

ÉCOLE NATIONALE DES FINANCES PUBLIQUES

Formation continue

INFORMATIQUE

Linux INF 715T

Exercices - Corrigés

91564-EC-06

Septembre 2015

AVERTISSEMENT

**LA REPRODUCTION ET L'UTILISATION DE CE DOCUMENT NE SONT
AUTORISÉES QUE POUR LA FORMATION PROFESSIONNELLE ORGANISÉE
PAR LA DIRECTION GENERALE DES FINANCES PUBLIQUES.**

TP 1 : Prise de contact

Lancer le programme terminal (gnome-terminal)
Le prompt indique le nom d'utilisateur, le nom de machine et le répertoire de travail (~ pour répertoire d'accueil).
[stagXX@a00-00 ~]\$ <i>exit</i>
Commande de fermeture de l'interpréteur
\$ <i>uname</i>
Affichage du nom du noyau système (Linux).
\$ <i>uname -a</i>
Affichage de toutes les informations système avec l'option -a.
\$ <i>who</i>
Affichage de la liste des utilisateurs connectés.
\$ <i>whoami</i>
Affichage du nom d'utilisateur courant.
\$ <i>echo Bonjour</i>
La commande <i>echo</i> reçoit un paramètre et l'affiche.
\$ <i>echo Bonjour \n</i>
Ici, deux paramètres sont affichés : Bonjour et c protégé par \.
\$ <i>echo "Bonjour \n"</i>
Affichage d'un seul paramètre : Bonjour \n
\$ <i>echo -e "Bonjour \n"</i>
Avec l'option -e, <i>echo</i> interprète les caractères d'échappement.
\$ <i>cal</i>
Par défaut : affichage du mois en cours.
\$ <i>cal 07 1789</i>
cal reçoit deux paramètres : le mois et l'année.
\$ <i>date</i>
Affichage de la date système.
\$ <i>wc /etc/passwd</i>
La commande <i>wc</i> affiche le nombre de lignes, de mots et de caractères du fichier passé en argument.
\$ <i>wc -l /etc/passwd</i>
Avec l'option -l seul le nombre de lignes est affiché.
\$ <i>wc -lw /etc/passwd</i>
L'option -w permet l'affichage du nombre de mots
\$ <i>daate</i>
La commande <i>daate</i> n'existe pas dans le chemin de recherche des commandes.
\$ <i>DATE</i>
UNIX est sensible à la casse (DATE différent de date).
\$ <i>who -z</i>
L'option -z est passée à la commande <i>who</i> , qui affiche un message d'erreur (option invalide).
\$ <i>wc fic</i>
Le fichier <i>fic</i> n'existe pas, la commande <i>wc</i> indique un message d'erreur.
\$ <i>cal 36 1988</i>
Idem pour le 36 ème mois avec la commande <i>cal</i>
\$ <i>man cal</i>
Appuyez sur « q » pour quitter.
Manuel de cal
\$ <i>man exit</i>
<i>exit</i> est une commande interne du bash.

```
$ man bash  
puis taper « /exit »  
puis « n » et/ou « N »
```

Manuel du bash. Recherche de la chaîne « exit », en avant ou en arrière.

TP 2 : Exploration des fichiers

1. Afficher votre répertoire courant

```
pwd
```

2. Remontez d'un niveau dans l'arborescence

```
cd ..
```

3. Quel est votre nouveau répertoire de travail ?

```
pwd
```

4. Allez dans le répertoire « `/usr/share/man` »

```
cd /usr/share/man
```

5. Lister les fichiers de l'emplacement courant

```
ls
```

6. Allez dans le répertoire « `/usr/share/man/man2` » en utilisant une notation relative

```
cd man2
```

7. Allez dans votre répertoire de connexion

```
cd
```

8. Utilisez l'éditeur less pour ouvrir le fichier « `/etc/hosts` » puis quitter l'application (touche q).

```
less /etc/hosts
```

