

TP9 : Les commandes – Manipuler les fichiers et les répertoires

1. Les commandes de gestion de fichiers et de répertoires.

1. Depuis la VM **Deb13serveur**, créez l'utilisateur **guest** (mot de passe **guest**) avec la commande **adduser** (**adduser guest**). Déloguez-vous (commande **logout** ou **exit** et loguez-vous en tant que **guest**).
2. Repérez le caractère ~ (tilde). Dans quel répertoire êtes-vous ?

```
guest@US:~$ pwd
/home/guest
guest@US:~$
```

3. Copiez le fichier **/etc/passwd** dans le répertoire courant et nommez la copie **mot_de_passe** (dans un premier temps, vous faites une erreur dans le nom du fichier).

```
guest@US:~$ cp /etc/password mot_de_passe
cp: cannot stat '/etc/password': No such file or directory
guest@US:~$ cp /etc/passwd mot_de_passe
guest@US:~$ ls -l
total 4
-rw-r--r-- 1 guest guest 1085 nov. 25 18:49 mot_de_passe
guest@US:~$ _
```

4. Copiez les fichiers **/etc/group** et **/etc/profile** dans le répertoire courant en conservant leur nom d'origine. Listez les fichiers en affichant leurs attributs.

```
guest@US:~$ cp /etc/group /etc/profile .
guest@US:~$ ls -l
total 12
-rw-r--r-- 1 guest guest 681 nov. 25 19:32 group
-rw-r--r-- 1 guest guest 1085 nov. 25 18:49 mot_de_passe
-rw-r--r-- 1 guest guest 665 nov. 25 19:32 profile
guest@US:~$
```

5. Listez l'ensemble des fichiers du répertoire y compris les fichiers cachés.

```
guest@US:~$ ls -a
. .bash_history .bashrc group profile
.. .bash_logout .cache mot_de_passe .profile
guest@US:~$ ls -la
total 40
drwxr-xr-x 3 guest guest 4096 nov. 25 19:32 .
drwxr-xr-x 5 root root 4096 nov. 25 13:38 ..
-rw----- 1 guest guest 59 nov. 25 18:22 .bash_his
-rw-r--r-- 1 guest guest 220 nov. 25 13:38 .bash_log
-rw-r--r-- 1 guest guest 3486 nov. 25 13:38 .bashrc
drwx----- 2 guest guest 4096 nov. 25 13:39 .cache
-rw-r--r-- 1 guest guest 681 nov. 25 19:32 group
-rw-r--r-- 1 guest guest 1085 nov. 25 18:49 mot_de_passe
-rw-r--r-- 1 guest guest 665 nov. 25 19:32 profile
-rw-r--r-- 1 guest guest 675 nov. 25 13:38 .profile
guest@US:~$
```

Répertoire courant
(guest) et
répertoire père
(home)

6. Créez un répertoire **un_rep** et listez les fichiers avec leurs attributs.

```
guest@US:~$ mkdir un_rep
guest@US:~$ ls -l
total 16
-rw-r--r-- 1 guest guest 681 nov. 25 19:32 group
-rw-r--r-- 1 guest guest 1085 nov. 25 18:49 mot_de_passe
-rw-r--r-- 1 guest guest 665 nov. 25 19:32 profile
drwxrwxr-x 2 guest guest 4096 nov. 25 19:38 un_rep
guest@US:~$ ls -F
group mot_de_passe profile un_rep/
guest@US:~$ _
```

Remarques :

- L'option **-F** de la commande **ls** ajoute un suffixe aux fichiers. Ce suffixe est un slash dans le cas des répertoires et une étoile dans le cas d'une commande ;
 - Le x associé au répertoire **un_rep** est le **droit d'accès** à ce répertoire et non pas le **droit d'exécution** (commande ou script).
7. Copiez le fichier **group** dans le répertoire courant (**g2** est le nom de la copie), détruisez le fichier **group**, créez un alias (dans la console et non pas dans le fichier **.bashrc**), renommez **g2** en **group**.

```

quest@US:~$ ls
group mot_de_passe profile un_rep
quest@US:~$ cp group g2
quest@US:~$ ls
g2 group mot_de_passe profile un_rep
quest@US:~$ rm -i group
rm: remove regular file `group'? n
quest@US:~$ alias rm='rm -i'
quest@US:~$ rm group
rm: remove regular file `group'? n
quest@US:~$ unalias rm
quest@US:~$ rm group
quest@US:~$ ls
g2 mot_de_passe profile un_rep
quest@US:~$ mv g2 group
quest@US:~$ ls
group mot_de_passe profile un_rep
quest@US:~$ _

```

Remarque : la destruction d'un fichier par la commande **rm** est très dangereuse. Il n'y a pas de possibilité de revenir en arrière. La commande ne demande pas de confirmation sauf si l'on utilise l'option **-i**. Les alias ne sont valables que pour la session. Il faut donc les sauvegarder par exemple dans **~/bash_profile** ou dans **~/bashrc**

