

TP9 : Les commandes – Manipuler les fichiers et les répertoires

Sommaire

1 - Les commandes de gestion de fichiers et de répertoires.....2
2 - Les utilitaires.....6

1 - Les commandes de gestion de fichiers et de répertoires

→ On **ajoute** un nouvel utilisateur **guest** avec la commande **adduser** :

```
root@DEB12Server: ~#adduser guest
Ajout de l'utilisateur « guest » ...
Ajout du nouveau groupe « guest » (1001) ...
Ajout du nouvel utilisateur « guest » (1001) avec le groupe « guest » (1001) ...
adduser : Le répertoire personnel « /home/guest » existe déjà. Pas de modification de ce répertoire.
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
Modifier les informations associées à un utilisateur pour guest
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
  NOM []:
  Numéro de chambre []:
  Téléphone professionnel []:
  Téléphone personnel []:
  Autre []:
Cette information est-elle correcte ? [O/n]
Ajout du nouvel utilisateur « guest » aux groupes supplémentaires « users » ...
Ajout de l'utilisateur « guest » au groupe « users » ...
```

→ On se déconnecte de **root** avec **exit** et on se **connecte** à l'utilisateur **guest** :

```
DEB12Server login: guest
Password:
```

```
guest@DEB12Server: ~$
```

* On repère grâce au tilde qu'on se trouve dans le répertoire personnel.

```
guest@DEB12Server: ~$ pwd
/home/guest
```

→ On **copie** le fichier **/etc/passwd** dans ce répertoire et on appelle la copie **mot_de_passe** :

```
guest@DEB12Server: ~$ cp /etc/password mot_de_passe
cp: impossible d'évaluer '/etc/password': Aucun fichier ou dossier de ce type
guest@DEB12Server: ~$ cp /etc/passwd mot_de_passe
guest@DEB12Server: ~$ ls -l
total 4
-rw-r--r-- 1 guest guest 1222 19 déc. 20:49 mot_de_passe
```

3

→ On **copie** aussi les fichiers **/etc/group** et **/etc/profile** dans ce répertoire en laissant leur nom d'origine puis on **liste** les fichiers avec leur attributs :

```
guest@DEB12Server:~$ cp /etc/group /etc/profile .
guest@DEB12Server:~$ ls -l
total 12
-rw-r--r-- 1 guest guest 709 19 déc. 20:55 group
-rw-r--r-- 1 guest guest 1222 19 déc. 20:49 mot_de_passe
-rw-r--r-- 1 guest guest 769 19 déc. 20:55 profile
```

→ On **liste** tous les fichiers y compris les fichiers **cachés** :

```
guest@DEB12Server:~$ ls -a
.  ..  .bash_logout  .bashrc  group  mot_de_passe  .profile  profile
guest@DEB12Server:~$ ls -la
total 32
drwx----- 2 guest guest 4096 19 déc. 20:55 .
drwxr-xr-x 4 root root 4096 19 déc. 20:31 ..
-rw-r--r-- 1 guest guest 220 19 déc. 20:31 .bash_logout
-rw-r--r-- 1 guest guest 3526 19 déc. 20:31 .bashrc
-rw-r--r-- 1 guest guest 709 19 déc. 20:55 group
-rw-r--r-- 1 guest guest 1222 19 déc. 20:49 mot_de_passe
-rw-r--r-- 1 guest guest 807 19 déc. 20:31 .profile
-rw-r--r-- 1 guest guest 769 19 déc. 20:55 profile
```

→ On crée un répertoire **un_rep** et on liste les fichiers avec leur attributs :

```
guest@DEB12Server:~$ mkdir un_rep
guest@DEB12Server:~$ ls -l
total 16
-rw-r--r-- 1 guest guest 709 19 déc. 20:55 group
-rw-r--r-- 1 guest guest 1222 19 déc. 20:49 mot_de_passe
-rw-r--r-- 1 guest guest 769 19 déc. 20:55 profile
drwxr-xr-x 2 guest guest 4096 19 déc. 21:19 un_rep
```

→ On **copie** le fichier **group** dans le répertoire actuel en le renommant **g2**.

```
guest@DEB12Server:~$ ls
group  mot_de_passe  profile  un_rep
guest@DEB12Server:~$ cp group g2
guest@DEB12Server:~$ ls
g2  group  mot_de_passe  profile  un_rep
```

