

TP7: Les commandes – Les principaux caractères du shell

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

```
guest@DS1:~$ cal > cal.txt
guest@DS1:~$ cat cal.txt
  Décembre 2021
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
```

1ère étape : Redirection du résultat de la commande **cal** dans un fichier pour ensuite visualiser le fichier créer avec les commandes « **cal > cal.txt** » et « **cat cal.txt** ».

```
guest@DS1:~$ date > cal.txt
guest@DS1:~$ cat cal.txt
mar. 07 déc. 2021 16:55:08 CET
```

2ème étape : Redirection du résultat de la commande **date** dans le même fichier, en effectuant « **date > cal.txt** » et « **cat cal.txt** ».

```
guest@DS1:~$ cal > histo.txt
guest@DS1:~$ date >> histo.txt
guest@DS1:~$ more histo.txt
  Décembre 2021
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
mar. 07 déc. 2021 16:56:16 CET
```

3ème étape : Redirection du résultat de la commande **cal** dans le fichier indiqué, puis ajout du résultat de la commande **date**, en saisissant « **cal > histo.txt** », « **date >> histo.txt** » et « **more histo.txt** ».

```
guest@DS1:~$ > histo.txt
guest@DS1:~$ cat histo.txt
```

4ème étape : Suppression du contenu du fichier par une redirection, avec « > histo.txt » et « cat histo.txt ».

```
root@DS1 ~#cut -d , -f 1 notes.csv > eleves.txt
root@DS1 ~#ls -l
total 20
-rw-r--r-- 1 root root 73 7 déc. 16:58 eleves.txt
-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 211 7 déc. 16:49 notes.csv
-rw-r--r-- 1 root root 73 7 déc. 16:42 prenom_tries
root@DS1 ~#cat eleves.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

5ème étape : Redirection des valeurs du premier champs du fichier **note.csv** dans un nouveau fichier nommé **eleves.txt** . Nous saisissons alors « cut -d , -f 1 notes.csv > eleves.txt », « cat eleves.txt » puis « ls -l ».

```

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

```

6ème étape : Même manipulation sans écraser le contenu du fichier **eleves.txt**, puis affichage de son contenu avec les commandes « **cut -d , -f 1 notes.csv** » **eleves.txt** » et « **cat eleves.txt** ».

```

root@DS1 ~#sort eleves.txt > avecdoublons
root@DS1 ~#cat avecdoublons
Antoine
Antoine
Azer
Azer
Cedric
Cedric
David
David
Denis
Denis
Fabien
Fabien
Nicolas
Nicolas
Souphiane
Souphiane
Tiphaine
Tiphaine
Xavier
Xavier

```

7ème étape : Nous effectuons un tri sur le contenu du fichier **eleves.txt** suivi d'une redirections du résultat dans un nouveau fichier, **avecdoublons.txt**. Les commandes sont donc « **sort eleves.txt > avecdoublons** » et « **cat avecdoublons** ».

```
root@DS1 ~#uniq avecdoublons > sansdoublons
root@DS1 ~#cat sansdoublons
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

8ème étape : Suppressions des doublons avec la commande « **uniq avecdoublons > sansdoublons** » pour le rediriger dans un fichier nommé **sansdoublons.txt**, ensuite nous affichons le contenu dans la console avec « **cat sansdoublons** ».

```
root@DS1 ~#cat fichier_inexistant
cat: fichier_inexistant: Aucun fichier ou dossier de ce type
root@DS1 ~#
```

9ème étape: Tout s’affiche dans la console, la sortie standard ainsi que la sortie d’erreurs, vérification avec la commande « **cat fichier inexistant** ».

```
root@DS1 ~#cut -d , -f 1 fichier_inexistant.csv > eleves.txt
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type
root@DS1 ~#cat eleves.txt
root@DS1 ~#_
```

