

TP 10- Les commandes – Les principaux caractères du shell

Nesrine El Ahmadi

BTS SIO

Table des matières

1. Les redirections (>, >>, 2>, 2>>, 2>&1 et)....	2
2. Utilisation des jokers.....	12
3. Utilisation des caractères de protection.....	13
4. Autres caractères.....	14

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

- On redirige le résultat de la commande cal dans un fichier. On visualise le fichier ainsi créé.

```
guestt@DEB13Server:~$ cal > cal.txt
guestt@DEB13Server:~$ cat cal.txt
    Décembre 2025
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 redirige le résultat de la commande date dans le même fichier. L'ancien contenu est perdu.

```
guestt@DEB13Server:~$ date > cal.txt
guestt@DEB13Server:~$ cat cal.txt
ven. 12 déc. 2025 16:03:04 CET
```

- On redirige le résultat de la commande cal dans un fichier. On y ajoute le résultat de la commande date.

```
guestt@DEB13Server:~$ cal > histo.txt
guestt@DEB13Server:~$ date >> histo.txt
guestt@DEB13Server:~$ more histo.txt
    Décembre 2025
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. 12 déc. 2025 16:03:57 CET
```

TP 10- Les commandes – Les principaux caractères du shell

- On supprime le contenu du fichier par une redirection :

```
guestt@DEB13Server:~$ > histo.txt  
guestt@DEB13Server:~$ cat histo.txt
```

- On redirige les valeurs du premier champ du fichier notes.csv dans un nouveau fichier nommé eleves.txt. On constate l'existence de ce dernier et on visualise son contenu (en se loge en root) :

```
root@DEB13Server: ~#cut -d , -f 1 notes.csv > eleves.txt  
root@DEB13Server: ~#ls -l  
total 16  
-rw-r--r-- 1 root root 78 12 déc. 16:08 eleves.txt  
-rw-r--r-- 1 root root 73 11 déc. 16:31 etudiants.txt  
-rw-r--r-- 1 root root 212 11 déc. 16:51 notes.csv  
-rw-r--r-- 1 root root 73 11 déc. 16:36 prenoms_tries  
root@DEB13Server: ~#cat eleves.txt  
Antoine  
Azer  
Cedric  
David  
Denis  
Fabien  
Nicolas  
Souphiane  
Tiphaine  
Xavier  
SLAM  
root@DEB13Server: ~#
```

- On recommence la même opération mais sans écraser le contenu du fichier eleves.txt. On affiche son contenu dans la console :

```
root@DEB13Server: ~#cut -d , -f 1 notes.csv >> eleves.txt
root@DEB13Server: ~#cat eleves.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
SLAM
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
SLAM
```

- On effectue un tri sur le contenu du fichier eleves.txt et redirigez le résultat dans un nouveau fichier nommé avecdoublons.txt. On affiche son contenu dans la console :

TP 10- Les commandes – Les principaux caractères du shell

```
root@DEB13Server: ~#sort eleves.txt > avecdoublons
root@DEB13Server: ~#cat avecdoublons
Antoine
Antoine
Azer
Azer
Cedric
Cedric
David
David
Denis
Denis
Fabien
Fabien
Nicolas
Nicolas
SLAM
SLAM
Souphiane
Souphiane
Tiphaine
Tiphaine
Xavier
Xavier
```

- On supprime les doublons avec la commande uniq et redirigez le résultat dans un nouveau fichier nommé sansdoublons.txt. On affiche son contenu dans la console :

```
root@DEB13Server: ~#uniq avecdoublons > sansdoublons
root@DEB13Server: ~#cat sansdoublons
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
SLAM
Souphiane
Tiphaine
Xavier
root@DEB13Server: ~#
```

Remarque : la commande uniq peut évidemment afficher le résultat dans la console. Si l'on veut enregistrer le résultat dans un fichier, elle peut être utilisée directement sans passer par une redirection :

