

TP:10

1. Redirigez le résultat de la commande cal dans un fichier. Visualisez le fichier ainsi créé.

```
root@DEB12Server: ~#cal
  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

root@DEB12Server: ~#
```

2. Redirigez le résultat de la commande date dans le même fichier. L'ancien contenu est perdu.

```
guest@DEB12Server:~$ date > cal.txt
guest@DEB12Server:~$ cat cal.txt
mar. 16 déc. 2025 08:38:41 CET
guest@DEB12Server:~$
```

3. Redirigez le résultat de la commande cal dans un fichier. Ajoutez-y le résultat de la commande date. Clear

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

mar. 16 déc. 2025 08:42:56 CET
guest@DEB12Server:~$ _
```

4. Supprimez le contenu du fichier par une redirection :

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

5. Redirigez les valeurs du premier champ du fichier notes.csv dans un nouveau fichier nommé eleves.txt. Constatez l'existence de ce dernier et visualisez son contenu :

TP:10

```
root@DEB12Server: ~# cut -d , -f 1 notes.csv > eleves.txt
root@DEB12Server: ~# ls -l
total 16
-rw-r--r-- 1 root root 73 16 déc. 08:51 eleves.txt
-rw-r--r-- 1 root root 75 11 déc. 16:21 etudiant.txt
-rw-r--r-- 1 root root 211 11 déc. 16:44 notes.csv
-rw-r--r-- 1 root root 75 11 déc. 16:30 prenom_tries
root@DEB12Server: ~#
```

6. Recommencez la même opération mais sans écraser le contenu du fichier eleves.txt. Affichez son contenu dans la console :

```
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
root@DEB12Server: ~#
```

7. Effectuez un tri sur le contenu du fichier eleves.txt et redirigez le résultat dans un nouveau fichier nommé avecdoublons.txt. Affichez son contenu dans la console :

TP:10

```
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
root@DEB12Server: ~#_
```

8. Supprimez les doublons avec la commande `uniq` et redirigez le résultat dans un nouveau fichier nommé `sansdoublons.txt`. Affichez son contenu dans la console :

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

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

TP:10

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

10. 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)

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

11. Redirigez l'éventuelle sortie d'erreurs dans un fichier à part pour en conserver une trace. Affichez le contenu du fichier erreurs.log :

```
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 16 déc. 09:05 avecdoublons
-rw-r--r-- 1 root root  0 16 déc. 09:17 eleves.txt
-rw-r--r-- 1 root root  73 16 déc. 09:01 eleve.txt
-rw-r--r-- 1 root root  64 16 déc. 09:17 erreurs.log
-rw-r--r-- 1 root root  75 11 déc. 16:21 etudiant.txt
-rw-r--r-- 1 root root 211 11 déc. 16:44 notes.csv
-rw-r--r-- 1 root root  75 11 déc. 16:30 prenom_tries
-rw-r--r-- 1 root root  73 16 déc. 09:11 sansdoublons
root@DEB12Server: ~#
```

12. Redirigez la sortie standard ou l'éventuelle sortie d'erreurs dans un seul et même fichier nommé sio1.txt. Affichez son contenu :

```
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 nom
root@DEB12Server: ~#
```

13. Redirigez la sortie standard ou l'éventuelle sortie d'erreurs à la fin du fichier sio1.txt. Affichez son contenu :

```
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 nom
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce nom
root@DEB12Server: ~#
```

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

TP:10

```
root@DEB12Server: ~# 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                1
 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                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@DEB12Server: ~#_
```

15. Affichez les 3 dernières lignes d'un fichier trié.

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

16. Affichez triées, page par page, les lignes d'un fichier qui ne commencent pas par « # ».

TP:10

