

## TP6 : Les commandes – Manipuler les fichiers et les répertoires.

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

```
guest@DS1:~$ pwd  
/home/guest
```

**1ère étape :** La commande « pwd » nous indique dans quel répertoire nous sommes, ici c'est **guest**.

```
guest@DS1:~$ cp /etc/password mot_de_passe  
cp: impossible d'évaluer '/etc/password': Aucun fichier ou dossier de ce type  
guest@DS1:~$ cp /etc/passwd mot_de_passe  
guest@DS1:~$ ls -l  
total 8  
drwxr-xr-x 2 guest guest 4096 30 nov. 17:12 liens  
-rw-r--r-- 1 guest guest 1474 30 nov. 17:18 mot_de_passe
```

**2ème étape :** Nous copions le fichier **/etc/passwd** dans le répertoire courant et le nommant **mot\_de\_passe** avec la commande « cp /etc/passwd **mot\_de\_passe** », puis nous listons la liste des fichiers en affichant leurs attributs.

```
guest@DS1:~$ cp /etc/group /etc/profile .  
guest@DS1:~$ ls -l  
total 16  
-rw-r--r-- 1 guest guest 720 30 nov. 17:19 group  
drwxr-xr-x 2 guest guest 4096 30 nov. 17:12 liens  
-rw-r--r-- 1 guest guest 1474 30 nov. 17:18 mot_de_passe  
-rw-r--r-- 1 guest guest 769 30 nov. 17:19 profile
```

**3ème étape :** Nous copions également les fichiers **/etc/group** et **/etc/profile** dans le répertoire courant avec « cp /etc/group /etc/profile » en conservant cette fois-ci leur nom d'origine, puis nous listons une nouvelle fois les fichiers en affichant leurs attributs.

```
guest@DS1:~$ ls -a
. ... .bash_history .bash_logout .bashrc group liens mot_de_passe .profile profile
guest@DS1:~$ ls -la
total 40
drwxr-xr-x 3 guest guest 4096 30 nov. 17:19 .
drwxr-xr-x 5 root root 4096 30 nov. 16:57 ..
-rw----- 1 guest guest 247 30 nov. 17:16 .bash_history
-rw-r--r-- 1 guest guest 220 30 nov. 16:57 .bash_logout
-rw-r--r-- 1 guest guest 3526 30 nov. 16:57 .bashrc
-rw-r--r-- 1 guest guest 720 30 nov. 17:19 group
drwxr-xr-x 2 guest guest 4096 30 nov. 17:12 liens
-rw-r--r-- 1 guest guest 1474 30 nov. 17:18 mot_de_passe
-rw-r--r-- 1 guest guest 807 30 nov. 16:57 .profile
-rw-r--r-- 1 guest guest 769 30 nov. 17:19 profile
```

**4ème étape :** Nous listons l'ensemble des fichiers du répertoire ainsi que les fichiers cachés avec la commande « ls -a ».

```
guest@DS1:~$ mkdir un_rep
guest@DS1:~$ ls -l
total 20
-rw-r--r-- 1 guest guest 720 30 nov. 17:19 group
drwxr-xr-x 2 guest guest 4096 30 nov. 17:12 liens
-rw-r--r-- 1 guest guest 1474 30 nov. 17:18 mot_de_passe
-rw-r--r-- 1 guest guest 769 30 nov. 17:19 profile
drwxr-xr-x 2 guest guest 4096 30 nov. 17:22 un_rep
guest@DS1:~$ ls -F
group liens/ mot_de_passe profile un_rep/
```

**5ème étape :** Création d'un répertoire nommé **un\_rep** avec la commande « mkdir un rep », puis nous listons les fichiers et leurs attributs « ls -l ».

```
guest@DS1:~$ ls
group liens mot_de_passe profile un_rep
guest@DS1:~$ cp group g2
guest@DS1:~$ ls
g2 group liens mot_de_passe profile un_rep
guest@DS1:~$ rm -i group
rm : supprimer 'group' ou type fichier ? n
guest@DS1:~$ alias rm='rm -i'
guest@DS1:~$ rm group
rm : supprimer 'group' du type fichier ? n
guest@DS1:~$ unalias rm
guest@DS1:~$ rm g2
guest@DS1:~$ ls
g2 liens mot_de_passe profile un_rep
guest@DS1:~$ mv g2 group
guest@DS1:~$ ls
group liens mot_de_passe profile un_rep
```

