

# Conte-Rendu TP6 : Les commandes – Manipuler les fichiers et les répertoires

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

Tout d'abord, on se connecte en guest. Puis on demande, grâce à la commande **pwd** dans quel répertoire nous nous situons. On se situe donc, dans le répertoire **/home/guest**.

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

On copie le fichier **/etc/passwd** dans le répertoire courant avec la commande **cp**, et on nomme la copie **mot\_de\_passe**.

```
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 déc. 15 13:36 liens
-rw-r--r-- 1 guest guest 1475 déc. 16 13:36 mot_de_passe
guest@DS1:~$ _
```

On copie les fichiers **/etc/group** et **/etc/profile** dans le répertoire courant indiqué par le ".", grâce à la commande **cp**. On liste ensuite les fichiers, avec la commande **ls -l**.

```
guest@DS1:~$ cp /etc/group /etc/profile .
guest@DS1:~$ ls -l
total 16
-rw-r--r-- 1 guest guest 720 déc. 16 13:37 group
drwxr-xr-x 2 guest guest 4096 déc. 15 13:36 liens
-rw-r--r-- 1 guest guest 1475 déc. 16 13:36 mot_de_passe
-rw-r--r-- 1 guest guest 767 déc. 16 13:37 profile
guest@DS1:~$ _
```

On liste l'ensemble des fichiers du répertoire y compris ceux qui sont cachés (grâce au **a**, *signifiant all*), avec la commande **ls -la**.

```
guest@DS1:~$ ls -a
. .. .bash_logout .bashrc .gnupg group liens mot_de_passe .profile profile
guest@DS1:~$ ls -la
total 40
drwxr-xr-x 4 guest guest 4096 déc. 16 13:37 .
drwxr-xr-x 5 root root 4096 déc. 15 13:27 ..
-rw-r--r-- 1 guest guest 220 déc. 15 13:27 .bash_logout
-rw-r--r-- 1 guest guest 3526 déc. 15 13:27 .bashrc
drwx----- 3 guest guest 4096 déc. 15 13:28 .gnupg
-rw-r--r-- 1 guest guest 720 déc. 16 13:37 group
drwxr-xr-x 2 guest guest 4096 déc. 15 13:36 liens
-rw-r--r-- 1 guest guest 1475 déc. 16 13:36 mot_de_passe
-rw-r--r-- 1 guest guest 807 déc. 15 13:27 .profile
-rw-r--r-- 1 guest guest 767 déc. 16 13:37 profile
guest@DS1:~$ _
```

On crée un répertoire `un_rep`, grâce à la commande `mkdir un_rep`. On liste les fichiers avec leurs attributs, en utilisant `ls -l`.

```
guest@DS1:~$ mkdir un_rep
guest@DS1:~$ ls -l
total 20
-rw-r--r-- 1 guest guest 720 déc. 16 13:37 group
drwxr-xr-x 2 guest guest 4096 déc. 15 13:36 liens
-rw-r--r-- 1 guest guest 1475 déc. 16 13:36 mot_de_passe
-rw-r--r-- 1 guest guest 767 déc. 16 13:37 profile
drwxr-xr-x 2 guest guest 4096 déc. 16 13:41 un_rep
guest@DS1:~$ ls -F
group liens/ mot_de_passe profile un_rep/
guest@DS1:~$
```

On copie le fichier `group` en `g2` avec la commande `cp group g2`, on détruit le fichier `group` avec la commande `rm group`, et on renomme le fichier `g2` en `group` grâce à `mv g2 group`.

```
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' du 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 group
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
guest@DS1:~$
```

On se déplace dans le répertoire `/usr/bin`, grâce à la commande `cd /usr/bin`, puis on vérifie que l'on est bien dans ce répertoire avec la commande `pwd`. On liste le contenu du répertoire avec `ls -l who`, puis on quitte ce répertoire avec la commande `cd`.

```
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 56128 févr. 28 2019 who
guest@DS1:/usr/bin$ cd
guest@DS1:~$ _
```

On visualise les fichiers `/usr/bin/who` et `/etc/issue`, avec les commandes respectives `file /usr/bin/who` et `file /etc/issue`.

```
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, for GNU/Linux 3.2.0, BuildID[sha1]=53d489dcd4b6a6d102ca46dee9f6d84023a7488f, stripped
guest@DS1:~$ file /etc/issue
/etc/issue: ASCII text
guest@DS1:~$
```

On visualise le contenu du fichier `/etc/network/interfaces`, grâce à la commande `cat /etc/network/interfaces`.