9. Copiez le fichier « `/etc/hosts` » dans votre répertoire de connexion.

```
cp /etc/hosts .
```

10. Renommez-le pour « `hosts.old` »

```
mv hosts hosts.old
```

11. Créez un lien symbolique vers « `hosts.old` » qui porte le nom « `hosts.lnk` »

```
ln -s hosts.old hosts.lnk
```

12. Copier « `hosts.lnk` » sous le nom de « `hosts.new` ».

```
cp hosts.lnk hosts.new
```

13. Créez un répertoire « `rep1` ».

```
mkdir rep1
```

14. Faites un listage long du répertoire courant

```
ls -l
```

15. Déplacez tous les fichiers commençant par « `hosts` » dans `rep1`

```
mv hosts* repl/
```

16. Allez dans « repl »

```
cd repl
```

17. Éditez avec less le fichier « hosts.lnk »

```
less hosts.lnk
```

18. Supprimez le fichier « hosts.old »

```
rm hosts.old
```

19. Rappelez la commande dans votre historique qui vous a permis d'éditer un fichier.

```
less hosts.lnk
```

20. Faites un listage long du répertoire courant. Que remarquez vous ?

```
ls -l
```

Le lien est cassé

TP 3 : Les droits

Préalable

Placez-vous dans votre répertoire de travail, vérifiez...

Créez un dossier `rep2` dans lequel vous créez un fichier nommé `test`.

Question 1

Positionner les droits `rwxr-x--x` sur le fichier `test` en mode symbolique.

Plusieurs solutions possibles, par exemple :

```
$ chmod u=rwx,g=rx,o=x test
```

```
$ chmod a=x,gu+r,u+w test
```

Question 2

Ôter le droit de lecture au groupe sur le fichier `test` en mode symbolique.

```
$ chmod g-r test
```

Question 3

Ajouter, en mode symbolique, le droit d'écriture à tous les utilisateurs, sur le dossier `rep2` (répertoire courant).

```
$ chmod a+w .
```

Question 4

Regarder les droits donnés par l'`umask` (en symbolique).

```
$ umask -S
u=rwx,g=rx,o=rx
```

Question 5

Regarder les droits associés au fichier `/etc/nsswitch.conf`

```
$ ls -l /etc/nsswitch.conf
-rw-r--r-- 1 stag01 stagiaire 1663 jui 23 2007 /etc/nsswitch.conf
```

Le propriétaire dispose des droits en lecture et écriture, le groupe et les autres utilisateurs n'ont que le droit de lire le fichier.

Question 6

Faire une copie de `/etc/nsswitch.conf` dans votre répertoire de connexion, constater à nouveau les droits.

```
$ cp /etc/nsswitch.conf ~
```

Résultat : Le fichier `nsswitch.conf` est copié dans le répertoire de connexion. Les propriétaires et groupes sont changés et égaux à l'uid et au gid de l'utilisateur qui a lancé la commande. Il y a application du masque aux droits existants du fichier. Résultat `rw-` pour le propriétaire, `r` pour le groupe et `r` pour les autres.

L'option `-p` permet de conserver les permissions et l'horodatage (date de modification, etc.).

Question 7

Modifier les droits du fichier `nsswitch.conf` (dans votre répertoire) de façon à donner tous les droits à tout le monde.

```
$ chmod a=rwx nsswitch.conf
```

Question 8

Copier `nsswitch.conf` en `nsswitch.sav`

Constater et expliquer les changements de droits.

```
$ cp nsswitch.conf nsswitch.sav
```

```
$ ls -l nsswitch.sav
-rwxr-xr-x 1 stag01 stagiaire 545 sept. 7 14:10 nsswitch.sav
```

**Résultat : Les droits sont égaux aux droits du fichier source + l'umask :
`rwX r-X r-X`**

Question 9

Dans votre répertoire de travail, créez deux nouveaux dossiers `rep3` et `rep4` et dans chacun de ces dossiers un fichier `test`.

Supprimer pour tous les utilisateurs le droit en exécution sur `rep3` et le droit en lecture sur `rep4`.

Testez d'exécuter un listage long des contenus de `rep3` et `rep4`.

```
$ cd
```

```
$ mkdir rep3 rep4
```

```
$ touch rep3/test rep4/test
```

```
$ chmod a-x rep3
```

```
$ chmod a-r rep4
```

```
$ ls -l rep3
ls: impossible d'accéder à rep3/test: Permission non accordée
total 0
-????????? ? ? ? ? ? test
```

```
$ ls -l rep4
ls -l rep4
ls: impossible d'ouvrir le répertoire rep4: Permission non accordée
```


Question 10

Exécuter les commandes suivantes afin de créer le fichier script.sh

```
echo " echo 'il est $(date)' " >rep3/script.sh
```

```
echo " echo 'il est $(date)' " >rep4/script.sh
```