8. Déplacez-vous dans un répertoire, visualisez les caractéristiques d'un fichier et revenez dans votre répertoire d'accueil (vous pouvez **remplacer la commande who par une autre : rm par exemple**).

```

→ quest@US:~$ cd /usr/bin
quest@US:/usr/bin$ pwd
/usr/bin
quest@US:/usr/bin$ ls -l who
-rwxr-xr-x 1 root root 47680 nov. 19 2012 who
→ quest@US:/usr/bin$ cd
quest@US:~$

```

9. Visualisez le type des fichiers **/usr/bin/rm** (une commande) et **/etc/issue** (un fichier texte).

```

quest@US:~$ file /usr/bin/who
/usr/bin/who: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically l
inked (uses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=0x7c64b5da41331275
8a158091c8066a1b4e0f488f, stripped
quest@US:~$ file /etc/issue
/etc/issue: ASCII text
quest@US:~$ _

```

10. Visualisez le contenu d'un fichier. La commande **cat** adaptée aux petits fichiers permet de visualiser tout le contenu dans la console d'un coup.

```

guest@US:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255

guest@US:~$ _

```

11. Visualisez le contenu du fichier `/etc/services` page par page : `less /etc/services` (ou `more`).

```

# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux      1/tcp                                # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp      sink null
discard     9/udp      sink null
sysstat     11/tcp     users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
gotd        17/tcp     quote
msp         18/tcp     # message send protocol
msp         18/udp
chargen     19/tcp     ttytst source
chargen     19/udp     ttytst source
ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp     fspd

/etc/services

```

On passe à l'écran suivant avec la touche **espace**. La **touche q** met fin à la commande `less`.

12. Visualisez le même fichier page par page avec la commande `more`.

```
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux      1/tcp                               # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp          sink null
discard     9/udp          sink null
sysstat     11/tcp         users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
qotd        17/tcp         quote
msp         18/tcp         # message send protocol
msp         18/udp
chargen     19/tcp         ttytst source
chargen     19/udp         ttytst source
ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp         fspd

--More-- (5%)
```

2. Les utilitaires.

1. Affichez les 10 premières lignes d'un fichier.

```
guest@US:~$ head /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
guest@US:~$
```

2. Affichez les 10 dernières lignes d'un fichier.

```
guest@US:~$ tail /etc/services
uboxd      20012/udp
binkp      24554/tcp          # binkp fidonet protocol
asp        27374/tcp          # Address Search Protocol
asp        27374/udp
csync2     30865/tcp          # cluster synchronization tool
dircproxy  57000/tcp          # Detachable IRC Proxy
tfido      60177/tcp          # fidonet EMSI over telnet
fido       60179/tcp          # fidonet EMSI over TCP

# Local services
guest@US:~$ _
```

3. Afficher les lignes d'un fichier qui contiennent la chaîne « HTTP ».

```
guest@US:~$ grep HTTP /etc/services
http       80/tcp          www          # WorldWideWeb HTTP
hkp        11371/tcp        # OpenPGP HTTP Keyserver
guest@US:~$
```

- ☞ Décommentez au préalable la ligne `alias grep='grep - -color=auto'` dans le fichier `.bashrc` puis déloguez-vous (commande `exit` ou `logout`) et reloguez-vous.

```
alias grep='grep --color=auto'
#alias fgrep='fgrep --color=auto'
```

4. Idem mais on ne tient pas compte de la casse.

```
guest@US:~$ grep -i HTTP /etc/services
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
http      80/tcp      www          # WorldWideWeb HTTP
http      80/udp      # HyperText Transfer Protocol
https     443/tcp      # http protocol over TLS/SSL
https     443/udp
http-alt  8080/tcp     webcache     # WWW caching service
http-alt  8080/udp
hkp       11371/tcp    # OpenPGP HTTP Keyserver
guest@US:~$
```

Par défaut, **grep** tient compte de la casse. Utilisez l'option **-i** pour renvoyer toutes les lignes contenant un mot donné sans tenir compte des majuscules ou des minuscules.

5. Affichez les lignes d'un fichier qui commencent par « http ».

```
guest@US:~$ grep '^http' /etc/services
http      80/tcp      www          # WorldWideWeb HTTP
http      80/udp      # HyperText Transfer Protocol
https     443/tcp      # http protocol over TLS/SSL
https     443/udp
http-alt  8080/tcp     webcache     # WWW caching service
http-alt  8080/udp
guest@US:~$
```

6. Affichez les 3 dernières lignes d'un fichier trié.

```
guest@US:~$ sort /etc/services | tail -3
zope-ftp  8021/tcp    # zope management by ftp
zserv    346/tcp     # Zebra server
zserv    346/udp
guest@US:~$
```

7. Affichez triées, page par page, les lignes d'un fichier qui ne commencent pas par « # ».

```
guest@US:~$ grep -v '^#' /etc/services | sort | more_
```

