

Module :

Système d'exploitation et programmation système

**Compte Rendu :**

**TP N°1 : C-Shell & Processus**

**Réaliser par : EL HANAFI Maha**

## Objectifs :

- créer des alias, complétion et historique
- Comprendre le fonctionnement des processus

### Exercice 1 : Processus

- 1.1) Pour connaître les options possibles de la commande **>ps** on tape la commande **>man ps**.

Pour visualiser tous les processus en cours avec l'utilisateur propriétaire on tape **>ps aux**

```
uir_student@ubuntu: ~  
PS(1) User Commands PS(1)  
NAME  
ps - report a snapshot of the current processes.  
SYNOPSIS  
ps [options]  
DESCRIPTION  
ps displays information about a selection of the active processes. If  
you want a repetitive update of the selection and the displayed  
information, use top(1) instead.  
This version of ps accepts several kinds of options:  
1 UNIX options, which may be grouped and must be preceded by a dash.  
2 BSD options, which may be grouped and must not be used with a dash.  
3 GNU long options, which are preceded by two dashes.  
uir_student@ubuntu:~$ ps aux  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root         1  0.2  0.3  4472  3428 ?        Ss   17:51   0:09 /sbin/init  
root         2  0.0  0.0      0     0 ?        S    17:51   0:00 [kthreadd]  
root         3  0.0  0.0      0     0 ?        S    17:51   0:02 [ksoftirqd/0]  
root         4  0.0  0.0      0     0 ?        S    17:51   0:00 [kworker/0:0]  
root         5  0.0  0.0      0     0 ?        S<   17:51   0:00 [kworker/0:0H]  
root         7  0.2  0.0      0     0 ?        S    17:51   0:08 [rcu_sched]  
root         8  0.0  0.0      0     0 ?        S    17:51   0:00 [rcu_bh]  
root         9  0.0  0.0      0     0 ?        S    17:51   0:00 [migration/0]  
root        10  0.0  0.0      0     0 ?        S    17:51   0:00 [watchdog/0]  
root        11  0.1  0.0      0     0 ?        S    17:51   0:06 [watchdog/1]  
root        12  0.0  0.0      0     0 ?        S    17:51   0:00 [migration/1]  
root        13  0.0  0.0      0     0 ?        S    17:51   0:02 [ksoftirqd/1]  
uir_student@ubuntu: ~  
uir_student@ubuntu:~$ man ps  
uir_student@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 2754 pts/1        00:00:00 bash  
 3351 pts/1        00:00:00 ps  
uir_student@ubuntu:~$ ps &  
[1] 3352  
uir_student@ubuntu:~$  PID TTY          TIME CMD  
 2754 pts/1        00:00:00 bash  
 3352 pts/1        00:00:00 ps
```

- 1.2) Pour lancer gedit en premier plan on tape la commande **>gedit**  
**>ctrl+z** : mettre en pause l'exécution du programme

La commande **>jobs** liste seulement les processus qui tournent en fond au sein d'une même console

Pour que le processus continue à tourner mais en arrière-plan on utilise la commande **>bg %n** (n=le numéro fourni par jobs)

la commande fg reprend un processus au premier plan, on tape **>fg %n**

```
uir_student@ubuntu:~$ gedit
^Z
[1]+  Stopped                  gedit
uir_student@ubuntu:~$ jobs
[1]+  Stopped                  gedit
uir_student@ubuntu:~$ bg %1
[1]+  gedit &
uir_student@ubuntu:~$ fg %1
gedit
```

- 1.3) La commande **>man kill** pour lire le manuel de la commande kill.  
Cette dernière envoie le signal indiqué aux processus mentionnés

```
uir_student@ubuntu: ~
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available
    signals. Particularly useful signals include HUP, INT, KILL, STOP,
    CONT, and 0. Alternate signals may be specified in three ways: -9,
    -SIGKILL or -KILL. Negative PID values may be used to choose whole
    process groups; see the PGID column in ps command output. A PID of -1
    is special; it indicates all processes except the kill process itself
    and init.

uir_student@ubuntu:~$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
```

- 1.4) Pour terminer le processus exécutant gedit on utilise la commande  
**>kill -9 %n** (n=le numéro fourni par jobs) (ou **>kill -SIGKILL %n** )

```
uir_student@ubuntu:~$ jobs
[1]+  Stopped                  gedit
uir_student@ubuntu:~$ kill -9 %1

[1]+  Stopped                  gedit
uir_student@ubuntu:~$ ps
  PID TTY          TIME CMD
 2754 pts/1        00:00:00 bash
 3507 pts/1        00:00:00 ps
[1]+  Killed                  gedit
```

- 1.5)

```
uir_student@ubuntu:~$ sleep 5
uir_student@ubuntu:~$ sleep 5 &
[1] 3512
```

- 1.6) **Ctrl+alt+t** pour ouvrir une autre fenêtre terminale textuellement.

