

## 1. Gestion des processus

Il faut se connecter sur une console virtuelle texte en tant qu'utilisateur ubuntu.

### 1.1. Visualisation des processus

1. Consulter la page du manuel en ligne (man) concernant la commande ps.

**« man ps »**

2. Lister les processus lancés à partir de votre shell courant.

**« ps »**

3. Lister tous vos processus s'exécutant actuellement sur le système.

**« ps -e » ou « ps -A » (man indique que -A est identique à -e)**

PID	TTY	TIME	CMD
-----	-----	------	-----

4. Lister tous les processus lancés sur le système (démons inclus) en affichant l'identité sous laquelle ils s'exécutent (UID)

(Indication : cumuler plusieurs options de la commande ps : affichage de tous les processus non rattachés à un terminal (démons), affiche de l'UID sous lequel les processus s'exécutent).

**« ps -ef »**

UID	PID	PPID	C	STIME	TTY	TIME	CMD
-----	-----	------	---	-------	-----	------	-----

**« ps -eF »**

UID	PID	PPID	SZ	RSS	PSR	C	STIME	TTY	TIME	CMD
-----	-----	------	----	-----	-----	---	-------	-----	------	-----

5. Lister de nouveau tous les processus s'exécutant sur le système de façon détaillée en employant d'autres options de la commande ps (Indication : cumuler plusieurs options de la commande ps : affichage de tous les processus, affichage détaillé).

**« ps -aux »**

### 1.2. Signaux et commande kill

1. Lister les signaux pouvant être envoyés aux processus (Indication : utiliser la commande kill avec l'option adéquate).

**« kill -l »**

2. A quoi correspondent ces signaux ?

**La liste des signaux obtenus en utilisant kill -l sont ceux qui peuvent être envoyés à un processus par la commande kill**

3. Essayer d'arrêter un processus appartenant à un autre utilisateur. Est-ce possible ?  
Pour cela, créer un utilisateur (« tux » par exemple), se connecter sous « tux » sur une console virtuelle, puis supprimer le processus « bash » de « linux »

**sudo useradd tux**

*sudo passwd tux*

*« Ctrl-Alt-F2 »*

*kill 2174*

*bash: kill: (2174) - Opération non permise*

*Nous ne pouvons pas arrêter un processus lancé par un autre utilisateur*

4. Sous l'utilisateur « tux », lancer la commande : sleep 999. Revenir sur la console de l'utilisateur « linux » et envoyer le signal SIGTERM au processus sleep 999 (en mode super utilisateur). Retourner sur la console de « tux » et observer le résultat

*sudo kill -sigterm 2417*

*Complété*

### **1.3. Commande top**

1. Afficher une liste rafraîchie en temps réel des processus s'exécutant sur le système.

*top - 14:21:41 up 48 min, 3 users, load average: 0.00, 0.03, 0.05*

*Tasks: 146 total, 2 running, 143 sleeping, 0 stopped, 1 zombie*

*Cpu(s): 0.8%us, 0.5%sy, 0.0%ni, 98.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st*

*Mem: 3563916k total, 1081876k used, 2482040k free, 66944k buffers*

*Swap: 0k total, 0k used, 0k free, 635980k cached*

*Se rafraîchit toutes les 3s.*

2. Afficher l'aide incluse avec la commande top.

*man top*

3. Ajouter la colonne PPID dans l'affichage de la commande top (Indication : utiliser le manuel en ligne man concernant la commande top).

*Appuyer d'abord sur F, puis naviguer avec les flèches, activer PPID avec espace, sortir avec ESC*

4. Afficher maintenant uniquement les processus de l'utilisateur ubuntu.

*top -u ubuntu*

5. Aller dans une autre console connectée sous le même utilisateur (« linux » et envoyer le signal 2 (SIGINT) au processus top que vous utilisez actuellement sans quitter la commande top (en super utilisateur). Que se passe t'il ?

*Cela a arrêté la mise à jour des processus*

### **1.4. Manipulation des processus**

1. Lancer la commande sleep 1111 en arrière-plan (Indication : un processus est lancé en arrière-plan lorsqu'on ajoute le caractère & en fin de ligne de commandes). Que signifie ce qui est retourné à l'écran ?

***sleep 1111 &***

***[1] 2447***

***Cela représente le PPID du processus exécuté en fond.***

2. Exécuter toujours en arrière-plan les commandes sleep 2222 et sleep 3333.
3. Lancer en avant-plan la commande sleep 4444. Puis faire en sorte de récupérer la main sur votre shell courant (Indication : ctrl-Z permet de suspendre un processus en avant-plan).
4. Lister les travaux (jobs) lancés à partir de votre shell courant (Indication : utiliser la commande jobs).

