

TP10 : Les commandes – Les principaux caractères du shell

Sommaire

<i>1 - Les redirections (>, >>, 2>, 2>>, 2>&1 et)</i>	2
<i>2 - Utilisation des jokers</i>	10
<i>3 - Utilisation des caractères de protection</i>	12
<i>4 - Autres caractères</i>	13

1 - Les redirections (>, >>, 2>, 2>>, 2>&1 et |)

→ On rentre le calendrier (**cal**) dans un nouveau fichier **cal.txt** puis on visualise le fichier créé (**cat**):

```
guest@DEB12Server:~$ cal > cal.txt
guest@DEB12Server:~$ cat cal.txt
  Décembre 2024
di lu ma me je ve sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

→ On **écrase** le contenu du fichier en rentrant la commande **date (date)** dans le même fichier et on le visualise :

```
guest@DEB12Server:~$ date > cal.txt
guest@DEB12Server:~$ cat cal.txt
ven. 20 déc. 2024 08:09:48 CET
```

→ On rentre le résultat de la commande **cal** ainsi que de la commande **date** dans le fichier **histo.txt** sans écraser le contenu puis on visualise tout avec **more** :

```
guest@DEB12Server:~$ cal > histo.txt
guest@DEB12Server:~$ date >> histo.txt
guest@DEB12Server:~$ more histo.txt_
```

```
  Décembre 2024
di lu ma me je ve sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

ven. 20 déc. 2024 08:16:10 CET
```

→ On **supprime** le contenu du fichier en écrasant tout le contenu pour ne rien mettre à la place puis on le visualise :

```
guest@DEB12Server:~$ > histo.txt
guest@DEB12Server:~$ cat histo.txt
guest@DEB12Server:~$ _
```

3

→ On **redirige** les valeurs du premier champ du fichier **notes.csv** dans un nouveau fichier qu'on appelle **eleves.txt** avec **cut** :

```
root@DEB12Server: ~#cut -d , -f 1 notes.csv > eleves.txt
root@DEB12Server: ~#ls -l
total 20
-rw-r--r-- 1 root root  73 20 déc. 15:51 eleves.txt
-rw-r--r-- 1 root root  73 20 déc. 15:47 etudiants.txt
drwxr-xr-x 2 root root 4096 18 déc. 09:56 liens
-rw-r--r-- 1 root root 211 20 déc. 15:50 notes.csv
-rw-r--r-- 1 root root 211 20 déc. 15:49 prenom_tries
root@DEB12Server: ~#cat eleves.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

→ On redirige encore une fois les valeurs du premier champ du fichier **notes.csv** dans **eleves.txt** mais on n'écrase pas le contenu :

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

4

→ On **trie** le fichier **eleves.txt** avec **sort** et on envoie le résultat dans un nouveau fichier **avecdoublons.txt** :

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

→ On supprime les doublons grâce à la commande **uniq** et on envoie le résultat dans un nouveau fichier **sansdoublons.txt** :

```
root@DEB12Server: ~#uniq avecdoublons > sansdoublons
root@DEB12Server: ~#cat sansdoublons
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

→ On affiche une **sortie d'erreur** sur la console :

```
root@DEB12Server: ~#cat fichier_inexistant
cat: fichier_inexistant: Aucun fichier ou dossier de ce type
```

→ On essaye de **rediriger** la sortie d'un fichier qui **n'existe pas** dans **eleves.txt** :

```
root@DEB12Server: ~#cut -d , -f 1 fichier_inexistant.csv > eleves.txt
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type
root@DEB12Server: ~#cat eleves.txt
root@DEB12Server: ~#
```

*** On a toujours une sortie d'erreur sur la console et le fichier **eleves.txt** est vide car on a écrasé ses données par aucunes données.**

→ On **enregistre la sortie d'erreur** de cette même commande dans un fichier **erreurs.log** plutôt que dans la console :