- 1.7)

```
uir_student@ubuntu:~$ sleep 30 &
[1] 3636
uir_student@ubuntu:~$ ps
  PID TTY          TIME CMD
 2754 pts/1        00:00:00 bash
 3636 pts/1        00:00:00 sleep
 3637 pts/1        00:00:00 ps
```

- 1.8) La commande **>kill -9 PID** tue le processus dont on connaît le PID

```
uir_student@ubuntu:~$ kill -9 3636
bash: kill: (3636) - No such process
[1]+  Done                    sleep 30
uir_student@ubuntu:~$ ps
  PID TTY          TIME CMD
 2754 pts/1        00:00:01 bash
 3640 pts/1        00:00:00 ps
uir_student@ubuntu:~$
```

## Exercice 2 : Historique

2.1) Pour visualiser toutes les commandes que j'ai tapé en utilisant la commande **>history**

```
765 fg %1
766 man kill
767 ps
768 jobs
769 gedit
770 ps
771 jobs
772 kill -9 %1
773 ps
774 sleep 5
775 sleep 5 &
776 man sleep
777 sleep 5
778 sleep 5 &
779 sleep help
780 sleep --help
781 sleep 30
782 sleep 30 &
783 ps
784 kill -9 3636
785 ps
786 kill -l
```

2.2)

```
uir_student@ubuntu:~$ history !784
history kill -9 3636
uir_student@ubuntu:~$ meta-!784
meta-kill -9 3636
```

2.3) Pour voir le nombre de commande qu'history peut stocker , on entre la commande

**>echo \$HISTSIZE**

```
uir_student@ubuntu:~$ echo $HISTSIZE
1000
```

Pour Fixer la taille d'historique à 20 commandes (ou bien changer ce nombre à 20) on tape la commande **>export HISTSIZE=20**

```
uir_student@ubuntu:~$ export HISTSIZE=20
uir_student@ubuntu:~$ echo $HISTSIZE
20
uir_student@ubuntu:~$
```

2.4 ) Lorsqu'on retape history, il s'affiche que les 20 dernières commandes

```
uir_student@ubuntu:~$ echo $HISTSIZE
20
uir_student@ubuntu:~$ history
779  sleep --help
780  sleep 30
781  sleep 30 &
782  ps
783  kill -9 3636
784  ps
785  kill -l
786  history
787  history meta-!
788  history meta-! 1
789  history meta-! n
790  history meta-! 787
791  history kill -9 3636
792  meta-!
793  meta-kill -9 3636
794  echo $HISTSIZE
795  echo HISTSIZE=20
796  export HISTSIZE=20
797  echo $HISTSIZE
798  history
```

### Exercice 3 : Complétion et Alias

3.1) Utilisant la complétion pour afficher toutes les commandes du répertoire /usr/bin qui commencent par ch

```
uir_student@ubuntu:~$ ls ch* /usr/sbin
ch  ch1  ch1.c  chatserver  chatserver.c  chatserver.c~  ch.c  ch.c~
```

3.2) créer des alias :

```
uir_student@ubuntu:~$ alias m='more'
uir_student@ubuntu:~$ alias del='rm -i'
uir_student@ubuntu:~$ alias ll='ls -al'
uir_student@ubuntu:~$ alias lr='ls -R'
uir_student@ubuntu:~$
```

exemple : >alias m='more'

**alias** : la commande qui nous permet de créer un alias

**m** : c'est l'alias, la commande que je veux saisir à la place d'une commande plus grande

**'more'** : c'est la commande qui sera exécutée lors de la saisie des alias

### Exercice 4 : Métacaractère

-Dans ce répertoire choisi n'existe pas les sources de programme C++  
en affichant les fichiers qui se terminent par .c utilisant la commande >ls \*.c

```
f10.c      f3.sh~   fich2.txt  p5.c~     s5.c~     test2.c~
f10.c~    f4.c~    file1      p66.c~    sc2.sh    TPUnix
f10.sh    f4.c~    file10     p.c~      sc2.sh~   Untitled Document 1
f10.sh~   f4.sh    file11     Pictures  scripts   Untitled Document 1~
f11.c     f4.sh~   file12     pipe1.c~  she2~    Videos
f11.c~    f5.c     file13     pr1.c~    she4~    vmware-tools-distrib

uir_student@ubuntu:~$ ls *.cc
ls: cannot access *.cc: No such file or directory
uir_student@ubuntu:~$ ls *.c
ch1.c      ch.c      client2.c  f11.c  f13.c  f2.c  f4.c  f6.c  f8.c  p1.c  test1.c
chatserver.c client1.c f10.c     f12.c  f1.c  f3.c  f5.c  f7.c  f9.c  t1.c  test2.c
```

