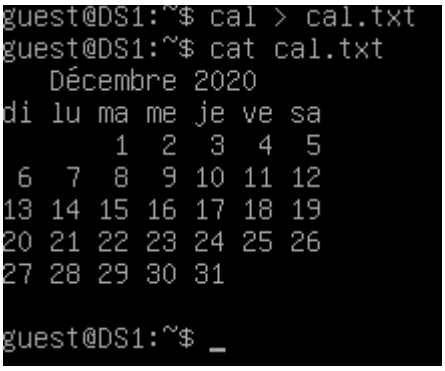
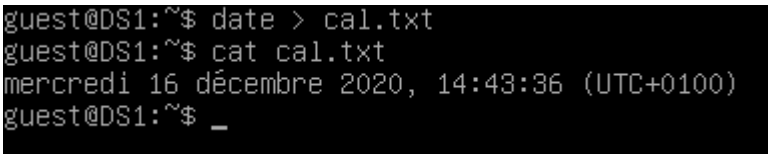
Conte-Rendu TP7 : Les principaux caractères du shell

1. **<u>Les redirections (>, >>, 2>, 2>>, 2>&1 et |)</u>**

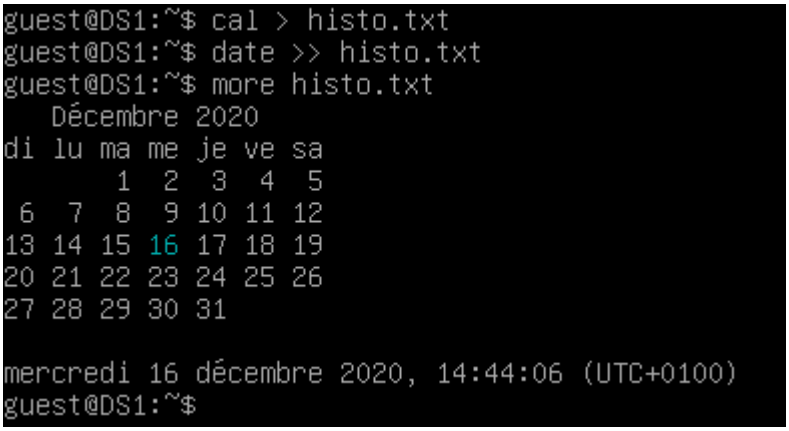
On se connecte en tant que guest. On redirige la commande cal dans le fichier cal.txt avec la commande **cal > cal.txt**. On l'affiche ensuite avec la commande **cat cal.txt**



On redirige la commande date dans le fichier cal.txt avec la commande **date > cal.txt**. On l'affiche ensuite avec la commande **cat cal.txt**, on constate que le contenu qui a été redirigé dans un premier lieu a été écrasé par le second.



On redirige la commande cal dans le fichier histo.txt avec la commande **cal > histo.txt**. Ensuite, on y ajoute le résultat de la commande date avec la commande **date >> histo.txt**. On affiche ensuite ce fichier page par page avec la commande **more histo.txt**. On constate que les contenus de cal et date s'affichent tous les deux.



On supprime le contenu de histo.txt, en faisant une redirection par la commande **> histo.txt**. De ce fait, on dirige un contenu vide vers le fichier, le contenu de histo.txt sera donc écrasé par ce contenu vide. On vérifie ensuite en affichant le fichier avec **cat histo.txt**.

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

On se connecte en tant que root. On va rediriger les valeurs du premiers champ du fichier notes.csv dans un nouveau fichié nommé eleves.txt, grâce à la commande **cut –d , -f 1 notes.csv > eleves.txt**. On liste ensuite le fichier avec **ls -l**, puis on l'affiche pour visualiser le contenu avec **cat eleves.txt**.

```
root@DS1: ~#cut -d , -f 1 notes.csv > eleves.txt
root@DS1: ~#ls -l
total 16
-rw-r--r-- 1 root root  73 déc.  16 14:46 eleves.txt
-rw-r--r-- 1 root root  73 déc.  16 14:30 etudiants.txt
-rw-r--r-- 1 root root 211 déc.  16 14:36 notes.csv
-rw-r--r-- 1 root root 211 déc.  16 14:35 prenom_tries
root@DS1: ~#cat eleves.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
root@DS1: ~#_
```