**6ème étape :** Nous copions le fichier group en g2, puis nous détruisons le fichier group afin de créer un alias, et nous renommmons g2 en group. Tout cela avec les commandes ci-dessus.

```
guest@DS1:~$ cd /usr/bin  
guest@DS1:/usr/bin$ pwd  
/usr/bin  
guest@DS1:/usr/bin$ ls -l who  
-rwxr-xr-x 1 root root 60384 24 sept. 2020 who  
guest@DS1:/usr/bin$ cd
```

**7ème étape :** Nous nous déplaçons dans le répertoire et visualisons les spécificités d'un fichier.

```
guest@DS1:~$ file /usr/bin/who  
/usr/bin/who: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=d590e93c3a4b5013c2d265b98d27179302829405, for GNU/Linux 3.2.0, stripped  
guest@DS1:~$ file /etc/issue  
/etc/issue: ASCII text
```

**8ème étape :** Visualisations du type des fichiers **/usr/bin/who** (une commande) et **/etc/issue** (un fichier texte), avec la commande « file /usr/bin/who » et « file /etc/issue ».

```
guest@DS1:~$ 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).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
allow-hotplug enp0s3  
iface enp0s3 inet dhcp
```

**9ème étape :** Nous visualisons tout le contenu du fichier dans la console avec la commande « cat /etc/network/interfaces ».

```
# 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
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
chargen    19/tcp          ttyst source
chargen    19/udp          ttyst source
ftp-data   20/tcp
ftp         21/tcp
fsp         21/udp          fspd
ssh         22/tcp          # SSH Remote Login Protocol
telnet     23/tcp
smtp        25/tcp          mail
time        37/tcp          timserver
time        37/udp          timserver
whois       43/tcp          nickname
tacacs      49/tcp          # Login Host Protocol (TACACS)
tacacs      49/udp
domain     53/tcp          # Domain Name Server
domain     53/udp
bootps     67/udp
bootpc     68/udp

/etc/services
```

**10ème étape :** Visualisations 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-number
s.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
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
chargen    19/tcp          ttyst source
chargen    19/udp          ttyst source
ftp-data   20/tcp
ftp        21/tcp
fsp        21/udp          fspd
ssh        22/tcp          # SSH Remote Login Protocol
telnet     23/tcp
smtp       25/tcp          mail
time        37/tcp          timserver
time        37/udp          timserver
whois      43/tcp          nickname
tacacs     49/tcp          # Login Host Protocol (TACACS)
tacacs     49/udp
domain    53/tcp          # Domain Name Server
domain    53/udp
bootps    67/udp
bootpc    68/udp
--Plus--(7%)L
```

**11ème étape :** Même application mais cette fois-ci avec la commande « **more** ».

## 2. Les utilitaires.

```
guest@DS1:~$ head /etc/services
# Network services, Internet style
#
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-number
s.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
guest@DS1:~$ _
```

**1ère étape :** Avec la commande « head /etc/services », nous affichons les 10 premières lignes du fichier.

```
guest@DS1:~$ 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
psync2       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
```

**2ème étape :** Cette fois-ci nous affichons les 10 dernières lignes du fichier avec la commande « tail /etc/services ».

```
guest@DS1:~$ grep HTTP /etc/services
http        80/tcp          www          # WorldWideWeb HTTP
https      443/udp          # HTTP/3
hkp        11371/tcp        # OpenPGP HTTP Keyserver
guest@DS1:~$ _
```

**3ème étape :** Ici nous affichons les lignes du fichier contenant la chaîne HTTP, grâce à la commande « grep HTTP /etc/services ».

```
guest@DS1:~$ grep -i HTTP /etc/services
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-number
s.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
hkps     11371/tcp     # OpenPGP HTTP Keyserver
```

**4ème étape :** Même manipulation mais sans tenir compte de la classe, donc cette fois la commande sera « grep -i HTTP /etc/services ».

```
guest@DS1:~$ 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
guest@DS1:~$
```

**5ème étape :** Nous affichons les lignes du fichier qui commence par http, la commande est « grep ‘^http’ /etc/services ».

```
guest@DS1:~$ sort /etc/services | tail -3
zope      9673/tcp      # zope server
zope-ftp  8021/tcp      # zope management by ftp
zserv     346/tcp       # Zebra server
```

**6ème étape :** La commande « sort /etc/services | tail -3 » permet d'afficher les trois dernières lignes du fichier trier.