```
root@DEB13Server: ~#uniq avecdoublons pasdedoublons
root@DEB13Server: ~#
```

- Par défaut, tout s'affiche dans la console, la sortie standard comme la sortie d'erreurs :

```
root@DEB13Server: ~#cat fichier_inexistant
cat: fichier_inexistant: Aucun fichier ou dossier de ce nom
root@DEB13Server: ~#
```

- Si l'on désire rediriger la sortie standard dans un fichier, le message d'erreur continue de s'afficher dans la console (on remarque également que le contenu du fichier eleves.txt a été supprimé) :

- On redirige l'éventuelle sortie d'erreurs dans un fichier à part pour en conserver une trace. On affiche le contenu du fichier erreurs.log :

```
root@DEB13Server: ~#cut -d , -f 1 fichier_inexistant.csv > eleves.txt 2> erreurs.log
root@DEB13Server: ~#ls -l
total 28
-rw-r--r-- 1 root root 156 12 déc. 16:11 avecdoublons
-rw-r--r-- 1 root root 0 12 déc. 16:15 eleves.txt
-rw-r--r-- 1 root root 64 12 déc. 16:15 erreurs.log
-rw-r--r-- 1 root root 73 11 déc. 16:31 etudiants.txt
-rw-r--r-- 1 root root 212 11 déc. 16:51 notes.csv
-rw-r--r-- 1 root root 78 12 déc. 16:13 pasdedoublons
-rw-r--r-- 1 root root 73 11 déc. 16:36 prenoms_tries
-rw-r--r-- 1 root root 78 12 déc. 16:12 sansdoublons
root@DEB13Server: ~#
```

- On redirige la sortie standard ou l'éventuelle sortie d'erreurs dans un seul et même fichier nommé sio1.txt. On affiche son contenu :

```
root@DEB13Server: ~#cut -d , -f 1 fichier_inexistant.csv > sio1.txt 2>&1
root@DEB13Server: ~#cat sio1.txt
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce nom
```

- On redirige la sortie standard ou l'éventuelle sortie d'erreurs à la fin du fichier sio1.txt. On affiche son contenu :

```
root@DEB13Server: ~#cut -d , -f 1 fichier_inexistant.csv >> sio1.txt 2>&1
root@DEB13Server: ~#cat sio1.txt
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce nom
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce nom
root@DEB13Server: ~#
```

TP 10- Les commandes – Les principaux caractères du shell

- On affiche le résultat d'une commande page par page : cal 2025 | more et touche espace (chainage de 2 commandes à l'aide du caractère « pipe » |).

```
root@DEB13Server: ~#cal 2025 | more
          2025
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           1 2 3 4 5       1 2 3 4 5 6 7 8
 5 6 7 8 9 10 11    2 3 4 5 6 7 8   2 3 4 5 6 7 8
12 13 14 15 16 17 18  9 10 11 12 13 14 15  9 10 11 12 13 14 15
19 20 21 22 23 24 25 16 17 18 19 20 21 22 16 17 18 19 20 21 22
26 27 28 29 30 31    23 24 25 26 27 28   23 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           1 2 3 4 5 6 7
 6 7 8 9 10 11 12    4 5 6 7 8 9 10   8 9 10 11 12 13 14
13 14 15 16 17 18 19 11 12 13 14 15 16 17 15 16 17 18 19 20 21
20 21 22 23 24 25 26 18 19 20 21 22 23 24 22 23 24 25 26 27 28
27 28 29 30          25 26 27 28 29 30 31 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           1 2 3 4 5 6
 6 7 8 9 10 11 12    3 4 5 6 7 8 9   7 8 9 10 11 12 13
13 14 15 16 17 18 19 10 11 12 13 14 15 16 14 15 16 17 18 19 20
20 21 22 23 24 25 26 17 18 19 20 21 22 23 21 22 23 24 25 26 27
27 28 29 30 31        24 25 26 27 28 29 30 28 29 30
                           31

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           1 2 3 4 5       1 2 3 4 5 6
 5 6 7 8 9 10 11    2 3 4 5 6 7 8   7 8 9 10 11 12 13
12 13 14 15 16 17 18  9 10 11 12 13 14 15 14 15 16 17 18 19 20
19 20 21 22 23 24 25 16 17 18 19 20 21 22 21 22 23 24 25 26 27
26 27 28 29 30 31    23 24 25 26 27 28 29 28 29 30 31
                           30
root@DEB13Server: ~#
```

- On affiche les 3 dernières lignes d'un fichier trié.