Il est impossible de créer un fichier dans rep3

Mais c'est possible dans rep4

Question 11

Déplacer le fichier « script.sh » dans votre répertoire de connexion et déterminer les droits minimaux nécessaires pour pouvoir exécuter ce script. Vous pouvez exécuter le script avec « ./script.sh ».

```
$ mv rep4/script.sh ~
```

Pour exécuter un script, il faut disposer au minimum des droits en lecture et en exécution.

```
$ chmod u+rx script.sh
```

```
$ ./script.sh
```

```
il est lundi 7 septembre 2015, 14:53:02 (UTC+0200)
```


Question 12

Supprimez tout ce que vous avez créé.

```
$ chmod -R a+rwX rep*
```

```
$ rm -rf rep* ~/script.sh
```


TP 4 : Les processus

1. En mode graphique, lancez un terminal.
2. Tuer tous les processus gedit existants : `killall gedit`
3. Lancer l'application « gedit ».
4. L'interpréteur de commande est-il utilisable ?
5. À l'aide des raccourcis clavier, envoyez un signal SIGSTOP au processus gedit.
6. Le processus gedit existe-t-il encore ? Est-il utilisable ?
7. Remettez gedit au premier plan pour qu'il continue son exécution.
8. Envoyer un signal d'arrêt (SIGTERM) à gedit.
9. Lancez « gedit » directement en arrière-plan
10. Dressez l'arborescence de l'éditeur
11. Utiliser killall pour le fermer.
12. Lancer la commande « `sleep 1000` » en arrière-plan. Notez le PID de ce nouveau processus.
13. Utilisez la commande « `echo $$` » pour connaître le numéro de processus de ce shell
-  14. Utilisez un second shell pour envoyer au premier le signal par défaut (SIGTERM). Que se passe-t-il ? Qu'est devenu le processus sleep 1000 ?
15. Dans un nouveau shell, lancer une commande « `nohup sleep 1000` » en arrière-plan. Notez le PID de ce nouveau processus.
16. Dans le second shell, répétez la commande de destruction du 1^{er} shell. Qu'en est-il du processus sleep ? Qui est son père ?

```
$ killall gedit
$ gedit
Interpréteur de commandes utilisable!!
$ ctrl-z
Le processus existe toujours mais n'est plus utilisable!
$ fg
$ ctrl-c
$ jobs
$ gedit &
$ pstree -p PID_gedit
gedit -> bash -> gnome-terminal -> init
$ sleep 1000 &
$ jobs
$ echo $$
$ top puis touche k ou kill -15 PID_du_Bash1
Le processus gedit a été détruit avec la destruction du shell parent
$ nohup sleep 1000 &
$ jobs
$ echo $$
$ gedit -> init
```

Son père est dorénavant l'init de l'utilisateur

TP 5 : Les processus (2)

Le script « pid » disponible dans le répertoire /grp/informatique/linux affiche son numéro de processus et celui de son père. Le lancer sous les quatre formes ci-dessous pour visualiser leur effet, et déterminer les droits nécessaires à l'exécution de ce shell pour chaque forme.

```
$ ./pid
```

Le script est lancé à partir du répertoire de travail. Le "/" est obligatoire car par défaut sous Unix les commandes ne sont pas cherchées dans le répertoire de travail.

```
$ sh pid
```

Un shell est lancé et exécute le script pid

```
$ . pid
```

Le script pid est exécuté dans le shell courant.

```
$ exec ./pid
```

Le processus shell fait un exec (appel système) vers un nouveau shell qui exécute le script pid. À la fin de celui-ci, la session est terminée : le nouveau shell a complètement remplacé l'ancien.

TP 6 : Flux et filtres

Question 1

Afficher la version triée du fichier `/etc/passwd`, suivant le login.

```
sort -t ":" -k 1,1 /etc/passwd
```

La commande `sort` sans argument ne suffit pas (voir utilisateurs avahi)

Question 2

Afficher la ligne de `/etc/passwd` qui contient `root`.

```
grep "root" /etc/passwd
```

Question 3

Dans le flux généré par `ifconfig`, relevez les lignes contenant le mot « `adr` »

```
ifconfig | grep "adr"
```



Question 4

Dans le flux généré par `ps -ef`, relevez la ligne concernant le processus « `/sbin/init` ».

```
$ ps -ef |grep "/sbin/init" |grep -v "grep"
```

Question 5

Affichez uniquement la deuxième ligne de la commande « `ifconfig eth0` »

```
$ ifconfig eth0 | head -2 | tail -1
```