10ème étape : Si l’on souhaite rediriger la sortie standard dans le fichier, on constate que le message d’erreur est toujours présent avec la commande « **cut -d , -f 1 fichier inexistant.csv > eleves.txt** » puis « **cat eleves.txt** », le fichier **eleves.txt** à également été supprimé.

```

root@DS1 ~#cu -d , -f 1 fichier_inexistant.csv > eleves.txt 2> erreurs.log
root@DS1 ~#ls -l
total 28
-rw-r--r-- 1 root root 146 7 déc. 17:01 avecdoublons
-rw-r--r-- 1 root root 0 7 déc. 17:05 eleves.txt
-rw-r--r-- 1 root root 33 7 déc. 17:05 erreurs.log
-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 211 7 déc. 16:49 notes.csv
-rw-r--r-- 1 root root 73 7 déc. 16:42 prenom_tries
-rw-r--r-- 1 root root 73 7 déc. 17:01 sansdoublons

```

11ème étape : Redirection de l'éventuelle sortie d'erreurs dans un fichier pour le conserver avec la commande « **cut -d , -f 1 fichier_inexistant.csv > eleves.txt 2> erreurs.log** ».

```

root@DS1 ~#cut -d , -f 1 fichier_inexistant.csv > sio1.txt 2>&1
root@DS1 ~#cat sio1.txt
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type

```

12ème étape : Redirection de la sortie standard ou d'une éventuelle sortie d'erreurs seulement dans le fichier **sio1.txt** avec la commande « **cut -d , -f 1 fichier_inexistant.csv > sio1.txt 2>&1** » puis nous saisissons « **cat sio1.txt** ».

```

root@DS1 ~#cut -d , -f 1 fichier_inexistant.csv >> sio1.txt 2>&1
root@DS1 ~#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

```

13ème étape : Cette fois-ci, redirection de la sortie standard ou d'une éventuelle sortie d'erreurs à la fin du fichier **sio1.txt** avec la commande « **cut -d , -f 1 fichier_inexistant.csv >> sio1.txt 2>&1** », puis nous saisissons « **cat sio1.txt** ».

2019																								
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						1	2							1	2			
6	7	8	9	10	11	12	3	4	5	6	7	8	9	3	4	5	6	7	8	9				
13	14	15	16	17	18	19	10	11	12	13	14	15	16	10	11	12	13	14	15	16				
20	21	22	23	24	25	26	17	18	19	20	21	22	23	17	18	19	20	21	22	23				
27	28	29	30	31			24	25	26	27	28			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					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					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								

14ème étape : Ici nous avons affiché le résultat de la commande « cal 2019 | more ».

```
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
```

15ème étape : Nous affichons les trois dernières lignes du fichier trié avec la commande « sort /etc/services | tail -3 ».

```
guest@DS1:~$ 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 management 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
amqps         5671/tcp     # AMQP protocol over TLS/SSL
asf-rmcp      623/udp      # ASF Remote Management and Control Protocol
asp          27374/tcp    # Address Search Protocol
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
obs          7000/tcp
ogp           179/tcp      # Border Gateway Protocol
ogpd          2605/tcp     # bgpd vty (zebra)
oiff         512/udp      comsat
binkp        24554/tcp    # binkp fidonet protocol
--Plus--
```

16ème étape : Nous affichons ici, trié page par page, les lignes du fichier qui ne commencent pas par #, avec la commande « **grep -v '^#' /etc/services | sort | more** ».

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

17ème étape : Nous classons par ordre alphabétique les prénoms présent dans le premier champ du fichier **notes.csv** avec la commande « **cut -d , -f 1 notes.csv | sort** ».

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

18ème étape : Nous enregistrons ensuite la liste dans un fichier nommé **prenomns_tries.txt**, en saisissant la commande « **cut -d , -f 1 notes.csv | sort > prenomns_tries.txt** », puis nous affichons celui-ci avec « **cat prenomns_tries.txt** ».

```
root@DS1 ~#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
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

19ème étape : Nous affichons la liste des utilisateurs page par page via la commande « **cat /etc/passwd | more** », ici nous trouvons le login, le mot de passe, user ID, ect... (De chaque utilisateurs).


```

root@DS1 ~#cut -d : -f4 /etc/passwd | sort -n | uniq
0
1
2
3
7
8
9
10
12
13
33
34
38
39
41
60
101
103
104
110
999
1000
1001
65534

```

20ème étape : Avec la commande « cut -d : -f4 /etc /passwd | sort -n | uniq », nous affichons la liste des GID (identifiant du groupe principal auquel appartient l'utilisateur).