```
guesstt@DEB13Server:~$ sort /etc/services | tail -3
zope          9678/tcp                      # zope server
zope-ftp      8021/tcp                      # zope management by ftp
zserv         346/tcp                       # Zebra server
guesstt@DEB13Server:~$
```

- On affichez triées, page par page, les lignes d'un fichier qui ne commencent pas par « # ».

```
guestt@DEB13Server:~$ grep -v '^#' /etc/services | sort | more
```

Le paramètre -v permet de connaître toutes les lignes qui ne contiennent pas un mot donné.

```
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  # AFS/Kerberos authentication
afs3-kaserver 7004/udp    # users & groups database
afs3-prserver 7002/udp    # remote cache manager service
afs3-rmtsys   7009/udp    # server-to-server updater
afs3-update   7008/udp    # volume location database
afs3-vlserver 7003/udp    # volume managment server
afs3-volser   7005/udp    # amanda backup services
amanda        10080/tcp   # amanda backup services
amandaidx    10082/tcp   # amanda backup services
amidxtape    10083/tcp   # amanda backup services
amqp          5672/sctp   # AMQP protocol over TLS/SSL
amqp          5672/tcp     # ASF Remote Management and Control Protocol
amqps         5671/tcp     # Address Search Protocol
asp           623/udp      # ASP
asp           27374/tcp   # Address Search Protocol
asp           27374/udp   # ASP
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    # Border Gateway Protocol
bgp           179/tcp      # bgpd vty (zebra)
bgpd          2605/tcp    # binkp fidonet protocol
biff          512/udp      # cannaserver
binkp         24554/tcp   # Cisco SCCP
bootpc        68/udp      # Common lisp build daemon
bootps        67/udp      # Clearcase
canna         5680/tcp    # cmip-agent
cfengine      5308/tcp    # cmip-man
chargen       19/tcp       ttytst source
chargen       19/udp      ttytst source
cisco-sccp    2000/tcp    # Cisco SCCP
clc-build-daemon 8990/tcp # cmip-agent
clearcase     371/udp      # cmip-man
cmip-agent    164/tcp      # ISO mgmt over IP (CMOT)
cmip-agent    164/udp
cmip-man     163/tcp      # cmip-man
cmip-man     163/udp
--Plus--
```

- On classe par ordre alphabétique les prénoms figurant dans le premier champ du fichier notes.csv :

TP 10- Les commandes – Les principaux caractères du shell

```
root@DEB13Server: ~#cut -d , -f 1 notes.csv | sort
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
SLAM
Souphiane
Tiphaine
Xavier
root@DEB13Server: ~#_
```

- On enregistre cette liste dans un fichier nommé prenoms_tries.txt :

```
root@DEB13Server: ~#cut -d , -f 1 notes.csv | sort > prenoms_tries.txt
root@DEB13Server: ~#cat prenoms_tries.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
SLAM
Souphiane
Tiphaine
Xavier
root@DEB13Server: ~#_
```

- On affiche la liste des utilisateurs du système local, page par page, à l'aide de la commande cat /etc/passwd | more. On trouve, dans ce fichier, les champs login, mot de passe, user ID, group ID, descriptif, répertoire personnel et shell de l'utilisateur.