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

On visualise le contenu du fichier `/etc/services` en utilisant la commande `less /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 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
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
ssh         22/tcp          # SSH Remote Login Protocol
telnet     23/tcp
smtp        25/tcp          mail
time        37/tcp          timeserver
time        37/udp          timeserver
rlp         39/udp          resource # resource location
/etc/services
```

On fait la même chose en utilisant la commande `more /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 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
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
ftp         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
rpl         39/udp                resource # resource location

--Plus--(5%)

```

## 2. Les utilitaires

On affiche les 10 premières lignes du fichier `/etc/services`, avec la commande `head /etc/services`.

```

guest@DS1:~$ 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 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.
guest@DS1:~$ _

```

On affiche les 10 dernières lignes du fichier `/etc/services`, avec la commande `tail /etc/services`.

```

guest@DS1:~$ tail /etc/services
vboxd      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@DS1:~$ _

```

On affiche toutes les lignes du fichier `/etc/services` qui contiennent la chaîne de caractères "HTTP" avec la commande `grep HTTP /etc/services`. La modification du prompt se fait dans le `.bashrc`.

```

guest@DS1:~$ grep HTTP /etc/services
http       80/tcp                www                # WorldWideWeb HTTP
hkp        11371/tcp              # OpenPGP HTTP Keyserver
guest@DS1:~$

```

On affiche la même chose, mais cette fois-ci sans tenir compte de la casse. Pour ce faire, on utilise la commande **grep -i HTTP /etc/services**.

```
guest@DS1:~$ 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
http-alt  8080/tcp    webcache    # WWW caching service
http-alt  8080/udp
hkp       11371/tcp   # OpenPGP HTTP Keyserver
guest@DS1:~$
```

On affiche toutes les lignes du fichier qui commencent par “http”. On utilise la commande **grep ^http /etc/services**.

```
guest@DS1:~$ grep '^http' /etc/services
http      80/tcp      www         # WorldWideWeb HTTP
https     443/tcp     # http protocol over TLS/SSL
http-alt  8080/tcp    webcache    # WWW caching service
http-alt  8080/udp
guest@DS1:~$ _
```

On trie dans un premier temps le fichier /etc/services, grâce à la commande **sort /etc/services**, puis on affiche ses 3 dernières lignes avec la commande **tail -3**.

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

On va trier, page par page, les lignes du fichier /etc/services qui ne commencent pas par “#”.

Pour trier, on utilisera la commande **sort**, pour afficher page par page, on utilisera la commande **more**, puis pour afficher les lignes du fichier /etc/services qui ne commencent pas par “#”, on se servira de la commande **grep -v ^# /etc/services**.

```
guest@DS1:~$ grep -v '^#' /etc/services | sort | more
```

```

acr-nema      104/tcp      dicom      # Digital Imag. & Comm. 300
afbackup     2988/tcp    # Afbackup system
afbackup     2988/udp
afmbackup    2989/tcp    # Afmbackup system
afmbackup    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
afs3-update  7008/udp
afs3-vlserver 7003/tcp    # volume location database
afs3-vlserver 7003/udp
afs3-volser  7005/tcp    # volume managment server
afs3-volser  7005/udp
amanda       10080/tcp   # amanda backup services
amanda       10080/udp
amandaidx    10082/tcp   # amanda backup services
--Plus--

```

On va afficher le nombre de lignes, de mots et de caractères du fichier `/etc/services`, avec la commande `wc /etc/services`.

```

guest@DS1:~$ wc /etc/services
 578 2558 18774 /etc/services
guest@DS1:~$

```

Pour afficher uniquement le nombre de lignes, on utilise la commande `wc -l /etc/services`.

```

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

```

On se connecte en tant que root, puis on crée un fichier `etudiants.txt` à l'aide de l'éditeur Nano.

```

GNU nano 3.2      etudiants.txt
Nicolas
Denis
Souphiane
Azer
Fabien
Cedric
Tiphaine
Xavier
Antoine
David_

```

On trie les prénoms par ordre alphabétique en utilisant la commande `sort etudiants.txt`.

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

Pour les trier dans l'ordre inverse, on utilise la commande **sort -r etudiants.txt**.

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

On va enregistrer le résultat du tri dans un fichier que l'on appellera `prenom_tries`, grâce à la commande **sort -o prenom\_tries etudiants.txt**, puis on va lister son contenu avec **ls -l**, et on va l'afficher en utilisant **cat prenom\_tries**.

```
root@DS1: ~#sort -o prenom_tries etudiants.txt
root@DS1: ~#ls -l
total 8
-rw-r--r-- 1 root root 73 déc. 16 14:30 etudiants.txt
-rw-r--r-- 1 root root 73 déc. 16 14:32 prenom_tries
root@DS1: ~#cat prenom_tries
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

On va conserver les 3 premiers caractères de chaque ligne du fichier, en utilisant la commande **cut -c -3 etudiants.txt**.

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

On va conserver les caractères 2 à 5 de chaque ligne du fichier, avec la commande **cut -c 2-5 etudiants.txt**.

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

On va conserver tous les caractères à partir du 3ème (inclus) de chaque ligne du fichier, en utilisant la commande **cut -c 3- etudiants.txt**.

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

On va ouvrir le fichier `prenom_tries` dans l'éditeur Nano, le compléter avec les informations ci-dessous, et le renommer `notes.csv`.

```
GNU nano 3.2 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 la commande `cut -d , -f 1 notes.csv`.

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

On conserve les 3 premières colonnes du fichier, avec la commande `cut -d , -f -3 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
root@DS1: ~#
```

On affiche les colonnes 1 et 3 du fichier, avec la commande `cut -d , -f 1,3 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
root@DS1: ~#_
```