***jobs***

***[1] En cours d'exécution sleep 1111 &***

***[2] En cours d'exécution sleep 2222 &***

***[3]- En cours d'exécution sleep 3333 &***

***[4]+ Stoppé sleep 4444***

5. Que signifient les commandes + et – dans la liste précédente ?

***Le + signifie que le processus est privilégié, le -, celui qui le sera une fois le + stoppé***

6. Relancer l'exécution du processus, suspendu précédemment, en arrière-plan (Indication : utiliser la commande bg).

***ubuntu@ubuntu-HP-Compaq-dx2420-Microtower:~\$ bg***

***[4]+ sleep 4444 &***

7. Lancer la commande man sleep et se positionner à sa fin.
8. Suspendre le processus man sans l'arrêter.

***Ctrl + z***

9. Arrêter le processus sleep 1111 en utilisant son numéro de travail (Indication : utiliser les commandes jobs et kill).

***kill -9 %2***

***jobs***

***[2] Processus arrêté sleep 2222***

10. Arrêter le processus sleep 2222 en utilisant cette fois son PID (Indication : utiliser les commandes ps et kill).

***PID TTY TIME CMD***

***2279 pts/1 00:00:00 bash***

***2451 pts/1 00:00:00 sleep***

***2455 pts/1 00:00:00 man***

***2467 pts/1 00:00:00 pager***

***2475 pts/1 00:00:00 sleep***

***485 pts/1 00:00:00 ps***

***[3] Complété sleep 3333***

11. Arrêter le processus sleep 3333 avec la séquence de touches ctrl-C (Important : relancer préalablement le travail correspondant en avant-plan avec la commande fg).

***fg***

***man sleep***

12. Retourner sur la page du manuel précédemment ouverte. Vous trouvez-vous toujours au même endroit dans la page (Indication : utiliser la commande fg) ?
13. Quitter le manuel en ligne.

## 2. Shell Bash

Il faut se connecter sur une console virtuelle texte en tant qu'utilisateur ubuntu.

### 2.1. Variables

1. Lister toutes les variables définies dans votre environnement shell.

***Commande env***

2. Affecter la chaîne de caractères « abc » à la variable var1. Puis afficher de nouveau la liste des variables dans l'environnement shell.

***var1=abc***

3. Afficher le contenu de la variable var1.

***echo \$var1***

***abc***

4. Afficher le contenu de la variable var2. Cela génère-t'il une erreur ?

***echo \$var2***

***Affiche une ligne vide***

5. Afficher le contenu de la variable HOME. Que représente cette variable ?

***echo \$HOME***

***/home/ubuntu***

***Représente l'adresse du dossier de l'utilisateur***

6. Modifier la valeur de la variable HOME par « /tmp ». Puis exécuter la commande cd et afficher votre répertoire de travail.

***HOME=/tmp***

***cd***

***pwd***

***/tmp***

7. Modifier la valeur de la variable LANG par « C ». Puis consulter la page du manuel électronique concernant la commande bash. Dans quelle langue est affichée la page du manuel ?

***LANG=C***

***Le manuel est maintenant en anglais***

8. Se déconnecter et se reconnecter sur la première console virtuelle texte en tant qu'utilisateur linux. Puis vérifier les valeurs des variables précédemment modifiées.

***echo \$LANG fr\_FR.UTF-8***

***Elles sont revenues à leurs valeurs initiales (en effet, on a pas utilisé export pour que les valeurs soient globales)***

9. Affecter la chaîne de caractères « Bond » à la variable var3.

***var3=Bond***

10. Afficher le texte « Bond, James Bond » en utilisant le contenu de la variable var3.

***echo \$var3, James \$var3***

***Bond, James Bond***

11. Afficher la chaîne de caractères « Bondir » en utilisant le contenu de la variable var3  
(Indication : lorsque le nom d'une variable est ambigu sur la ligne de commande, il est possible de délimiter son nom avec la syntaxe \${ } ).

***echo \${var3}ir Bondir***

12. Affecter la chaîne de caractères « lun » à la variable var4 et la chaîne de caractères « di » à la variable var5. Puis affecter la chaîne de caractères « lundi » à la variable var6 en réutilisant le contenu des variables var4 et var5.

