TD d'initiation à la ligne de commande Unix

dérnières modifications : 20 oct. 2003

Stéphane Salès
s.sales@tuxz.org

Table des matières

1.Nomenclature....2
1.1.Deux premières bonnes habitudes à prendre....2
1.1.1.La touche tabulation....2
1.1.2.RTFM....3
1.2.Pour bien comencer....4
1.2.1.Commandes de base....4
1.2.2.Manipulation :....6
1.2.3.Comme sous DOS les variables globales....6
1.2.4.Les redirections....7
2.Les processus :....8
2.1.La commande ps :....8
2.2.La commande kill :....9
2.3.La commande jobs :....9
3.Amusons nous un peu :....9
3.1.Netcat :....9
3.1.1.finir cette partie....9
4.Sources :....10

1.Nomenclature

Du texte affiché à l'écran :

```
texte
```

La touche «x» est préssée :

<x>

La comande toto est éxecutée en tant qu'utilisateur classique

```
$ toto
```

La comande toto est éxecutée en tant que super-utilisateur

```
# toto
```

1.1.Deux premières bonnes habitudes à prendre

1.1.1.La touche tabulation

La touche tabulation met en oeuvre une des fonctions très appréciables du shell, appelée la complétion. Si je tape le début d'une commande et que j'appuie sur <tabulation>, si il n'y a qu'une seule commande qui commence par ce que je viens de taper alors la commande est complété :

```
$ ls
fichier1 fichier12 fichier21 fichier212 crespa-
fichier
```

```
$ ls cre
```

à cette instant on presse la touche <tabulation>

```
$ ls crespa-fichier
```

en effet le seul fichier commençant par cre étant crespa-fichier la seule pression sur la touche <tabulation> à complété le mot

Essayons maintenant :

```
$ ls fic
```

<tabulation>

```
$ ls fichier
```

la complétion à bien fonctionné mais c'est arrêté car plusieurs fichiers commencent par le même motif. Si maintenant on appuie 2 fois sur <tabulation> :

```
fichier1 fichier12 fichier21 fichier212
```

le shell nous donne la liste des fichiers qui commencent par le motif tapé.

Essayez par vous même :

1 ***Créez les fichiers fichier1 fichier12 fichier21 fichier212 et crespa-fichier puis essayer :***

```
$ > fichier1
$ > fichier12
etc
```

puis

```
ls cre
```

<tabulation>

```
ls fic
```

<tabulation> observez puis tapez de nouveau <tabulation>

Désormais il sera inexcusable de votre part toute erreur du type «command not found» ou «file not found».

1.1.2.RTFM

RTFM est un acronyme de «Read The Fine Manual(il y a une version plus hard)»

Une rumeur de fond de couloir essaye de faire croire qu'un système GNU/Linux est difficile à utiliser, certes pour un utilisateur habitué à windows, la transition vers GNU/Linux peut etre difficile notamment car il lui à été donné de mauvaises habitudes, dont la plus génante étant celle de ne pas lire .

Un des gros avantages des systèmes GNU/Linux est que la documentation est plus qu'abondante, sur le net, mais aussi et surtout sur le système en lui même! En effet une documentation des plus complètes est accesible directement via la commande «**man**»

La commande man s'utilise ainsi :

man commande

man fichier

ainsi si je veux de l'aide sur la commande ls je taperais :

```
$ man ls
```

si je veux de l'aide sur le fichier de configuration «xinetd.conf» je taperais :

```
$ man xinetd.Conf
```

Essayer par vous meme :

2 ***Il existe plusieurs catégories de man combien ?***

Une meme commande (kill par exemple) peut exister dans plusieurs catégories de man :

3 ***Comment lui faire afficher tous les mans de kill ?***

4 ***Coment lui faire afficher uniquement le man de l'appel système kill() ?***

5 ***Comment lui faire afficher une description brève des***

différents mans de kill ?

ET ON OUBLIE PAS RTFM (bien entendu la documentation de la commande man est accessible par «man man»)

Désormais il sera inexcusable de votre part toute interrogation du style «mais quel est la syntaxe de cette commande ?**»**

1.2.Pour bien comencer

Attention :

- Le shell est sensible à la casse, le fichier MonFichier n'est pas le même que le fichier monfichier
- TOUT est un fichier sous Unix : une imprimante est un fichier, l'écran, la carte son etc

1.2.1.Commandes de base

Le minimum à connaître

Donner un descriptif accompagné d'exemple(s) des commandes suivantes :

6 ***cd***

7 ***ls***

8 ***cp***

cp -i

9 ***mv***

10 ***rm***

11 ***pwd***

12 ***mkdir***

13 ***mkdirhier***

14 ***cat***

15 ***less***

16 ***file***

17 ***tar***

18 ***ps***

19 ***top***

20 ***kill***

21 ***locate***

22 ***grep***

23 ***cut***

24 ***tail***

25 ***head***

26 ***wc***

La commande ls

Quels sont les effets des paramètres suivants sur la commande 'ls':