```

root@DS1 ~#ls -l | wc
  11    92   568
root@DS1 ~#ls -l
total 36
-rw-r--r-- 1 root root 146 7 déc. 17:01 avecdoublons
-rw-r--r-- 1 root root  0 7 déc. 17:05 eleves.txt
-rw-r--r-- 1 root root 33 7 déc. 17:05 erreurs.log
-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 211 7 déc. 16:49 notes.csv
-rw-r--r-- 1 root root 73 7 déc. 16:42 prenom_tries
-rw-r--r-- 1 root root 73 7 déc. 17:16 prenom_tries.txt
-rw-r--r-- 1 root root 73 7 déc. 17:01 sansdoublons
-rw-r--r-- 1 root root 130 7 déc. 17:10 sio1.txt

```

21ème étape : Via la commande « ls -l | wc » puis « ls -l » nous affichons le nombre de lignes et de caractères contenus dans la liste des fichiers du répertoire personnel root.

```

root@DS1 ~#ls -l | grep .txt
-rw-r--r-- 1 root root  0 7 déc. 17:05 eleves.txt
-rw-r--r-- 1 root root 73 7 déc. 16:40 etudiants.txt
-rw-r--r-- 1 root root 73 7 déc. 17:16 prenom_tries.txt
-rw-r--r-- 1 root root 130 7 déc. 17:10 sio1.txt

```

22ème étape : Nous affichons les fichiers en .txt se trouvant dans le répertoire root en saisissant la commande « ls -l | grep .txt ».

```

root@DS1 ~#cat notes.csv | grep -i azer
Azer, SISR, 13, AB

```

23ème étape : Pour finir nous affichons la ligne du fichier **notes.csv** qui concerne l'étudiant Azer en effectuant la commande « cat notes.csv | grep -i azer ».

2. Utilisation des jokers.

```

guest@DS1:/bin$ ls r*
ranlib  readelf  reportbug  resizepart  rlogin  routef  rtstat  run-with-aspell
rbash  readlink  report-hw  resolvectl  rm  routel  runcon  rview
rcp  realpath  reset  rev  rmdir  rpcgen  run-mailcap  rvim
rdma  renice  resizecons  rgrep  rnano  rsh  run-parts

```

1ère étape : En saisissant « /bin\$ ls r* » nous affichons les commandes commençant par r du répertoire /bin.

```

guest@DS1:/bin$ ls ?????
b2sum  chgrp  diff3  gmake  ijoin  lsipc  mandb  nroff  pidof  prove  sdiff  strip  watch  xzcmp
bzip2  chmod  dmesg  gpgsm  ipcmk  lsmem  mkdir  nstat  ping4  rbash  shred  tload  wdctl  zdiff
bzcat  chown  egrep  gprof  ipcrm  lsmem  mknod  pager  ping6  reset  skill  touch  which  zdump
bzexe  chown  eject  groff  lastb  lspci  mount  partx  pinky  rgrep  sleep  troff  write  zgrep
bzcat  cksum  false  grops  login  lsusb  namei  paste  pkill  rmdir  snice  tsort  xargs  zless
chage  clear  fgrep  gzexe  lsblk  lzcat  nohup  patch  pl2pm  rnano  split  uname  xauth  zmore
chcon  colrm  flock  iconv  lscpu  lzcmp  nproc  pgrep  print  rview  sprof  users  xzcat

```

2ème étape : En saisissant « /bin\$ ls ????? », nous affichons les commandes contenant 5 caractères.

```
guest@DS1:/bin$ ls e?  
ex
```

3ème étape : En saisissant « **/bin\$ ls e?** », nous affichons les commandes de 2 caractères commençant par **e**.