```

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-rmtdsys   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
amqp          5672/sctp
amqp          5672/tcp      # AMQP protocol over TLS/SSL
amqps         5671/tcp      # ASF Remote Management and Control Protocol
asp           623/udp      # Address Search Protocol
asp           27374/tcp
asp           27374/udp
auth          113/tcp      authentication tap ident
babel         6696/udp    # Babel Routing Protocol
bacula-dir    9101/tcp    # Bacula Director
bacula-fd     9102/tcp    # Bacula File Daemon
bacula-sd     9103/tcp    # Bacula Storage Daemon
bbs           7000/tcp
bgp            179/tcp      # Border Gateway Protocol
bgpd          2605/tcp    # bgpd vty (zebra)
biff          512/udp      comsat
binkp         24554/tcp    # binkp fidonet protocol
--Plus--
```

**7ème étape :** Avec la commande « grep -v ‘^#’ /etc/services | sort | more » permet d'afficher les lignes du fichier qui ne commencent pas par #, triées page par page.

```
guest@DS1:~$ wc /etc/services
361 1773 12813 /etc/services
```

**8ème étape :** La commande « wc /etc/services » affiche le nombre de lignes, de mots et de caractères du fichier.

```
guest@DS1:~$ wc -l /etc/services  
361 /etc/services
```

**9ème étape :** Ici nous affichons seulement le nombre de lignes d'un fichier avec la commande « **wc -l /etc/services** ».

```
GNU nano 5.4                                     etudiants.txt  
Nicolas  
Denis  
Souphiane  
Azer  
Fabien  
Cedric  
Tiphaine  
Xavier  
Antoine  
David
```

**10ème étape :** Après s'être loguer en tant que root, nous avons donc créer un fichier nommé **etudiants.txt** via l'éditeur Nano.

```
root@DS1 ~# sort etudiants.txt  
Antoine  
Azer  
Cedric  
David  
Denis  
Fabien  
Nicolas  
Souphiane  
Tiphaine  
Xavier
```

**11ème étape :** Avec la commande « **sort etudiants.txt** » nous trions dans l'ordre alphabétique les prénoms des étudiants.

```
root@DS1 ~#sort -r etudiants.txt
Xavier
Tiphaine
Souphiane
Nicolas
Fabien
Denis
David
Cedric
Azer
Antoine
```

**12ème étape :** Cette fois-ci nous trions les prénoms dans l'ordre inverse avec « sort -r etudiants.txt ».

```
root@DS1 ~#sort -o prenoms_tries etudiants.txt
root@DS1 ~#ls -l
total 12
-rw-r--r-- 1 root root    73  7 déc. 16:40 etudiants.txt
drwxr-xr-x 2 root root 4096 30 nov. 16:59 liens
-rw-r--r-- 1 root root    73  7 déc. 16:42 prenoms_tries
```

**13ème étape :** Nous enregistrons ensuite le résultat du tri dans un fichier nommé **prenoms\_tries**, ceci avec « sort -o prenoms\_tries etudiants.txt ».

```
root@DS1 ~#cut -c -3 etudiants.txt
Nic
Den
Sou
Aze
Fab
Ced
Tip
Xav
Ant
Dav
```

**14ème étape :** Nous conservons les trois premiers caractères de chaque prénoms du fichier **etudiants.txt** avec la commande « cut -c -3 etudiants.txt ».

```
root@DS1 ~#cut -c 2-5 etudiants.txt
icol
enis
uph
zer
abie
edri
ipha
avie
ntoi
avid
```

**15ème étape:** Nous conservons les caractères 2 à 5 de chaque ligne, en effectuant « **cut -c 2-5 etudiants.txt** ».

```
root@DS1 ~#cut -c 3- etudiants.txt
colas
nis
uphiane
er
bien
dric
phaine
vier
toine
vid
```

**16ème étape:** Nous conservons les caractères de chaque ligne à partir du troisième, en saisissant la commande « **cut -c 3- etudiants.txt** ».

```
GNU nano 5.4                               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
```

**17ème étape :** Via l'éditeur Nano nous complétons le fichier **prenoms\_tries** puis en l'enregistrant sous le nom **notes.csv**.

```
root@DS1 ~#cut -d , -f 1 notes.csv
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

**18ème étape:** Nous affichons la colonne 1 avec la commande « **cut -d , -f 1 notes.csv** ».

```
root@DS1 ~#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
```

**19ème étape :** Nous voulons conserver les trois premières colonnes donc nous effectuons la commande « **cut -d , -f -3 notes.csv** ».

```
root@DS1 ~#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
```

**20ème étape :** Nous affichons les valeurs des champs 1 et 3 en saisissant la commande « **cut -d , -f 1,3 notes.csv** ».