27 ***-l***

28 ***-a***

29 ***-t***

30 ***-F***

31 ***--color***

32 ***Donnez la commande pour afficher la liste détaillée du contenu(hors fichier caché) d'un répertoire***

33 ***Donnez la commande pour afficher la liste détaillée du contenu(fichier caché y compris) d'un répertoire***

34 ***Donnez la commande pour afficher la liste détaillée du contenu(fichier caché y compris) d'un répertoire, triée par date de modification***

35 ***Donnez la commande pour afficher la liste détaillée du contenu(fichier caché y compris) d'un répertoire, triée par date de modification du moins récent au plus récent***

36 ***Donnez la commande pour afficher la liste détaillée du contenu(fichier caché y compris) d'un répertoire, en couleur et avec le suffixe indiquant le type de chaque fichier***

La commande tar

Quels sont les effets des paramètres suivants sur la commande «tar»:

37 ***-c***

38 ***-x***

39 ***-f***

40 ***-z***

41 ***-j***

42 ***-t***

43 ***Donnez la commande permettant de créer l'archive archive.tar à partir du répertoire archive***

44 ***Donnez la commande permettant de créer l'archive archive.tar.gz à partir du répertoire archive***

45 ***Donnez la commande permettant de créer l'archive archive.tar.bz2 à partir du répertoire archive***

46 ***Donnez la commande permettant de lister le contenu de l'archive archive.tar.gz***

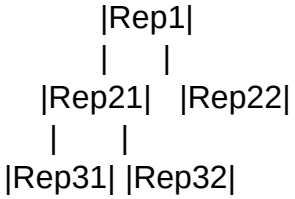
47 ***Donnez la commande permettant d'extraire l'archive archive.tar***

48 ***Donnez la commande permettant d'extraire l'archive archive.tar.gz***

49 ***Donnez la commande permettant d'extraire l'archive archive.tar.bz2***

1.2.2.Manipulation :

50 ***Créez l'arborescence suivante dans votre répertoire personnel:***



51 ***Créez une archive archive.tar de votre arborescence dans votre répertoire personnel***

52 ***Placez vous dans Rep1 et n'en bougez plus jusqu'à la fin de cette exercice***

53 ***Créez 2 fichiers vides essai311 et essai312 dans le répertoire Rep31 à l'aide de la commande touch***

54 ***Copiez le fichier essai311 dans le répertoire Rep22 en l'appelant essai221.bak***

55 ***Copiez le fichier essai312 dans le répertoire Rep22 en l'appelant essai222.bak***

56 ***Copiez le fichier essai312 dans le répertoire Rep32 en l'appelant essai321.new et effacer le fichier essai312***

57 ***Déplacer le fichier essai321.new dans le répertoire Rep21***

58 ***Effacer avec confirmation le fichier essai311***

59 ***Effacer le répertoire Rep22 et tout ce qu'il contient, sans confirmation***

60 ***Si comme il vous à été demandez de le faire vous n'avez pas bougé du répertoire Rep1, remontez dans le répertoire parent(avec la commande cd) et faites une archive de votre arborescence que vous appelez Reps.tar(commande tar)***

61 ***Vérifiez que votre archive contient bien votre arborescence***

62 ***Effacer en une seule commande et avec confirmation, votre arborescence complète(pas celle qui est dans votre archive)***

1.2.3.Comme sous DOS les variables globales

La commande «echo $variable» affiche le contenu de variable

63 ***Le symbole «~» équivaut à un répertoire bien particulier, lequel ?***

64 ***Les symboles «.» et «..» correspondent à des répertoires bien particuliers les quels ?***

Que contiennent les variables suivantes et quelles peuvent être leur utilité ?:

65 ***$PATH***

66 ***$LANG***

67 ***$OLDPWD***

68 ***$HOME***

69 ***$HOSTNAME***

70 ***$HOSTTYPE***

71 ***$USER***

72 ***$SHELL***

Par défaut le répertoire contenant la commande ifconfig n'est pas dans le $PATH d'un utilisateur, modifier la valeur de cette variable, avec la commande «export variable=valeur» pour qu'elle contienne le répertoire en question et ainsi que votre utilisateur puisse exécuter cette commande en tapant simplement ifconfig

La commande cd - permet de retourner dans le dernier répertoire où l'on s'est placé. Pour cela elle consulte la valeur d'une des variables globales ci-dessus afin de savoir dans quel répertoire se rendre . Toujours avec la commande export modifier la valeur d'une des variables du tableau ci-dessus afin que lorsque vous taperez cd - vous vous retrouviez dans le répertoire /var/log .