```
guest@DS1:/bin$ ls [wxyz]*  
w      x86_64-linux-gnu-gcc-10      xauth  
wall   x86_64-linux-gnu-gcc-ar      xsubpp  
watch  x86_64-linux-gnu-gcc-ar-10   xxd  
watchnupg x86_64-linux-gnu-gcc-nm     xz  
wc      x86_64-linux-gnu-gcc-nm-10   xzcat  
wdctl   x86_64-linux-gnu-gcc-ranlib  xzcmp  
wget    x86_64-linux-gnu-gcc-ranlib-10 xzdiff  
whatis  x86_64-linux-gnu-gcov        xzegrep  
whereis x86_64-linux-gnu-gcov-10     xzfgrep  
which   x86_64-linux-gnu-gcov-dump    xzgrep  
whiptail x86_64-linux-gnu-gcov-dump-10 xzless  
who      x86_64-linux-gnu-gcov-tool    xzmore  
whoami   x86_64-linux-gnu-gcov-tool-10 yes  
word-list-compress x86_64-linux-gnu-gold      ypdomainname  
write    x86_64-linux-gnu-gprof      zcat  
write.ul x86_64-linux-gnu-ld         zcmp  
x86_64   x86_64-linux-gnu-ld.bfd     zdiff  
x86_64-linux-gnu-addr2line x86_64-linux-gnu-ld.gold   zdump  
x86_64-linux-gnu-ar      x86_64-linux-gnu-lto-dump-10 zegrep  
x86_64-linux-gnu-as      x86_64-linux-gnu-nm        zfgrep  
x86_64-linux-gnu-c++filt x86_64-linux-gnu-objcopy    zforce  
x86_64-linux-gnu-cpp      x86_64-linux-gnu-objdump    zgrep  
x86_64-linux-gnu-cpp-10   x86_64-linux-gnu-ranlib     zipdetails  
x86_64-linux-gnu-dwp      x86_64-linux-gnu-readelf    zless  
x86_64-linux-gnu-elfedit  x86_64-linux-gnu-size       zmore  
x86_64-linux-gnu-g++      x86_64-linux-gnu-strings    znew  
x86_64-linux-gnu-g++-10   x86_64-linux-gnu-strip  
x86_64-linux-gnu-gcc      xargs
```

```

guest@DS1:/bin$ ls [w-z]*
w                               x86_64-linux-gnu-gcc-10        xauth
wall                            x86_64-linux-gnu-gcc-ar        xsubpp
watch                           x86_64-linux-gnu-gcc-ar-10     xxd
watchgnupg                      x86_64-linux-gnu-gcc-nm        xz
wc                               x86_64-linux-gnu-gcc-nm-10     xzcat
wdctl                           x86_64-linux-gnu-gcc-ranlib    xzcmp
wget                            x86_64-linux-gnu-gcc-ranlib-10 xzdiff
whatis                          x86_64-linux-gnu-gcov          xzegrep
whereis                         x86_64-linux-gnu-gcov-10       xzfgrep
which                           x86_64-linux-gnu-gcov-dump     xzgrep
whiptail                       x86_64-linux-gnu-gcov-dump-10  xzless
who                             x86_64-linux-gnu-gcov-tool     xzmore
whoami                         x86_64-linux-gnu-gcov-tool-10  yes
word-list-compress             x86_64-linux-gnu-gold          ypdomainname
write                          x86_64-linux-gnu-gprof         zcat
write.ul                       x86_64-linux-gnu-ld            zcmp
x86_64                         x86_64-linux-gnu-ld.bfd        zdiff
x86_64-linux-gnu-addr2line      x86_64-linux-gnu-ld.gold       zdump
x86_64-linux-gnu-ar             x86_64-linux-gnu-lto-dump-10   zegrep
x86_64-linux-gnu-as             x86_64-linux-gnu-nm            zfgrep
x86_64-linux-gnu-c++filt       x86_64-linux-gnu-objcopy       zforce
x86_64-linux-gnu-cpp           x86_64-linux-gnu-objdump       zgrep
x86_64-linux-gnu-cpp-10        x86_64-linux-gnu-ranlib        zipdetails
x86_64-linux-gnu-dwp           x86_64-linux-gnu-readelf       zless
x86_64-linux-gnu-elfedit       x86_64-linux-gnu-size          zmore
x86_64-linux-gnu-g++           x86_64-linux-gnu-strings       znew
x86_64-linux-gnu-g++-10        x86_64-linux-gnu-strip
x86_64-linux-gnu-gcc           xargs

```

4ème étape : Nous affichons les commandes commençant par w, x, y ou z avec « /bin\$ ls [w-z]* », de deux manières différentes.