On va effectuer la même opération, mais cette fois-ci en gardant le contenu actuel de eleves.txt, pour cela on va effectuer la commande **cut –d , -f 1 notes.csv >> eleves.txt.** On affiche ensuite le fichier avec **cat eleves.txt**, on constate que la liste de noms est en double.

```
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
root@DS1: ~#
```

On va dans un premier temps trier le contenu de eleves.txt par ordre alphabétique, puis ce contenu sera redirigé vers le fichier avecdoublons, grâce à la commande **sort eleves.txt > avecdoublons**. On affiche ensuite ce nouveau fichier avec **cat avecdoublons**.

```
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
root@DS1: ~#_
```

On va utiliser la commande uniq pour supprimer tous les doublons, puis rediriger le contenu vers le fichier sansdoublons, avec la commande **uniq avecdoublons > sansdoublons**. Ensuite, on l'affiche avec **cat sansdoublons.**

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

On aurait pu également enregistrer le résultat de uniq avecdoublons directement dans pasdedoublons sans passer par une redirection, en utilisant la commande **uniq avecdoublons pasdedoublons.**

```
root@DS1: ~#uniq avecdoublons pasdedoublons
root@DS1: ~#_
```

On affiche ici, une sortie d'erreur en affichant un dossier inexistant que l'on a appelé fichier_inexistant, en utilisant la commande **cat fichier_inexistant**.

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

On redirige le fichier inexistant vers eleves.txt en utilisant la commande **cut –d , -f 1 fichier_inexistant.csv.** Le fichier étant inexistant, un message d'erreur apparait. On affiche ensuite le fichier avec **cat eleves.txt**, celui-ci ne contient plus de données.

```
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: ~#_
```

On redirige la sortie d'erreur dans un fichier pour appeler erreurs.log grâce à la commande **cut –d , -f 1 fichier_inexistant.csv > eleves.txt 2> erreurs.log**. On liste ensuite tous les fichiers avec **ls –l**, on voit que le fichier erreurs.log est bien présent. On l'affiche avec la commande **cat erreurs.log**, et il affiche le message d'erreur.

```
root@DS1: ~#cut -d, -f 1 fichier_inexistant.csv > eleves.txt 2> erreurs.log
root@DS1: ~#ls -l
total 28
-rw-r--r-- 1 root root 146 déc.  16 14:47 avecdoublons
-rw-r--r-- 1 root root   0 déc.  16 14:51 eleves.txt
-rw-r--r-- 1 root root  65 déc.  16 14:51 erreurs.log
-rw-r--r-- 1 root root  73 déc.  16 14:30 etudiants.txt
-rw-r--r-- 1 root root 211 déc.  16 14:36 notes.csv
-rw-r--r-- 1 root root  73 déc.  16 14:49 pasdedoublons
-rw-r--r-- 1 root root 211 déc.  16 14:35 prenom_tries
-rw-r--r-- 1 root root  73 déc.  16 14:48 sansdoublons
root@DS1: ~#cat erreurs.log
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type
root@DS1: ~#
```

On redirige la sortie d'erreur dans un fichier nommé sio1.txt avec la commande **cut –d , -f 1 ficher_inexistant.csv > sio1.txt 2>&1**. On l'affiche ensuite avec **cat sio1.txt**, le message d'erreur s'affiche.

```
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
root@DS1: ~#_
```

On redirige la sortie d'erreur à la fin du fichier sio1.txt avec la commande **cut –d , -f 1 fichier_inexistant.csv >> sio1.txt 2>&1**, ensuite on l'affiche. Il nous affiche le message d'erreurs en double.

```
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
root@DS1: ~#
```

On se connecte en tant que guest. On affiche, page par page, cal 2019 avec la commande **cal 2019 | more**.

```
                               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  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

guest@DS1:~$ _
```

On trie le fichier /etc/services et on affiche les 3 dernières lignes, avec la commande **sort /etc/services | 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 affiche le fichier /etc/services, trié, page par page, toutes les lignes qui ne commencent pas par “#”. Pour cela, on utilise la commande **grep –v ‘^#’ /etc/services | sort | more.**

```
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 se connecte en tant que root. On classe par ordre alphabétique la première colonne du ficher notes.csv avec la commande **cut –d , -f 1 notes.csv | sort**.

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

On enregistre cette liste dans un fichier en effectuant une redirection que l’on appelle prenom_tries.txt avec la commande **cut –d , -f 1 notes.csv | sort > prenom_tries**, puis on affiche ce dernier en utilisant **cat prenom_tries.txt**.

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

On affiche la liste des utilisateurs du système local, page par page, à l'aide de la commande **cat /etc/passwd | more.**

```
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
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:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
sio:x:1000:1000:sio,,,:/home/sio:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
guest:x:1001:1001::/home/guest:/bin/bash
```

On affiche la liste des GID à l'aide de la commande **cut –d : -f 4 /etc/passwd | sort –n | uniq**.