-Les fichiers qui commencent par fic utilisant la commande >ls fic\*

```
uir_student@ubuntu:~/Desktop$ cd ..
uir_student@ubuntu:~$ ls fic*
fich1.txt fich1.txt~ fich2.txt
uir_student@ubuntu:~$
```

-afficher les fichier de la forme fichx.c x est un chiffre (de 0 à 9)

```
uir_student@ubuntu:~$ ls *.c [!0-9]
?      chatserver.c client1.c f10.c f12.c f1.c f3.c f5.c f7.c f9.c t1.c test2.c
ch1.c ch.c      client2.c f11.c f13.c f2.c f4.c f6.c f8.c p1.c test1.c
```

-les fichiers qui commencent par une majuscule comprise entre Bet I

```
uir_student@ubuntu:~$ ls [B-I]*
ch      ch.c~    f10.c~   f12.c   f1.sh~  f3.sh   f5.c~   f7.sh~  f9.sh   file12  file8
ch1     client1  f10.sh   f12.c~  f2.c    f3.sh~  f5.sh   f8.c    fich1.txt file13  file9
ch1.c   client1.c f10.sh~  f13.c   f2.c~   f4.c    f6.c    f8.c~   fich1.txt~ file2
chatserver client1.c~ f11.c   f13.c~  f2.sh   f4.c~   f6.sh   f8.sh   fich2.txt file3
chatserver.c client2   f11.c~  f1.c    f2.sh~  f4.sh   f7.c    f8.sh~  file1    file4
chatserver.c~ client2.c f11.sh   f1.c~   f3.c    f4.sh~  f7.c~   f9.c    file10   file5
ch.c     f10.c    f11.sh~  f1.sh   f3.c~   f5.c    f7.sh   f9.c~   file11   file7

Desktop:
folder ns3 ns-allinone-3.30.1 ns-allinone-3.30.1.tar.bz2 Old Firefox Data WIRELESS.pdf

Documents:
```

-Les fichiers dont le nom comporte au moins un caractère compris entre p et w

```
uir_student@ubuntu:~$ ls *[pw]
Desktop:
folder ns3 ns-allinone-3.30.1 ns-allinone-3.30.1.tar.bz2 Old Firefox Data WIRELESS.pdf

new:
rep3
uir_student@ubuntu:~$
```

-les fichiers dont les noms se terminent par une minuscule comprise entre l et p

```
uir_student@ubuntu:~$ ls [l-p]*
new1 p1 p1.c~ p3.c~ p5.c~ p.c~ pr1.c~ pr3.c~ pr5.c~ pr7.c~ prog2.c~
new2 p1.c p2.c~ p4.c~ p66.c~ pipe1.c~ pr2.c~ pr4.c~ pr6.c~ prog1.c~ prog3.c~

Music:

new:
rep3
```

-les fichiers dont le nom commence par une minuscule comprise entre b et f et  
le nom commence par une minuscule comprise entre r et u

```
uir_student@ubuntu:~$ ls [b-f]* ; ls [r-u]*
ch          ch.c~      f10.c~     f12.c      f1.sh~     f3.sh      f5.c~     f7.sh~     f9.sh      file12     file8
ch1         client1    f10.sh     f12.c~     f2.c~      f3.sh~     f5.sh     f8.c      fich1.txt  file13     file9
ch1.c       client1.c  f10.sh~    f13.c      f2.c~      f4.c~      f6.c      f8.c~     fich1.txt~ file2
chatserver  client1.c~ f11.c      f13.c~     f2.sh      f4.c~      f6.sh     f8.sh     fich2.txt  file3
chatserver.c client2    f11.c~     f1.c       f2.sh~     f4.sh      f7.c      f8.sh~    file1      file4
chatserver.c~ client2.c  f11.sh     f1.c~      f3.c       f4.sh~     f7.c~     f9.c      file10     file5
ch.c        f10.c     f11.sh~    f1.sh      f3.c~      f5.c       f7.sh     f9.c~     file11     file7

Desktop:
folder ns3 ns-allinone-3.30.1 ns-allinone-3.30.1.tar.bz2 Old Firefox Data WIRELESS.pdf

Documents:

Downloads:
s1.c~ s4.c~ sc2.sh she2~ she5~ signal1.c~ t1.c t2.c~ t4.c~ test test1.c~ test2.c~
s2.c~ s5.c~ sc2.sh~ she4~ shell3~ t1 t1.c~ t3.c~ t.c~ test1.c test2.c

r1:
f1.txt

r2:

rep1:
new
```