1.2.4.Les redirections

Il existe différentes méthodes de redirections :

Redirections de sortie : > et >>

La redirection de sortie redirige la sortie standard(stdout: par défaut l'écran) d'une commande dans un fichier, par exemple :

```
$ ls > fichier.liste
```

redirige la sortie de ls(donc la liste des fichiers contenu dans le répertoire courant) dans le fichier fichier.liste

La différence entre > et >> étant que le > écrase le fichier destination alors que le >> écrit à la suite du fichier

Redirection d'entrée : <

La redirection d'entré est «identique» à la redirection de sortie mais comme son nom l'indique redirige l'entrée standard(stdin: par défaut le clavier) depuis un fichier, par exemple la commande write permet d'envoyer un message à un utilisateur connécté à notre machine :

```
$ write demo
Bonjour
```

<ctrl><d>

l'utilisateur demo reçoit :

```
Message from demo2@local_machine on pts/2 at 11:13
...
Bonjour
EOF
```

Maintenant on va envoyer un message contenu dans un fichier ...

```
$ cat message
Bonjour,
ceci est un message qui pourrait être très très
long
mais malheureusement je n'ai pas beaucoup d'idée à
l'instant
et je ne sais pas vraiment quoi mettre dedans
```

... grâce à la redirection <:

```
$ write demo < message
```

Et l'utilisateur demo reçoit bien :

```
Message from demo2@local_machine on pts/2 at 11:21
...
Bonjour,
ceci est un message qui pourrait être très très
long
mais malheureusement je n'ai pas beaucoup d'idée à
l'instant
et je ne sais pas vraiment quoi mettre dedans
EOF
```

La redirection << existe mais elle à une utilité bien différente des trois redirections vues précedemment et son usage est relativement peu fréquent aussi nous ne triterons pas son utilisation dans ce cours.Pour plus d'infos sur cette dernière je vous renvoie à l'«Advanced Bash-Scripting Guide» chapitre 17. [1]

Le pipe |

Le pipe permet d'envoyer la sortie d'une commande sur l'entrée d'une autre commande, par exemple :

```
$ ls | grep toto
```

permet d'afficher tous les fichiers contenus dans le répertoire courant dont le nom contient le motif toto

2.Les processus :

2.1.La commande ps :

«ps -x» permet d'afficher la liste de nos processus
«ps -ax» permet d'afficher la liste des processus de tous les utilisateurs
les paramètres «-u» et «-l» permettent d'obtenir plus de détails sur les dits processus
«ps --help» donne un résumé de tout les affichages possibles de la commande «ps»

2.2.La commande kill :

La commande «kill» sert à envoyer un signal à un processus.
4 signaux couramment utilisé sont STOP, HUP, TERM et KILL.(kill -l donne la liste des signaux et leur numéro identifiant). :
STOP(correspond à la combinaison de touche <ctrl><z>) permet de stopper le processus sans le tuer
HUP(correspond à la combinaison de touche <ctrl><d>)
TERM permet de mettre fin à un processus de manière normal(comme si on avait choisi fermer dans un menu, ou comme si on avait cliquer sur la croix de fermeture d'une fenêtre)
KILL(correspond à la combinaison de touche <ctrl><c>) tue le processus «brutalement».

73	***Lancer quelques processus : par exemple xedit, gcalc, gtksee***
74	***Testez les différents signaux sur ces processus, et ensuite tuez les tous avec le signal TERM***
75	***Lancer gtksee et envoyer lui un signal SIGSTOP(soit avec kill -SIGSTOP pid depuis un autre terminal soit en tapant <ctrl><z> dans le terminal courant) vous remarquez que vous avez stopper le processus gtksee et qu'en plus vous récupérez la main sur le terminal d'où vous aviez lancer gtksee, taper maintenant bg(pour background) dans le terminal dans lequel vous aviez lancé gtksee. Que constatez vous ?***
76	***Renvoyer lui un signal SIGSTOP(depuis un autre terminal) et cette fois taper fg(pour foreground). Que constatez***

vous ?

2.3.La commande jobs :

A tout moment vous pouvez connaître quels sont les processus lancés à partir du terminal en cours avec la commande jobs

3.Amusons nous un peu :

3.1.Netcat :

3.1.1.finir cette partie

demo@hote_distant:$ netcat -l -p 6789 | tar xvfz -
ou
demo@hote_distant:$ netcat -l -p 6789 > sauvegarde.tgz
ou
...
puis
demo2@hote_local:$ tar cvfz - /etc | netcat hote_distant 6789

4.Sources :

Advanced Bash Scripting guide : http://tldp.org/LDP/abs/html/

[1] http://tldp.org/LDP/abs/html/here-docs.html