```

guest@DS1:/bin$ ls [!a-v]*
['
VBoxClient          x86_64-linux-gnu-g++      x86_64-linux-gnu-strip
VBoxControl         x86_64-linux-gnu-g++-10  xargs
VBoxDRMClient       x86_64-linux-gnu-gcc      xauth
w                   x86_64-linux-gnu-gcc-10   xsubpp
wall                x86_64-linux-gnu-gcc-ar   xxd
watch               x86_64-linux-gnu-gcc-ar-10 xz
watchnupg           x86_64-linux-gnu-gcc-nm   xzcat
wc                  x86_64-linux-gnu-gcc-nm-10 xzcmp
wdctl               x86_64-linux-gnu-gcc-ranlib xzdiff
wget                x86_64-linux-gnu-gcc-ranlib-10 xzegrep
whatis              x86_64-linux-gnu-gcov     xzfgrep
whereis             x86_64-linux-gnu-gcov-10  xzgrep
which               x86_64-linux-gnu-gcov-dump xzless
whiptail            x86_64-linux-gnu-gcov-dump-10 xzmore
who                 x86_64-linux-gnu-gcov-tool yes
whoami              x86_64-linux-gnu-gcov-tool-10 ypdomainname
word-list-compress x86_64-linux-gnu-gold     zcat
write               x86_64-linux-gnu-gprof    zcmp
write.ul            x86_64-linux-gnu-ld       zdiff
x86_64              x86_64-linux-gnu-ld.bfd   zdump
x86_64-linux-gnu-addr2line x86_64-linux-gnu-ld.gold  zegrep
x86_64-linux-gnu-ar  x86_64-linux-gnu-lto-dump-10 zfgrep
x86_64-linux-gnu-as  x86_64-linux-gnu-nm       zforce
x86_64-linux-gnu-c++filt x86_64-linux-gnu-objcopy  zgrep
x86_64-linux-gnu-cpp x86_64-linux-gnu-objdump  zipdetails
x86_64-linux-gnu-cpp-10 x86_64-linux-gnu-ranlib   zless
x86_64-linux-gnu-dwp x86_64-linux-gnu-readelf  zmore
x86_64-linux-gnu-elfedit x86_64-linux-gnu-size     znew
x86_64-linux-gnu-elfedit x86_64-linux-gnu-strings

```

5ème étape : Cette fois-ci nous affichons les commandes qui ne commencent pas par une lettre comprise entre a et v, en saisissant « /bin\$ ls [!a-v]* ».

3. Utilisation des caractères de protection.

```

guest@DS1:~$ touch #commentaire
touch: op rande de fichier manquant
Saisissez « touch --help » pour plus d'informations.
guest@DS1:~$ touch \#commentaire
guest@DS1:~$ ls -l *comme*
-rw-r--r-- 1 guest guest 0 7 d c. 17:27 '#commentaire'

```

1ère étape : Comme le \ demande au shell de ne pas interpr ter le caract re suivant. Nous essayons donc de cr er un fichier commen ant par un # qui normalement d bute un commentaire, avec les commandes « touch #commentaire », « touch \#commentaire » et « ls -l *comme* ».

```
guest@DS1:~$ touch 'Fichier dont le nom contient des espaces'
guest@DS1:~$ ls -l *espace*
-rw-r--r-- 1 guest guest 0  7 déc. 17:28 'Fichier dont le nom contient des espaces'
```

2ème étape : Ici on crée un fichier là où le nom de celui-ci contient des espaces, en saisissant la commande « **touch 'Fichier dont le nom contient des espaces'** ».

```
guest@DS1:~$ echo "Mon répertoire de connexion : $HOME"
Mon répertoire de connexion : /home/guest
```

3ème étape: Ici nous utilisons les doubles quotes pour que le caractère \$ soit interprété, avec la commande « **echo "Mon répertoire de connexion : \$HOME"** ».

4. Autres caractères.

```
guest@DS1:~$ echo "voici la date : " ; date
voici la date :
mar. 07 déc. 2021 17:30:07 CET
```

1ère étape : On constate ici que le ; sépare les commandes, nous effectuons donc « **echo "voici la date : " ; date** ».

```
guest@DS1:~$ echo "Voici la date : `date`"
Voici la date : mar. 07 déc. 2021 17:32:15 CET
```

2ème étape : Ici les anti-quotes interprètent une commande dans une autre commande ce qui génère dynamiquement des arguments, nous saisissons alors « **echo "Voici la date : `date`"** ».