→ On crée un **alias** sur la commande **rm** avec le paramètre **-i** pour demander la confirmation de la suppression avec **alias rm='rm -i'** puis on supprime le fichier original **group** :

```
guest@DEB12Server:~$ alias rm='rm -i'
guest@DEB12Server:~$ rm group
rm : supprimer 'group' du type fichier ? n
```

4

→ On renomme **g2** en **group** :

```
guest@DEB12Server:~$ unalias rm
guest@DEB12Server:~$ rm group
guest@DEB12Server:~$ ls
g2  mot_de_passe  profile  un_rep
guest@DEB12Server:~$ mv g2 group
guest@DEB12Server:~$ ls
group  mot_de_passe  profile  un_rep
```

→ On se déplace dans un répertoire avec **cd**, on visualise les caractéristiques d'un fichier et on retourne au répertoire d'accueil :

```
guest@DEB12Server:~$ cd /usr/bin
guest@DEB12Server:/usr/bin$ pwd
/usr/bin
guest@DEB12Server:/usr/bin$ ls -l who
-rwxr-xr-x 1 root root 60432 20 sept. 2022 who
guest@DEB12Server:/usr/bin$ cd
guest@DEB12Server:~$
```

→ On visualise le type des fichiers **/usr/bin/rm** et **/etc/issue** avec **file** :

```
guest@DEB12Server:~$ file /usr/bin/rm
/usr/bin/rm: ELF 64-bit LSB pie executable, x86-64, version
52030f20cd66139fba719927b, for GNU/Linux 3.2.0, stripped
guest@DEB12Server:~$ file /etc/issue
/etc/issue: ASCII text
```

→ On visualise le contenu d'un fichier avec la commande **cat** :

```
guest@DEB12Server:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on yo
ur system
# and how to activate them. For more information, see interf
aces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
auto enp0s3
iface enp0s3 inet dhcp
```

→ On visualise le contenu du fichier `/etc/services` page par page avec la commande **less** :

```
# Network services, Internet style
#
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml .
#
# 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 multiplex
tplexer
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
chargen        19/tcp          ttytst source
chargen        19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
```

→ On visualise le même fichier avec la commande **more** :

```
# Network services, Internet style
#
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml .
#
# 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 multiplex
tplexer
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
chargen        19/tcp          ttytst source
chargen        19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
```

2 - Les utilitaires

→ On affiche les **10 premières lignes** d'un fichier avec **head** :

```
guest@DEB12Server:~$ head /etc/services
# Network services, Internet style
#
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml .
#
# 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
```

→ On affiche les **10 dernières lignes** d'un fichier avec **tail** :

```
guest@DEB12Server:~$ tail /etc/services
sgi-cad         17004/tcp       # Cluster Admin daemon
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
```

→ On affiche les lignes d'un fichier qui contiennent la chaîne **HTTP** :

```
guest@DEB12Server:~$ grep HTTP /etc/services
http           80/tcp          www             # WorldWideWeb HTTP
https         443/udp         # HTTP/3
hkp           11371/tcp       # OpenPGP HTTP Keyserver
```

→ On décommente la ligne **alias grep='grep --color=auto'** dans le fichier **.bashrc** avec **nano** :

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

7

→ On affiche les lignes d'un fichier qui contiennent la chaîne **HTTP** avec **grep** :

```
guest@DEB12Server:~$ grep HTTP /etc/services
http      80/tcp      www         # WorldWideWeb HTTP
https     443/udp     # HTTP/3
hkp       11371/tcp   # OpenPGP HTTP Keyserver
```

→ On fait la même chose sans tenir compte des majuscules ou minuscules avec le paramètre **-i** :

```
guest@DEB12Server:~$ grep -i HTTP /etc/services
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml .
http      80/tcp      www         # WorldWideWeb HTTP
https     443/tcp     # http protocol over TLS/SSL
https     443/udp     # HTTP/3
http-alt  8080/tcp    webcache    # WWW caching service
hkp       11371/tcp   # OpenPGP HTTP Keyserver
```