```
root@DS1: ~#cut -d : -f 4 /etc/passwd | sort -n | uniq
0
1
2
3
7
8
9
10
12
13
33
34
38
39
41
60
102
103
104
110
999
1000
1001
65534
root@DS1: ~#_
```

On affiche le nombre de lignes et de caractères contenus dans la liste des fichiers du répertoire personnel de root avec la commande **ls –l | wc**. Ensuite, on affiche les fichiers du répertoire avec **ls –l**.

```
root@DS1: ~#ls -l | wc
     11      92     566
root@DS1: ~#ls -l
total 36
-rw-r--r-- 1 root root 146 déc.  16 14:47 avecdoublons
-rw-r--r-- 1 root root   0 déc.  16 14:51 eleves.txt
-rw-r--r-- 1 root root  65 déc.  16 14:51 erreurs.log
-rw-r--r-- 1 root root  73 déc.  16 14:30 etudiants.txt
-rw-r--r-- 1 root root 211 déc.  16 14:36 notes.csv
-rw-r--r-- 1 root root  73 déc.  16 14:49 pasdedoublons
-rw-r--r-- 1 root root  73 déc.  16 15:03 prenoms_tries.txt
-rw-r--r-- 1 root root 211 déc.  16 14:35 prenom_tries
-rw-r--r-- 1 root root  73 déc.  16 14:48 sansdoublons
-rw-r--r-- 1 root root 130 déc.  16 14:53 sio1.txt
root@DS1: ~#_
```

On affiche les fichiers .txt qui sont contenus dans le répertoire personnel de root, avec la commande **ls –l | grep .txt**.