Le paramètre **-v** permet de connaître toutes les lignes qui ne contiennent pas un mot donné.

```
acr-nema  104/tcp     dicom        # Digital Imag. & Comm. 300
acr-nema  104/udp     dicom
afbackup  2988/tcp    # Afbbackup system
afbackup  2988/udp
afmbakup  2989/tcp    # Afmbakup system
afmbakup  2989/udp
afpovertcp 548/tcp     # AFP over TCP
afpovertcp 548/udp
afs3-bos  7007/tcp    # basic overseer process
afs3-bos  7007/udp
afs3-callback 7001/tcp    # callbacks to cache managers
afs3-callback 7001/udp
afs3-errors 7006/tcp    # error interpretation service
afs3-errors 7006/udp
afs3-fileserver 7000/tcp    bbs          # file server itself
afs3-fileserver 7000/udp    bbs
afs3-kaserver 7004/tcp    # AFS/Kerberos authentication
afs3-kaserver 7004/udp
afs3-prserver 7002/tcp    # users & groups database
afs3-prserver 7002/udp
afs3-rmtsys  7009/tcp    # remote cache manager service
afs3-rmtsys  7009/udp
afs3-update  7008/tcp    # server-to-server updater
--More--
```

8. Affichez le nombre de lignes, de mots et de caractères du fichier **/etc/services**.

```

guest@US:~$ wc /etc/services
605 2627 19281 /etc/services
guest@US:~$

```

9. Affichez uniquement le nombre de lignes d'un fichier.

```

guest@US:~$ wc -l /etc/services
605 /etc/services
guest@US:~$

```

10. Loguez-vous en tant que **root** (commande **exit** ou **logout** au préalable) et créez un fichier **etudiants.txt** à l'aide de l'éditeur Nano :

```

GNU nano 2.2.6      File: etudiants.txt

Nicolas
Denis
Souphiane
Azer
Fabien
Cedric
Tiphaine
Xavier
Antoine
David

```

11. Triez les prénoms des étudiants dans l'ordre alphabétique :

```

root@US:~# sort etudiants.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
root@US:~#

```

12. Triez les prénoms dans l'ordre inverse :

```

root@US:~# sort -r etudiants.txt
Xavier
Tiphaine
Souphiane
Nicolas
Fabien
Denis
David
Cedric
Azer
Antoine
root@US:~#

```

13. Enregistrez le résultat du tri dans un fichier nommé **prenoms_tries** :

```

root@US:~# sort -o prenoms_tries etudiants.txt
root@US:~# ls -l
total 1032
-rw-r--r-- 1 root root 724696 nov. 10 2013 arborescence.txt
-rw-r--r-- 1 root root    73 oct. 26 19:42 etudiants.txt
-rw-r--r-- 1 root root    73 oct. 26 20:31 prenoms_tries
-rw-r--r-- 1 root root 321071 nov. 26 2013 readme
root@US:~# cat prenoms_tries
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
root@US:~#

```

14. Conservez les 3 premiers caractères de chaque ligne du fichier **etudiants.txt** :

```
root@US:~# cut -c -3 etudiants.txt
Nic
Den
Sou
Aze
Fab
Ced
Tip
Xav
Ant
Dav
root@US:~#
```

15. Conservez les caractères 2 à 5 de chaque ligne :

```
root@US:~# cut -c 2-5 etudiants.txt
icol
enis
ouph
zer
abie
edri
ipha
avie
ntoi
avid
root@US:~#
```

16. Conservez les caractères de chaque ligne à partir du troisième :

```
root@US:~# cut -c 3- etudiants.txt
colas
nis
uphiane
er
bien
dric
phaine
vier
toine
vid
root@US:~#
```

17. Ouvrez à l'aide de l'éditeur Nano, le fichier **prenoms_tries**, complétez-le en séparant les valeurs par un délimiteur (une virgule en l'espèce) dans le but de l'enregistrer (**Ctrl+o**) au format CSV sous le nom **notes.csv** :

```
GNU nano 2.2.6      File: notes.csv
Antoine, SISR, 13, AB
Azer, SISR, 13, AB
Cedric, SISR, 13, AB
David, SISR, 14, AB
Denis, SLAM, 17, TB
Fabien, SISR, 13, AB
Nicolas, SISR, 17, TB
Souphiane, SLAM, 16, TB
Tiphaine, SLAM, 15, B
Xavier, SLAM, 11, M
```

18. Affichez la colonne 1 :

```
root@US:~# cut -d , -f 1 notes.csv
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
root@US:~#
```

19. Conservez les trois premières colonnes :

```
root@US:~# cut -d , -f -3 notes.csv
Antoine, SISR, 13
Azer, SISR, 13
Cedric, SISR, 13
David, SISR, 14
Denis, SLAM, 17
Fabien, SISR, 13
Nicolas, SISR, 17
Souphiane, SLAM, 16
Tiphaine, SLAM, 15
Xavier, SLAM, 11
root@US:~#
```

20. Affichez les valeurs des champs 1 et 3 :

```
root@US:~# cut -d , -f 1,3 notes.csv
Antoine, 13
Azer, 13
Cedric, 13
David, 14
Denis, 17
Fabien, 13
Nicolas, 17
Souphiane, 16
Tiphaine, 15
Xavier, 11
root@US:~#
```