```
tacacs      49/tcp      # Login Host Protocol (TACACS)
tacacs      49/udp
talk        517/udp
tcpmux      1/tcp      # TCP port service multiplexer
telnet      23/tcp
telnet      992/tcp    # Telnet over SSL
tfido       60177/tcp  # fidonet EMSI over telnet
tftp        69/udp
time        37/tcp     timserver
time        37/udp     timserver
tinc        655/tcp    # tinc control port
tinc        655/udp
tproxy      8081/tcp   # Transparent Proxy
uucp        540/tcp    uucpd      # uucp daemon
venus       2430/tcp   # codacon port
venus       2430/udp   # Venus callback/wbc interface
venus-se    2431/tcp   # tcp side effects
venus-se    2431/udp   # udp sftp side effect
webmin      10000/tcp
who         513/udp    whod
whois       43/tcp     nickname
winn6       22273/tcp  # winn6
x11-1       6001/tcp
x11-2       6002/tcp
x11-3       6003/tcp
x11-4       6004/tcp
x11-5       6005/tcp
x11         6000/tcp   x11-0      # X Window System
x11-6       6006/tcp
x11-7       6007/tcp
xdmcp       177/udp    # X Display Manager Control Protocol
xinetd      9098/tcp
xmms2       9667/tcp   # Cross-platform Music Multiplexing System
xmpp-client 5222/tcp   jabber-client # Jabber Client Connection
xmpp-server 5269/tcp   jabber-server # Jabber Server Connection
xtel        1313/tcp   # french minitel
xtelw       1314/tcp   # french minitel
z3950       210/tcp    wais        # NISO Z39.50 database
zabbix-agent 10050/tcp  # Zabbix Agent
zabbix-trapper 10051/tcp # Zabbix Trapper
zebra       2601/tcp   # zebra vty
zebrasrv    2600/tcp   # zebra service
zephyr-clt  2103/udp   # Zephyr serv-hm connection
zephyr-hm   2104/udp   # Zephyr hostmanager
zephyr-srv  2102/udp   # Zephyr server
zip         6/ddp     # Zone Information Protocol
zope        9673/tcp   # zope server
zope-ftp    8021/tcp   # zope management by ftp
zserv       346/tcp    # Zebra server
```

17. Classez par ordre alphabétique les prénoms figurant dans le premier champ du fichier notes.csv :

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

TP:10

18. Enregistrez cette liste dans un fichier nommé `prenoms_tries.txt` :

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

19. Affichez 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@DEB12Server: ~# cat/etc/passwd | more
-bash: cat/etc/passwd: Aucun fichier ou dossier de ce nom
root@DEB12Server: ~#_
```

20. Affichez 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` :

21. Affichez le nombre de lignes et de caractères contenus dans la liste des fichiers du répertoire personnel de root :

TP:10

```
root@DEB12Server: ~# ls -l | wc
    11     92     562
root@DEB12Server: ~# ls -l
total 36
-rw-r--r--  1 root root 146 16 déc.  09:05 avecdoublons
-rw-r--r--  1 root root   0 16 déc.  09:17 eleves.txt
-rw-r--r--  1 root root  73 16 déc.  09:01 eleve.txt
-rw-r--r--  1 root root  64 16 déc.  09:17 erreurs.log
-rw-r--r--  1 root root  75 11 déc.  16:21 etudiant.txt
-rw-r--r--  1 root root 211 11 déc.  16:44 notes.csv
-rw-r--r--  1 root root  75 11 déc.  16:30 prenoms_tries
-rw-r--r--  1 root root  73 16 déc.  09:40 prenoms_tries.txt
-rw-r--r--  1 root root  73 16 déc.  09:11 sansdoublons
-rw-r--r--  1 root root 128 16 déc.  09:27 sio1.txt
root@DEB12Server: ~#
```

22. Affichez les fichiers .txt contenus dans le répertoire personnel de root

23. Affichez la ligne du fichier notes.csv concernant l'étudiant Azer :

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

2. Utilisation des jokers.

1. Connectez-vous en tant que guest. Déplacez-vous dans le répertoire /bin à l'aide de la commande cd /bin.

2. Affichez les commandes commençant par r du répertoire /bin

```
guest@DEB12Server:/$ cd /bin
guest@DEB12Server:/bin$ ls r*
rbash readlink rename.ul reportbug reset resizepart rgrep rmdir routel run0 run-parts rview
rdma realpath renice report-hw resizecons rev rm rnano rstat runcon run-with-aspell
guest@DEB12Server:/bin$ _
```