Question 6 curl

Donnez la ligne qui récupère les flux de la commande : `ls -l /etc /VAR` en mettant :

- la sortie d'erreur en écrasement dans le fichier « `erreur.log` »
- la sortie standard en écrasement dans le fichier « `result.log` »

```
ls -l /etc /VAR 2>erreur.log 1>result.log
```

Question 7

Trier le fichier `/etc/passwd` en fonction du numéro d'utilisateur (le troisième champ), le résultat étant envoyé dans un fichier « `passwd_trie` » de votre répertoire.

```
sort -n -t":" -k3,3 /etc/passwd > ~/passwd_trieUID
```

Tri du fichier en mode numérique, le séparateur est « : », le tri s'effectue sur le troisième champ.

Question 8

Le septième champ du fichier `/etc/passwd` contient le shell par défaut. Comptez le nombre de shells utilisés sur le système.

```
cut -d: -f 7 /etc/passwd | sort -u | wc -l
```

Question 9 Pas vraiment nécessaire...

Copier le fichier `/etc/services` sans utiliser la commande `cp`.

```
cat /etc/services > test
cat < /etc/services > test
tee test < /etc/services > /dev/null
```

Question 10 Une petite modification

Pour la commande `"ls -l"`, faites en sorte que la sortie standard s'affiche à l'écran et s'ajoute au fichier `"cumul.log"` situé dans votre répertoire de travail.

```
ls -l | tee -a cumul.log
```


TP 7 : Recherche de fichiers

Question 1

Chercher dans « /home » un fichier d'extension « pdf » de plus de 10Ko.

```
find /home -name "*.pdf" -size +10k 2>/dev/null
```

Question 2

Rechercher un fichier nommé `resolv.conf` dans tout le système de fichier

Solution find censurée !

```
locate resolv.conf
```

Question 3

Rechercher les fichiers nommés *exactement* `resolv.conf` dans le système de fichier. La section « exemple » du manuel en ligne pourrait bien vous être utile.

```
locate -b '\resolv.conf'
```

Question 4

Créez un fichier de plus de 90 jours :

```
touch -t 1412011100 test.sauv
```

Créez une commande avec `find` permettant de supprimer dans votre arborescence tous les fichiers `"*.sauv"` et qui ont plus de 90 jours

```
$ find . -name '*.sauv' -mtime +90 -exec rm -f {} \; 2>/dev/null
```

Conseil du jour : passer une fois `find` « à blanc » pour connaître les fichiers qu'il va trouver.

Question 5

Donner aux autres les droits de traverser les sous-répertoires du répertoire courant

```
find . -type d -exec chmod o+x {} \; 2>/dev/null
```


TP 8 : Archivage

Question 1 Commande tar

- Archivez votre répertoire de travail dans le répertoire temporaire « /tmp »
L'archive obtenu devra être nommée archive_stag0X.tar.gz
- Listez le contenu de l'archive obtenu
- Vérifiez qu'il contient le fichier .bash_history
- Désarchivez dans /tmp/rep_stag0X/ (le créer si non existant)

Solution

```
tar -cvzf /tmp/archive_${USER}.tar.gz /home/stag01
tar -tzvf /tmp/archive_${USER}.tar.gz
tar -tzvf /tmp/archive_${USER}.tar.gz | grep .bash_history
mkdir /tmp/rep_${USER}
tar -xzvf /tmp/archive_${USER}.tar.gz -C /tmp/rep_${USER}
```


TP 9 : Asynchrone

Objectif

Écrire plusieurs lignes de crontab permettant de réaliser les actions suivantes.

1. Effectuez la commande

`echo "ca marche $(date)"`

Toutes les 2 minutes entre 8 h et 12 h, 13 h et 18 h, du lundi au vendredi

Vous devez rediriger la sortie standard dans le fichier `/tmp/test`

2. Sauvegardez votre répertoire de travail tous les soirs à 20 h dans `/tmp/archive.tar`.
3. Mettez en place une commande qui nettoie toutes les 5 minutes votre répertoire d'accueil des fichiers commençant par `toto`

Solution

```
* 8-11,14-17 * * 1-5 echo "ca marche $(date)" >>/tmp/test  
0 20 * * * tar -czf /tmp/archive.tar.gz /home/stag01  
*/5 * * * * rm -f /home/stag01/toto*
```



Nettoyage

Nettoyez la crontab avec la commande

```
crontab -r
```