→ On affiche les lignes d'un fichier qui commencent par **http** :

```
guest@DEB12Server:~$ grep '^http' /etc/services
http      80/tcp      www         # WorldWideWeb HTTP
https     443/tcp     # http protocol over TLS/SSL
https     443/udp     # HTTP/3
http-alt  8080/tcp    webcache    # WWW caching service
```

→ On affiche les **3 dernières lignes** d'un fichier trié avec **sort** pour trier :

```
guest@DEB12Server:~$ sort /etc/services | tail -3
zope      9673/tcp    # zope server
zope-ftp  8021/tcp    # zope management by ftp
zserv     346/tcp     # Zebra server
```

→ On affiche les lignes d'un fichier qui ne commencent pas par "#" page par page :

```
guest@DEB12Server:~$ grep -v '^#' /etc/services | sort | more_
acr-nema      104/tcp      dicom        # Digital Imag. & Comm. 300
afpovertcp   548/tcp      # AFP over TCP
afs3-bos     7007/udp     # basic overseer process
afs3-callback 7001/udp     # callbacks to cache managers
afs3-fileserver 7000/udp
afs3-kaserver 7004/udp     # AFS/Kerberos authentication
afs3-prserver 7002/udp     # users & groups database
afs3-rmtsys  7009/udp     # remote cache manager service
afs3-update  7008/udp     # server-to-server updater
afs3-vlserver 7003/udp     # volume location database
afs3-volser  7005/udp     # volume managment server
amanda       10080/tcp    # amanda backup services
amandaidx    10082/tcp    # amanda backup services
amidxtape    10083/tcp    # amanda backup services
```

→ On affiche le nombre de lignes, de mot et de caractères dans le fichier **/etc/services** :

```
guest@DEB12Server:~$ wc /etc/services
 361  1773 12813 /etc/services
```

→ On affiche uniquement le nombre de lignes du fichier :

```
guest@DEB12Server:~$ wc -l /etc/services
361 /etc/services
```

→ On se log en tant que **root** et on créé un fichier **etudiants.txt** avec **nano** :

```
GNU nano 7.2 etudiants.txt *
Nicolas
Denis
Souphiane
Azer
Fabien
Cedric
Tiphaine
Xavier
Antoine
David
```

→ On **trie** le nom des étudiants par ordre alphabétique avec **sort** :

```
root@DEB12Server: ~#sort etudiants.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

→ On les trie dans l'ordre inverse avec le paramètre **-r** :

```
root@DEB12Server: ~#sort -r etudiants.txt
Xavier
Tiphaine
Souphiane
Nicolas
Fabien
Denis
David
Cedric
Azer
Antoine
```

→ On trie le fichier étudiants et on **enregistre** le résultat dans un fichier **prenoms_tries** avec le paramètre **-o** :

```
root@DEB12Server: ~#sort -o prenoms_tries etudiants.txt
root@DEB12Server: ~#ls -l
total 8
-rw-r--r-- 1 root root 73 20 déc. 00:52 etudiants.txt
-rw-r--r-- 1 root root 73 20 déc. 00:57 prenoms_tries
root@DEB12Server: ~#cat prenoms_tries
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

→ On garde uniquement les **3 premiers caractères** des lignes du fichier **etudiant** :

```
root@DEB12Server: ~#cut -c -3 etudiants.txt
Nic
Den
Sou
Aze
Fab
Ced
Tip
Xav
Ant
Dav
```

→ On garde les caractères **2 à 5** de chaque ligne :

```
root@DEB12Server: ~#cut -c 2-5 etudiants.txt
icol
enis
ouph
zer
abie
edri
ipha
avie
ntoi
avid
```

→ On retire les **3 premiers caractères** de chaque ligne :

```
root@DEB12Server: ~#cut -c 3- etudiants.txt
colas
nis
uphiane
er
bien
dric
phaine
vier
toine
vid
```

11

→ On ouvre le fichier **prenoms_tries** avec **nano**, on ajoute des valeurs en les séparant par des virgules puis on l'enregistre sous **notes.csv** :

```
GNU nano 7.2 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
```

→ On affiche la **première colonne** du fichier avec **cut** :

```
root@DEB12Server: ~#cut -d , -f 1 notes.csv
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

→ On affiche les **3 premières colonnes** du fichier :

```
root@DEB12Server: ~#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
```

12

→ On affiche les **1ère** et **3ème** colonnes :

```
root@DEB12Server: ~#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
```