3. Affichez les commandes composées de 5 caractères.

```
guest@DEB12Server:/bin$ ls ?????
b2sum bzip2 chmod clear egrep flock iconv ld.so lsipc lspci mandb namei nstat pgrep pinky pzstd rmdir
bzip2 chage chown colrm eject groff ijoin login lsirq lsusb mkdir nohup pager pidof pkill rbash rnano
bzcmp chcon chown diff3 false grops ipcmk lsblk lsmem lzcat mknod nproc partx ping4 pl2pm reset rview
bzexe chgrp cksum dmesg fgrep gzexe ipcrm lscpu lsmod lzcmp mount nroff paste ping6 prove rgrep sdiff
guest@DEB12Server:/bin$
```

4. Affichez les commandes de 2 caractères commençant par e

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

TP:10

5. Affichez les commandes commençant par un w, un x, un y ou un z.

```
guest@DEB12Server:/bin$ ls [wxyz]*
w      wc      whereis  who      x86_64  xz      xzegrep  xzmore   zcmp     zfgrep
waitpid wdctl   which    whoami   xargs   xzcat   xzfgrep  yes      zdiff   zforce
wall   wget    which.debianutils word-list-compress xauth   xzcmp   xzgrep   ypdomainname zdump   zgrep
watch  whatis  whiptail wtmpdb   xsubpp  xzdiff  xzless   zcat     zegrep  zipdetails
guest@DEB12Server:/bin$ ls [w-z]*
w      wc      whereis  who      x86_64  xz      xzegrep  xzmore   zcmp     zfgrep
waitpid wdctl   which    whoami   xargs   xzcat   xzfgrep  yes      zdiff   zforce
wall   wget    which.debianutils word-list-compress xauth   xzcmp   xzgrep   ypdomainname zdump   zgrep
watch  whatis  whiptail wtmpdb   xsubpp  xzdiff  xzless   zcat     zegrep  zipdetails
guest@DEB12Server:/bin$ _
```

6. Affichez les commandes qui ne commencent pas par une lettre comprise entre a et v.

```
guest@DEB12Server:/bin$ ls [!a-v]*
['
VBoxAudioTest  VBoxDRMClient  wc      which    word-list-compress  xsubpp  xzegrep  yes
VBoxClient      w              wdctl   which.debianutils  wtmpdb   xz      xzfgrep  ypdomainname
VBoxClient      waitpid        wget    whiptail  x86_64   xzcat   xzgrep   zcat
VBoxClient-all wall           whatis  who        xargs    xzcmp   xzless   zcmp
VBoxControl     watch          whereis whoami     xauth    xzdiff  xzmore   zdiff
guest@DEB12Server:/bin$ _
```

3. Utilisation des caractères de protection.

1. 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.

```
guest@DEB12Server:~$ touch #commentaire
touch: opérande de fichier manquant
Saisissez « touch --help » pour plus d'informations.
guest@DEB12Server:~$ touch \#commentaire
guest@DEB12Server:~$ ls -l *comme*
-rw-rw-r-- 1 guest guest 0 16 déc. 10:17 '#commentaire'
guest@DEB12Server:~$ _
```

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

```
guest@DEB12Server:~$ touch 'ficjier dont le nom contient des espaces'
guest@DEB12Server:~$ ls -l *espace*
-rw-rw-r-- 1 guest guest 0 16 déc. 10:21 'ficjier dont le nom contient des espaces'
guest@DEB12Server:~$
```

TP:10

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

```
guest@DEB12Server:~$ echo "Mon répertoire de connexion : $HOME"
Mon répertoire de connexion : /home/guest
guest@DEB12Server:~$
```

4. Autres caractères.

1. Le point-virgule sépare les commandes.

```
guest@DEB12Server:~$ echo "voici la date : " ; date
voici la date :
mar. 16 déc. 2025 10:24:59 CET
guest@DEB12Server:~$ _
```

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

```
guest@DEB12Server:~$ echo "voici la date : 'date'"
voici la date : 'date'
guest@DEB12Server:~$
```