***var4=lun***

***var5=di***

***var6=\${var4}\${var5}***

***echo \$var6***

***lundi***

13. Supprimer les variables var4, var5 et var6 de votre environnement shell  
(Indication : utiliser la commande unset).

***unset var4***

***unset var5 var6***

14. Lancer un nouveau shell avec la commande bash. Est-ce que la variable var3 est définie dans le nouvel environnement shell ? Puis revenir dans votre shell précédent.

***Var3 est une variable d'environnement, elle n'existe pas dans un nouveau shell.***

15. Faire en sorte que la variable var3 soit définie dans les environnements des processus fils et vérifier (Indication : utiliser la commande export).

***var3=toto***

***export var3***

***bash***

***echo \$var3***

***toto***

## **2.2. Jokers ou caractères génériques**

1. Aller dans le répertoire /etc.

***cd /etc***

2. Lister tous les fichiers dont le nom commence par la lettre « r ». Le résultat de la commande est-il bien celui attendu (Indication : utiliser le caractère générique \*) ?

***ls r\****

***rc.local resolv.conf rmt rpc rsyslog.conf***

***rc0.d:***

***K20speech-dispatcher S10unattended-upgrades S31umountnfs.sh S60umountroot***

***K74bluetooth S20endsigs S35networking S90halt***

***README S30urandom S40umountfs***

***Cela liste aussi les fichiers dans les répertoires commençant par r.***

3. Lister de nouveau tous les fichiers dont le nom commence par la lettre « r » sans afficher le contenu des répertoires correspondants (Indication : utiliser la commande ls avec l'option adéquate).

***ls r\* -d***

***rc0.d rc2.d rc4.d rc6.d rc5.d resolv.conf rpc rsyslog.d***

***rc1.d rc3.d rc5.d rc.local resolv.conf rmt rsyslog.conf***

4. Afficher tous les fichiers dont le nom contient la chaîne de caractères « rc ».

***ls \*rc\****

5. Afficher tous les fichiers dont le nom comporte trois caractères (Indication : utiliser le caractère générique ?).

***ls ???***

***rmt rpc***

***(...)***

6. Afficher tous les fichiers dont le nom commence par la chaîne de caractères « rc » suivie d'un caractère quelconque et se terminant par la chaîne de caractère « .d ».

***ls (-f) rc\*.d***

***rc0.d:***

***S40umountfs README . K74bluetooth***

***. S90halt S20endsigs***

***S35networking S10unattended-upgrades S30urandom***

***S31umountnfs.sh S60umountroot K20speech-dispatcher***

7. Afficher les fichiers dont les noms sont rc2.d, rc3.d et rc4.d sans les préciser explicitement individuellement (Indication : utiliser les caractères génériques [ et ]).

***ls rc[2-4].d***

***rc2.d:***

***README S50pulseaudio S70pppd-dns S99grub-common***

***S20kerneloops S50rsync S75sudo S99ondemand***

***S20speech-dispatcher S50saned S90binfmt-support S99rc.local***

***S25bluetooth S70dns-clean S99acpi-support***

Afficher tous les fichiers dont le nom ne commence pas par les lettres « a », « b » et « c »  
(indication : utiliser ls | egrep ...)

8.

```
ls | egrep -e^[^a-c].*$  
ConsoleKit  
dbus-1  
debconf.conf  
debian_version  
(...)
```

*Aussi : ls -h | egrep '^[^abc].\*'*

### 2.3. Redirections et tubes

1. Exécuter la commande cat sans arguments (cat suivi de la touche ENTREE) puis taper quelques mots (chacun terminé par la touche ENTREE) et terminer-la normalement (Indication : utiliser les touches Ctrl-Z). Que constatez-vous ?

```
cat  
test  
test  
blabla  
blabla  
truc  
truc  
^Z
```

**[1]+ Stoppé**                      **cat**

*Les mots tapés sont affichés à la ligne en dessous.*

*On a pu écrire librement comme dans un éditeur de texte. Le texte écrit est recopié sur la sortie standard*

2. Exécuter la commande wc sans arguments puis taper quelques mots et terminer-la normalement. Que constatez-vous ?

```
wc  
blabla  
test  
truc  
^Z
```

**[2]+ Stoppé**                      **wc**

*Les messages ne sont pas re-affichés.*

*Idem sauf que le texte écrit n'est pas recopié.*

3. Utiliser la commande cat pour afficher le contenu du fichier /etc/hosts de deux manières : en passant le nom du fichier en argument puis en utilisant une redirection vers un fichier fic1.

```
cat /etc/hosts  
cat /etc/hosts > fic1  
cat fic1
```

