 6:

The password for the next level is stored in a file somewhere under the **inhere** directory and has all of the following properties:

- human-readable
- 1033 bytes in size
- not executable

Level 7:

The password for the next level is stored **somewhere on the server** and has all of the following properties:

- owned by user bandit7
- owned by group bandit6
- 33 bytes in size

<u>Level 8:</u>

The password for the next level is stored in the file **data.txt** next to the word **millionth**

<u>Level 9:</u>

The password for the next level is stored in the file **data.txt** and is the only line of text that occurs only once

<u>Level 10:</u>

The password for the next level is stored in the file **data.txt** in one of the few human-readable strings, preceded by several '=' characters.