```
root@DEB12Server: ~#cut -d , -f 1 fichier_inexistant.csv > eleves.txt 2> erreurs.log
root@DEB12Server: ~#ls -l
total 28
-rw-r--r-- 1 root root 146 20 déc. 15:55 avecdoublons
-rw-r--r-- 1 root root  0 20 déc. 16:05 eleves.txt
-rw-r--r-- 1 root root  65 20 déc. 16:05 erreurs.log
-rw-r--r-- 1 root root  73 20 déc. 15:47 etudiants.txt
drwxr-xr-x 2 root root 4096 18 déc. 09:56 liens
-rw-r--r-- 1 root root 211 20 déc. 15:50 notes.csv
-rw-r--r-- 1 root root 211 20 déc. 15:49 prenoms_tries
-rw-r--r-- 1 root root  73 20 déc. 15:57 sansdoublons
```

→ On redirige la **sortie standard** ainsi que la **sortie d'erreurs** dans un fichier **sio1.txt** :

```
root@DEB12Server: ~#cut -d , -f 1 fichier_inexistant.csv > sio1.txt 2>&1
root@DEB12Server: ~#cat sio1.txt
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type
```

→ On redirige la même chose mais sans écraser le contenu de **sio1.txt** :

```
root@DEB12Server: ~#cut -d , -f 1 fichier_inexistant.csv >> sio1.txt 2>&1
root@DEB12Server: ~#cat sio1.txt
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type
```

→ On affiche le résultat d'une commande **page par page** avec **| more** :

```

root@DEB12Server: ~#cal 2024 | more
2024
   Janvier                Février                Mars
di lu ma me je ve sa di lu ma me je ve sa di lu ma me je ve sa
   1  2  3  4  5  6           1  2  3           1  2
  7  8  9 10 11 12 13    4  5  6  7  8  9 10    3  4  5  6  7  8  9
14 15 16 17 18 19 20    11 12 13 14 15 16 17    10 11 12 13 14 15 16
21 22 23 24 25 26 27    18 19 20 21 22 23 24    17 18 19 20 21 22 23
28 29 30 31             25 26 27 28 29             24 25 26 27 28 29 30
                                   31

   Avril                Mai                Juin
di lu ma me je ve sa di lu ma me je ve sa di lu ma me je ve sa
   1  2  3  4  5  6           1  2  3  4           1
  7  8  9 10 11 12 13    5  6  7  8  9 10 11    2  3  4  5  6  7  8
14 15 16 17 18 19 20    12 13 14 15 16 17 18    9 10 11 12 13 14 15
21 22 23 24 25 26 27    19 20 21 22 23 24 25    16 17 18 19 20 21 22
28 29 30                 26 27 28 29 30 31    23 24 25 26 27 28 29
                                   30

   Juillet                Août                Septembre
di lu ma me je ve sa di lu ma me je ve sa di lu ma me je ve sa
   1  2  3  4  5  6           1  2  3           1  2  3  4  5  6  7
  7  8  9 10 11 12 13    4  5  6  7  8  9 10    8  9 10 11 12 13 14
14 15 16 17 18 19 20    11 12 13 14 15 16 17    15 16 17 18 19 20 21
21 22 23 24 25 26 27    18 19 20 21 22 23 24    22 23 24 25 26 27 28
28 29 30 31             25 26 27 28 29 30 31    29 30

   Octobre                Novembre                Décembre
di lu ma me je ve sa di lu ma me je ve sa di lu ma me je ve sa
   1  2  3  4  5           1  2           1  2  3  4  5  6  7
  6  7  8  9 10 11 12    3  4  5  6  7  8  9    8  9 10 11 12 13 14
13 14 15 16 17 18 19    10 11 12 13 14 15 16    15 16 17 18 19 20 21
20 21 22 23 24 25 26    17 18 19 20 21 22 23    22 23 24 25 26 27 28
27 28 29 30 31         24 25 26 27 28 29 30    29 30 31

```

→ On affiche les **3 dernières lignes** d'un fichier trié (**tail**) :