```
root@DEB13Server: ~#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:/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
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon:/usr/lib/dhcpcd:/bin/false
systemd-timesync:x:991:991:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:990:990:System Message Bus:/nonexistent:/usr/sbin/nologin
sio:x:1000:1000:sio,,,,:/home/sio:/bin/bash
vboxadd:x:989:1::/var/run/vboxadd:/bin/false
sshd:x:987:65534:sshd user:/run/sshd:/usr/sbin/nologin
guest:x:1001:1001::/home/guest:/bin/bash
guestt:x:1002:1002:guestt,,,,:/home/guestt:/bin/bash
root@DEB13Server: ~#
```

- On affiche la liste des GID (identifiant du groupe principal auquel appartient l'utilisateur) à l'aide de la commande cut -d : -f 4 /etc/passwd | sort -n | uniq :

```
root@DEB13Server: ~#cut -d : -f 4 /etc/passwd | sort -n | uniq
0
1
2
3
7
8
9
10
12
13
33
34
38
39
60
990
991
998
1000
1001
1002
65534
root@DEB13Server: ~#
```

TP 10- Les commandes – Les principaux caractères du shell

- On affiche le nombre de lignes et de caractères contenus dans la liste des fichiers du répertoire personnel de root :

```
root@DEB13Server: ~#ls -l | wc
      12      101      618
root@DEB13Server: ~#ls -l
total 40
-rw-r--r-- 1 root root 156 12 déc. 16:11 avecdoublons
-rw-r--r-- 1 root root   0 12 déc. 16:15 eleves.txt
-rw-r--r-- 1 root root  64 12 déc. 16:15 erreurs.log
-rw-r--r-- 1 root root  73 11 déc. 16:31 etudiants.txt
-rw-r--r-- 1 root root 212 11 déc. 16:51 notes.csv
-rw-r--r-- 1 root root  78 12 déc. 16:13 pasdedoublons
-rw-r--r-- 1 root root  73 11 déc. 16:36 prenoms_tries
-rw-r--r-- 1 root root  78 12 déc. 16:32 prenoms_tries.txt
-rw-r--r-- 1 root root  78 12 déc. 16:12 sansdoublons
-rw-r--r-- 1 root root 128 12 déc. 16:25 sio1.txt
-rw-r--r-- 1 root root 128 12 déc. 16:18 sio.txt
root@DEB13Server: ~#
```

- On affiche les fichiers .txt contenus dans le répertoire personnel de root :

```
root@DEB13Server: ~#ls -l | grep .txt
-rw-r--r-- 1 root root   0 12 déc. 16:15 eleves.txt
-rw-r--r-- 1 root root  73 11 déc. 16:31 etudiants.txt
-rw-r--r-- 1 root root  78 12 déc. 16:32 prenoms_tries
-rw-r--r-- 1 root root 128 12 déc. 16:25 sio1.txt
-rw-r--r-- 1 root root 128 12 déc. 16:18 sio.txt
root@DEB13Server: ~#
```

- On affiche la ligne du fichier notes.csv concernant l'étudiant Azer :

```
root@DEB13Server: ~#cat notes.csv | grep -i azer
Azer, SISR, 13, AB
root@DEB13Server: ~#
```

2. Utilisation des jokers.

- On se connecte en tant que guest. On se déplace dans le répertoire /bin à l'aide de la commande cd /bin.

- On affiche les commandes commençant par r du répertoire /bin.

```
guestt@DEB13Server:/bin$ ls r*
ranlib  rdma   readlink  rename.ul  reportbug  reset      resizepart  rgrep  rmdir  routel  rtstat  runcon  run-with-aspell
rash  readelf  realpath  renice    report-hw  resizecons  rev       rm     rnano  rpcgen  run0    run-parts  rview
guestt@DEB13Server:/bin$ _
```

- On affiche les commandes composées de 5 caractères.