```
root@DS1: ~#ls -l | grep .txt
-rw-r--r-- 1 root root   0 déc.  16 14:51 eleves.txt
-rw-r--r-- 1 root root  73 déc.  16 14:30 etudiants.txt
-rw-r--r-- 1 root root  73 déc.  16 15:03 prenoms_tries.txt
-rw-r--r-- 1 root root 130 déc.  16 14:53 sio1.txt
root@DS1: ~#_
```

On affiche la ligne du fichier notes.csv qui concerne l'étudiant nommé Azer, avec la commande **cat notes.csv | grep –i azer**.

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

2. Utilisation des jokers.

On se connecte en tant que guest. On se déplace dans le répertoire bin avec la commande **cd /bin**. On affiche les commandes commençant par r dans le repertoire bin, en utilisant la commande **ls r***.

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

On affiche toutes les commandes qui sont composées de 5 caractères dans le répertoire bin avec la commande **ls ?????**.

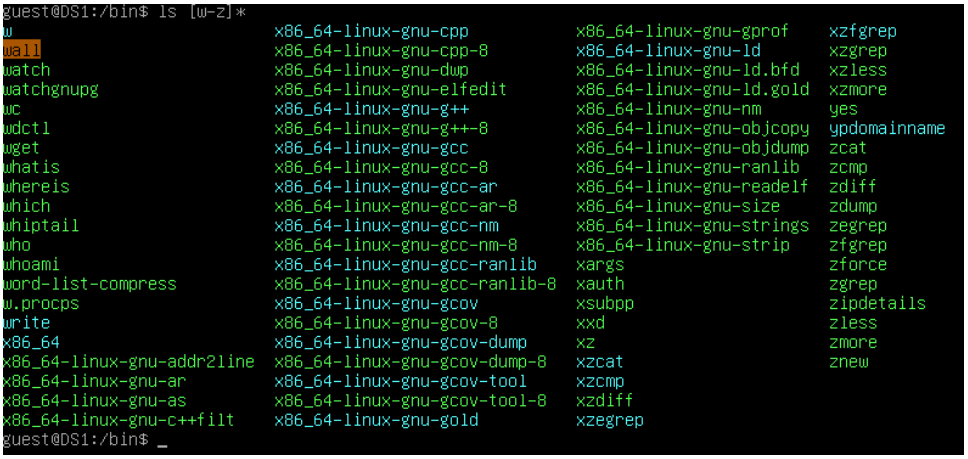
```
guest@DS1:/bin$ ls ?????
b2sum  chmod  dmesg  gpgsm  ipcrm  lspci  namei  patch  print  rview  strip  wdctl  zdump
bzcat  choom  egrep  gprof  lastb  lsusb  nohup  pgrep  prove  sdiff  tload  which  zgrep
bzcmp  chown  eject  groff  login  lzcat  nproc  pidof  pydoc  shred  touch  write  zless
bzexe  cksum  false  grops  lsblk  lzcmp  nroff  ping4  rbash  skill  troff  xargs  zmore
bzip2  clear  fgrep  gzexe  lscpu  mandb  nstat  ping6  reset  sleep  tsort  xauth
chage  colrm  flock  iconv  lsipc  mkdir  pager  pinky  rgrep  snice  uname  xzcat
chcon  cpp-8  g++-8  ijoin  lsmem  mknod  partx  pkill  rmdir  split  users  xzcmp
chgrp  diff3  gcc-8  ipcmk  lsmod  mount  paste  pl2pm  rnano  sprof  watch  zdiff
guest@DS1:/bin$
```

On affiche toutes les commandes qui sont composées de 2 caractères et qui commencent par e dans le répertoire bin, avec la commande **ls e?.**

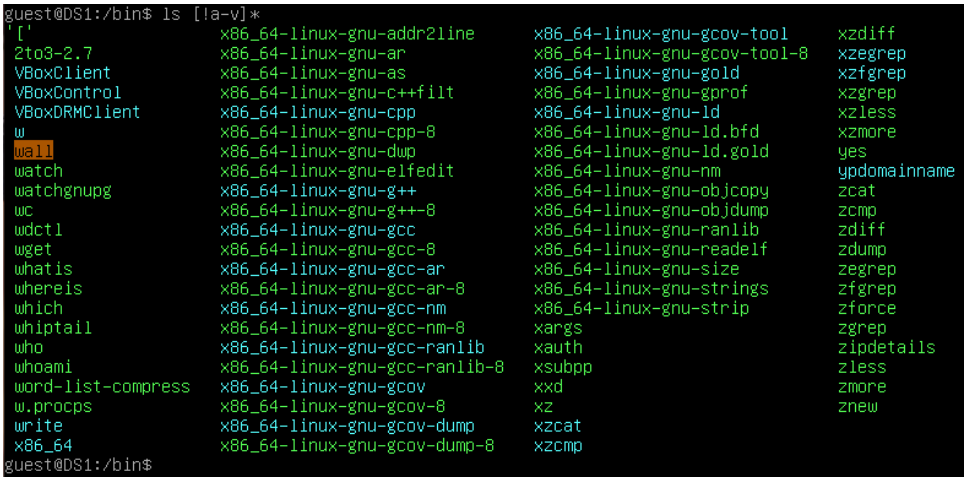
```
guest@DS1:/bin$ ls e?
ex
guest@DS1:/bin$
```

On affiche toutes les commandes qui commencent par un w, un x, un y ou un z dans le répertoire bin, avec la commande **ls [wxyz]*** ou **ls [w-z]***.