```

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

```

7

→ On affiche page par page les lignes d'un fichier qui ne commencent pas par # triées en excluant les lignes contenant un mot donné grâce à -v :

```
root@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
```

→ On **classe** les valeurs du premier champ de **notes.csv** par ordre **alphabétique** :

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

→ On **enregistre la sortie** de cette commande dans un fichier **prenoms_tries.txt** :

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

→ On affiche la liste des utilisateurs du système local page par page :

```
root@DEB12Server: ~#cat /etc/passwd | more
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

→ On affiche la liste des GID :

```
root@DEB12Server: ~#cut -d : -f 4 /etc/passwd | sort -n | uniq
```

```
0
1
2
3
7
```

→ On affiche le nombre de lignes et de caractères contenues dans la liste des fichiers du répertoire personnel de **root** :

```
root@DEB12Server: ~#ls -l | wc
```

```
11 92 569
```

```
root@DEB12Server: ~#ls -l
```

```
total 36
-rw-r--r-- 1 root root 146 20 déc. 15:55 avecdoublons
-rw-r--r-- 1 root root 0 20 déc. 16:05 eleves.txt
-rw-r--r-- 1 root root 65 20 déc. 16:05 erreurs.log
-rw-r--r-- 1 root root 73 20 déc. 15:47 etudiants.txt
drwxr-xr-x 2 root root 4096 18 déc. 09:56 liens
-rw-r--r-- 1 root root 211 20 déc. 15:50 notes.csv
-rw-r--r-- 1 root root 211 20 déc. 15:49 prenoms_tries
-rw-r--r-- 1 root root 73 20 déc. 16:20 prenoms_tries.txt
-rw-r--r-- 1 root root 73 20 déc. 15:57 sansdoublons
-rw-r--r-- 1 root root 130 20 déc. 16:09 sio1.txt
```

→ On affiche tout les fichiers **.txt** présents dans le répertoire personnel de **root** :

```
root@DEB12Server: ~#ls -l | grep .txt
```

```
-rw-r--r-- 1 root root 0 20 déc. 16:05 eleves.txt
-rw-r--r-- 1 root root 73 20 déc. 15:47 etudiants.txt
-rw-r--r-- 1 root root 73 20 déc. 16:20 prenoms_tries.txt
-rw-r--r-- 1 root root 130 20 déc. 16:09 sio1.txt
```

→ On affiche la ligne du fichier **notes.csv** qui contient le mot **"azer"** :

```
root@DEB12Server: ~#cat notes.csv | grep -i azer  
Azer, SISR, 13, AB
```

2 - Utilisation des jokers

→ On se connecte en tant que **guest** et on se déplace dans le répertoire **/bin** grâce à **cd /bin** :

```
guest@DEB12Server:/bin$ _
```

→ On affiche toutes les commandes commençant par r du répertoire **/bin** :

```
guest@DEB12Server:/bin$ ls r*
ranlib  rdma      readlink  rename.ul  reportbug  reset
rbash  readelf  realpath  renice     report-hw  resizecons
```

→ On affiche les commandes du répertoire qui contiennent 5 caractères :

```
guest@DEB12Server:/bin$ ls ??????
b2sum  colrm  gzexe  lsmod  nstat  prove  split  xargs
bzcat  diff3  iconv  lspci  pager  pzstd  sprof  xauth
bzcmp  dmesg  ijoin  lsusb  partx  rbash  strip  xzcat
bzexe  egrep  ipcmk  lzcat  paste  reset  tload  xzcmp
bzip2  eject  ipcrm  lzcmp  patch  ngrep  touch  zdiff
chage  false  lastb  mandb  pgrep  rmdir  troff  zdump
chcon  fgrep  ld.so  mkdir  pidof  rnano  tsort  zgrep
chgrp  flock  login  mknod  ping4  rview  uname  zless
chmod  gmake  lsblk  mount  ping6  sdiff  users  zmore
choom  gpgsm  lscpu  namei  pinky  shred  watch
chown  gprof  lsipc  nohup  pkill  skill  wdctl
cksum  groff  lsirq  nproc  pl2pm  sleep  which
clear  groups  lsmem  nroff  print  snice  write
```