```
guestt@DEB13Server:/bin$ ls ****?
b2sum  chage  chown  dmesg  flock  gzexe  ld.so  lsirq  lzcat  mount  nstat  pgrep  pkill  reset  sdiff  split  tsort  which  zdiff
bzcat  chcon  cksum  egrep  gmake  iconv  login  lsmem  lzcmp  namei  pager  pidof  pl2pm  rgrep  shred  strip  uname  xargs  zdump
bzcmp  chgrp  clear  eject  gorof  ijoin  lsblk  lsmod  mandb  nohup  partx  ping4  prove  rmdir  skill  tload  users  xauth  zgrep
bzexe  chmod  colrm  false  groff  ipcmk  lscpu  lspci  mkdir  nproc  paste  ping6  pzstd  rnano  sleep  touch  watch  xzcat  zless
bzip2  choom  diff3  fgrep  grops  ipcrm  lsipc  lsusb  mknod  nroff  patch  pinky  rbash  rview  snice  troff  wdctl  xzcmp  zmore
guestt@DEB13Server:/bin$ _
```

- On affiche les commandes de 2 caractères commençant par e.

```
guestt@DEB13Server:/bin$ ls e?
ex
guestt@DEB13Server:/bin$ _
```

- On affiche les commandes commençant par un w, un x, un y ou un z.

```
guestt@DEB13Server:/bin$ ls [wxyz]*
w          word-list-compress           x86_64-linux-gnu-gcc-ar          x86_64-linux-gnu-ld.bfd      xz          zdump
waitpid    wtmpdb                      x86_64-linux-gnu-gcc-ar-14        x86_64-linux-gnu-lto-dump    xzcat        zgrep
wall        x86_64                     x86_64-linux-gnu-gcc-nm          x86_64-linux-gnu-lto-dump-14 xzcmp        zfgrep
watch       x86_64-linux-gnu-addr2line  x86_64-linux-gnu-gcc-nm-14       x86_64-linux-gnu-nm          xzdiff        zforce
wc          x86_64-linux-gnu-ar         x86_64-linux-gnu-gcc-ranlib      x86_64-linux-gnu-objcopy    xzegrep      zgrep
wdctl       x86_64-linux-gnu-as        x86_64-linux-gnu-gcc-ranlib-14  x86_64-linux-gnu-objdump    xzfgrep      zipdetails
wget       x86_64-linux-gnu-c++filt    x86_64-linux-gnu-gcov            x86_64-linux-gnu-ranlib      xzgrep       zless
whatis      x86_64-linux-gnu-cpp       x86_64-linux-gnu-gcov-14        x86_64-linux-gnu-readelf    xzless       zmore
whereis     x86_64-linux-gnu-cpp-14    x86_64-linux-gnu-gcov-dump      x86_64-linux-gnu-size       xzmore       znew
which       x86_64-linux-gnu-elfedit   x86_64-linux-gnu-gcov-dump-14   x86_64-linux-gnu-strings    yes         zstd
which.debianutils x86_64-linux-gnu-g++  x86_64-linux-gnu-gcov-tool      x86_64-linux-gnu-strip      ydomainname zstdcat
whiptail    x86_64-linux-gnu-g++-14   x86_64-linux-gnu-gcov-tool-14   xargs                    zcat        zstdgrep
who         x86_64-linux-gnu-gcc       x86_64-linux-gnu-gprof          xauth                   zcmp        zstdless
whoami      x86_64-linux-gnu-gcc-14   x86_64-linux-gnu-ld            xsubpp                  zdiff        zstdmt
```

TP 10- Les commandes – Les principaux caractères du shell

- On affiche les commandes qui ne commencent pas par une lettre comprise entre a et v.