```
guest@DS1:/bin$ ls [wxyz]*
w                          x86_64-linux-gnu-cpp          x86_64-linux-gnu-gprof    xzfgrep
wall                       x86_64-linux-gnu-cpp-8        x86_64-linux-gnu-ld       xzgrep
watch                      x86_64-linux-gnu-dwp          x86_64-linux-gnu-ld.bfd   xzless
watchgnupg                 x86_64-linux-gnu-elfedit      x86_64-linux-gnu-ld.gold  xzmore
wc                         x86_64-linux-gnu-g++          x86_64-linux-gnu-nm       yes
wdctl                      x86_64-linux-gnu-g++-8        x86_64-linux-gnu-objcopy  ypdomainname
wget                       x86_64-linux-gnu-gcc          x86_64-linux-gnu-objdump  zcat
whatis                     x86_64-linux-gnu-gcc-8        x86_64-linux-gnu-ranlib   zcmp
whereis                    x86_64-linux-gnu-gcc-ar       x86_64-linux-gnu-readelf  zdiff
which                      x86_64-linux-gnu-gcc-ar-8     x86_64-linux-gnu-size     zdump
whiptail                   x86_64-linux-gnu-gcc-nm       x86_64-linux-gnu-strings  zegrep
who                        x86_64-linux-gnu-gcc-nm-8     x86_64-linux-gnu-strip    zfgrep
whoami                     x86_64-linux-gnu-gcc-ranlib   xargs                     zforce
word-list-compress         x86_64-linux-gnu-gcc-ranlib-8 xauth                     zgrep
w.procps                   x86_64-linux-gnu-gcov         xsubpp                    zipdetails
write                      x86_64-linux-gnu-gcov-8       xxd                       zless
x86_64                     x86_64-linux-gnu-gcov-dump    xz                        zmore
x86_64-linux-gnu-addr2line x86_64-linux-gnu-gcov-dump-8  xzcat                     znew
x86_64-linux-gnu-ar        x86_64-linux-gnu-gcov-tool    xzcmp
x86_64-linux-gnu-as        x86_64-linux-gnu-gcov-tool-8  xzdiff
x86_64-linux-gnu-c++filt   x86_64-linux-gnu-gold         xzegrep
guest@DS1:/bin$ _
```



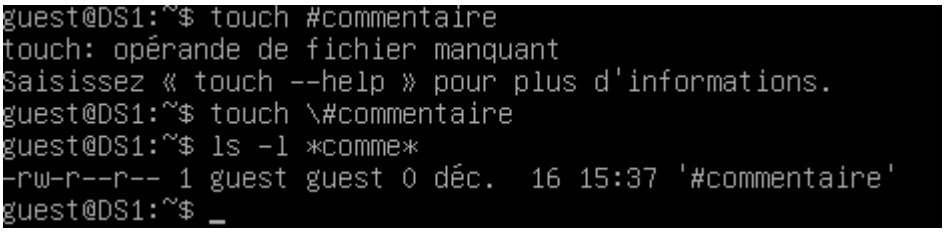
On affiche toutes les commandes qui ne commencent pas par une lettre comprise entre a et v dans le répertoire bin, avec la commande **ls [!a-v]***.



3. **<u>Utilisation des caractères de protection.</u>**

On quitte le répertoire bin, en utilisation la commande cd.

Le backslash “\” sert à demander au shell de ne pas interpréter le caractère suivant. Ici, lorsque l’on exécute la commande **touch #commentaire**, un message d’erreur s’affiche, car on ne peut pas créer un commentaire. Cependant, en faisant **touch \#commentaire**, la fonction du # est annulé, le fichier “#commentaire” est donc créé. On le vérifie en utilisant la commande **ls –l *comme***.



Les simples quotes ‘ ‘ servent à protéger un ensemble de caractères. Ici, on arrive à créer un fichier dont le nom est séparé par des espaces en utilisant la commande **touch ‘Fichier dont le nom contient des espaces’**, ce qui sans ça serait impossible. On vérifie, avec **ls –l *espace***.

```
guest@DS1:~$ touch 'Fichier dont le nom contient des espaces'
guest@DS1:~$ ls -l *espace*
-rw-r--r-- 1 guest guest 0 déc.  16 15:38 'Fichier dont le nom contient des espaces'
guest@DS1:~$
```

Les doubles quotes “ “ ont la même fonctionnalité que les simples quotes, sauf que le caractère $ garde sa fonctionnalité à l’intérieur. Il sert à référencer les variables. Ici, on demande un echo, pour savoir dans quel répertoire de connexion, nous nous situons. On utilise la commande **echo “Mon répertoire de connexion : $HOME”**.

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

4. **<u>Autres caractères</u>**

Le point-virgule, sert à séparer les commandes. On utilise ici la commande **echo “voici la date :” ; date**. La première commande, la demande d’écho est exécutée en première, puis la demande de la date est exécutée par la suite.

```
guest@DS1:~$ echo "voici la date :" ; date
voici la date :
mercredi 16 décembre 2020, 15:39:19 (UTC+0100)
guest@DS1:~$ _
```

Les anti-quotes ` ` permettent d’interpréter une commande dans une autre commande. Ici, on utilise la commande **echo “Voici la date : `date`”**. On a donc la commande echo qui est exécutée, ainsi que la commande date.

```
guest@DS1:~$ echo "Voici la date : `date`"
Voici la date : mercredi 16 décembre 2020, 15:39:51 (UTC+0100)
guest@DS1:~$ _
```