4. Utiliser la commande wc pour compter le nombre de lignes du fichier /etc/passwd de deux manières : en passant le nom du fichier en argument puis en utilisant une redirection. Que constatez-vous ?

```
wc /etc/hosts  
9 25 249 /etc/hosts
```

```
Il y a donc 9 lignes.  
wc /etc/hosts > fic1
```

```
Le fichier « fic1 » contient désormais :  
9 25 249 /etc/hosts
```

5. Utiliser la commande cat pour écrire quelques mots dans le fichier /tmp/ficcat (Indication : rediriger la sortie de la commande avec le caractère >).

```
cat > ficcat  
test  
toto  
^C
```

```
cat ficcat  
test  
toto
```

6. Utiliser la commande cat sans argument pour copier le fichier /tmp/ficcat en /tmp/ficcat2 (Indication : rediriger l'entrée de la commande avec le caractère < et la sortie avec le caractère >).

```
cat < /tmp/ficcat > /tmp/ficcat2  
cat ficcat2  
test  
toto
```

7. Utiliser la commande cat pour ajouter quelques mots au fichier /tmp/ficcat existant (Indication : rediriger la sortie de la commande avec les caractères >>).

```
cat >> ficcat  
mots ajoutés  
^C
```

```
cat ficcat  
test  
toto
```



***mots ajoutés***

8. Utiliser la commande cat pour concaténer le contenu des fichiers /tmp/ficcat et /etc/hosts dans le fichier /tmp/ficcat3.

***cat /tmp/ficcat /etc/hosts > /tmp/ficcat3***

***cat ficcat3***

9. Aller dans le répertoire /etc et rediriger la sortie de la commande ls dans le fichier /tmp/ls.out. Que contient ce fichier ?

***ls > ls.out***

***Le contenu de la commande ls (depuis le dossier /etc)***

10. Utiliser la commande wc sans argument pour compter le nombre de lignes contenues dans le fichier /tmp/ls.out.

***wc /tmp/ls.out***

***223 223 2187 /tmp/ls.out***

***223 lignes***

11. Combien y-a-t'il de fichiers dans le répertoire /etc ? Supprimer le fichier /tmp/ls.out.

***223 fichiers.***

***rm /tmp/ls.out***

12. Sans utiliser de fichier intermédiaire, compter de nouveau le nombre de fichiers présents dans le répertoire /etc (Indication : utiliser un pipe).

***ls | wc***

***223 223 2187***

13. En employant un tube (ou « pipe »), compter de nouveau le nombre de fichiers présents dans le répertoire /etc tout en écrivant le résultat de la commande ls dans le fichier /tmp/ls.out

(Indication : utiliser la commande tee).

***ls | tee /tmp/ls.out | wc***

***223 223 2187***

***Affiche à l'écran le résultat de wc et écrit dans ls.out le résultat de ls***

14. Exécuter la commande « ls /etc/passwd flop » et noter son résultat.

***ls /etc/passwd flop***

***ls: impossible d'accéder à flop: Aucun fichier ou dossier de ce type /etc/passwd***

15. Exécuter la commande « ls /etc/passwd flop » en redirigeant les sorties dans le fichier /tmp/ls.out et les messages dans le fichier /tmp/ls.err (Indication : rediriger la sortie de la commande avec le caractère > et l'erreur avec les caractères 2>).

***ls /etc/passwd flop >/tmp/ls.out 2> /tmp/ls.err***

***cat /tmp/ls.err***

***ls: impossible d'accéder à flop: Aucun fichier ou dossier de ce type***

```
cat /tmp/ls.out  
/etc/passwd
```

16. Exécuter la commande « `ls /etc/passwd flop` » en redirigeant les sorties dans le fichier `/tmp/ls.out` et en supprimant les messages d'erreur (Indication : utiliser le fichier spécial `/dev/null`).

```
ls /etc/passwd flop > /tmp/ls.out 2> /dev/null
```

```
cat /tmp/ls.out/etc/passwd  
cat /dev/null
```

17. Exécuter la commande « `ls /etc/passwd flop` » en redirigeant les sorties et les messages d'erreur dans le fichier `/tmp/ls.out` (Indication : rediriger les messages d'erreur de la commande au même endroit que la sortie avec les caractères `2>&1`).

```
ls /etc/passwd flop > /tmp/ls.out 2>&1
```

```
cat /tmp/ls.out  
ls: impossible d'accéder à flop: Aucun fichier ou dossier de ce type  
  
/etc/passwd
```