→ On affiche les commandes qui contiennent 2 caractères et commencent par e :

```
guest@DEB12Server:/bin$ ls e?
ex
```

→ On affiche les commandes commençant par w, x, y ou z :

```
guest@DEB12Server:/bin$ ls [wxyz]*
```

```
guest@DEB12Server:/bin$ ls [w-z]*
```

```
x86_64-linux-gnu-gp-display-html  
x86_64-linux-gnu-gp-display-src  
x86_64-linux-gnu-gp-display-text  
x86_64-linux-gnu-gprof  
x86_64-linux-gnu-gprofng  
x86_64-linux-gnu-ld  
x86_64-linux-gnu-ld.bfd  
x86_64-linux-gnu-ld.gold  
x86_64-linux-gnu-lto-dump
```

→ On affiche les commandes qui ne commencent pas par une lettre comprise entre a et v (! pour la négation):

```
guest@DEB12Server:/bin$ ls [!a-v]*_
```

```
x86_64-linux-gnu-gp-display-html  
x86_64-linux-gnu-gp-display-src  
x86_64-linux-gnu-gp-display-text  
x86_64-linux-gnu-gprof  
x86_64-linux-gnu-gprofng  
x86_64-linux-gnu-ld  
x86_64-linux-gnu-ld.bfd  
x86_64-linux-gnu-ld.gold
```

3 - Utilisation des caractères de protection

→ On utilise un **backslash** pour **ne pas interpréter** le caractère suivant comme une commande et pouvoir créer un fichier débutant par # qui est normalement un commentaire :

```
guest@DEB12Server:~$ touch #commentaire
touch: op rande de fichier manquant
Saisissez « touch --help » pour plus d'informations.
guest@DEB12Server:~$ touch \#commentaire
guest@DEB12Server:~$ ls -l *comme*
-rw-r--r-- 1 guest guest 0 20 d c. 16:44 '#commentaire'
```

→ On utilise des **simples quotes** pour pouvoir cr er un fichier contenant des espaces dans son nom :

```
guest@DEB12Server:~$ touch 'Fichier dont le nom contient des espaces'
guest@DEB12Server:~$ ls -l *contient*
-rw-r--r-- 1 guest guest 0 21 d c. 12:52 'Fichier dont le nom contient des espaces'
```

→ On utilise des **doubles quotes** pour inclure une **variable** dans un texte :

```
guest@DEB12Server:~$ echo "Mon r pertoire de connexion : $HOME"
Mon r pertoire de connexion : /home/guest
```

4 - Autres caractères

→ On utilise un **point-virgule** pour **séparer** plusieurs commandes :

```
guest@DEB12Server:~$ echo "voici la date :" ; date
voici la date :
sam. 21 déc. 2024 12:59:16 CET
```

→ On se sert des **anti-quotes** pour qu'une commande soit **interprétée** dans une autre commande :

```
guest@DEB12Server:~$ echo "Voici la date : `date`"
Voici la date : sam. 21 déc. 2024 13:01:10 CET
```

→ On regarde la différence entre **echo date** et **echo `date`** :

```
guest@DEB12Server:~$ echo date
date
guest@DEB12Server:~$ echo `date`
sam. 21 déc. 2024 13:02:07 CET
```

* **echo date** affiche simplement "date" car **echo** sert à afficher du texte et que **date** est interprétée comme du texte du fait qu'elle est dans une autre commande.

echo `date` affiche la date car les anti-quotes permettent à **echo** d'interpréter **date** comme une commande.