```
guestt@DEB13Server:/bin$ ls [!a-v]*  
[  
  which  
VBoxAudioTest  which.debianutils  
VBoxClient    whiptail  
VBoxClient-all who  
VBoxControl   whoami  
VBoxDRMClient word-list-compress  
w  wtmpdb  
waitpid  
wall  
watch  
wc  
wdctl  
wget  
whatis  
whereis  
x86_64-linux-gnu-addr2line  
x86_64-linux-gnu-ar  
x86_64-linux-gnu-as  
x86_64-linux-gnu-c++filt  
x86_64-linux-gnu-cpp  
x86_64-linux-gnu-cpp-14  
x86_64-linux-gnu-elfedit  
x86_64-linux-gnu-g++  
x86_64-linux-gnu-g++-14  
x86_64-linux-gnu-gcc  
x86_64-linux-gnu-gcc-14  
x86_64-linux-gnu-gcc-ar  
x86_64-linux-gnu-gcc-nm  
x86_64-linux-gnu-gcc-nm-14  
x86_64-linux-gnu-gcc-ranlib  
x86_64-linux-gnu-gcc-ranlib-14  
x86_64-linux-gnu-gcov  
x86_64-linux-gnu-gcov-14  
x86_64-linux-gnu-gcov-dump  
x86_64-linux-gnu-gcov-dump-14  
x86_64-linux-gnu-gcov-tool  
x86_64-linux-gnu-gcov-tool-14  
x86_64-linux-gnu-gprof  
x86_64-linux-gnu-ld  
x86_64-linux-gnu-ld.bfd  
x86_64-linux-gnu-lto-dump  
x86_64-linux-gnu-lto-dump-14  
x86_64-linux-gnu-nm  
x86_64-linux-gnu-objcopy  
x86_64-linux-gnu-objdump  
x86_64-linux-gnu-ranlib  
x86_64-linux-gnu-readelf  
x86_64-linux-gnu-size  
x86_64-linux-gnu-strings  
x86_64-linux-gnu-strip  
xargs  
xauth  
xsubpp  
xz  
zgrep  
zcat  
zdiff  
zdump  
zegrep  
zforce  
zgrep  
zdiff  
zgrep  
zipdetails  
zless  
zmore  
znew  
zless  
zmore  
zstd  
yes  
zstdcat  
zstdgrep  
zcat  
zcmp  
zstdless  
zstdmt  
guestt@DEB13Server:/bin$
```

Remarque : le caractère point d'exclamation dans le joker crochet signifie la négation.

3. Utilisation des caractères de protection.

- Le backslash « \ » demande au shell de ne pas interpréter le caractère suivant. Dans l'exemple, on essaye de créer un fichier commençant par un « # » qui normalement débute un commentaire.

```
guestt@DEB13Server:~$ touch #commentaire  
touch: opérande de fichier manquant  
Saisissez « touch --help » pour plus d'informations.  
guestt@DEB13Server:~$ touch \'#commentaire'  
guestt@DEB13Server:~$ ls -l *comm*
```

- Les simples quotes ('...') protègent un ensemble de caractères. Dans l'exemple, on crée un fichier dont le nom contient des espaces.

```
guestt@DEB13Server:~$ touch 'Fichier dont le nom contient des espaces'  
guestt@DEB13Server:~$ ls -l *espace*  
-rw-rw-r-- 1 guestt guestt 0 12 déc. 16:49 'Fichier dont le nom contient des espaces'  
guestt@DEB13Server:~$
```

- Les doubles quotes ("...") fonctionnent comme les simples quotes excepté que le caractère \$, qui référence les variables, est interprété.

```
guesstt@DEB13Server:~$ echo "Mon répertoire de connexion : $HOME"  
Mon répertoire de connexion : /home/guesstt  
guesstt@DEB13Server:~$ _
```

4. Autres caractères.

- Le point-virgule sépare les commandes.

```
guesstt@DEB13Server:~$ echo "voici la date :" ; date  
voici la date :  
ven. 12 déc. 2025 16:52:15 CET
```

- Les anti-quotes permettent d'interpréter une commande dans une commande. On génère ainsi dynamiquement des arguments.

```
guesstt@DEB13Server:~$ echo "voici la date : `date`"  
voici la date : ven. 12 déc. 2025 16:54:03 CET  
guesstt@DEB13Server:~$
```

- On saisisse les commandes echo date et echo `date`.

```
guesstt@DEB13Server:~$ echo date  
date  
guesstt@DEB13Server:~$
```

```
guesstt@DEB13Server:~$ echo `date`  
ven. 12 déc. 2025 16:55:34 CET  
guesstt@DEB13Server:~$ _
```