

TP10 : Les commandes – Les principaux caractères du shell

Sommaire

1. Les redirections (>, >>, 2>, 2>>, 2>&1 et).....	2
2. Utilisation des jokers.....	5
3. Utilisation des caractères de protection.	9
4. Autres caractères.....	10

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

- Premièrement je rédige le résultat de la commande cal dans un fichier et je visualise le fichier ainsi créé :

```
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
```

- Ensuite je redirige 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
ven. 13 déc. 2024 16:05:18 CET
```

- Et aussi je rédige le résultat de la commande cal dans un fichier. Ajoutez-y le résultat de la commande date.

```
  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. 13 déc. 2024 16:06:04 CET
```

- Après je supprime le contenu du fichier par une redirection :

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

- Je redirige les valeurs du premier champ du fichier notes.csv dans un nouveau fichier nommé eleves.txt. Puis je constate l'existence de ce dernier et visualisez son contenu :

```
root@DEB12Server: ~#cut -d , -f 1 notes.csv > eleves.txt
root@DEB12Server: ~#ls -l
total 20
-rw-r--r-- 1 root root 73 13 déc. 16:11 eleves.txt
-rw-r--r-- 1 root root 73 13 déc. 15:43 etudiants.txt
drwxr-xr-x 2 root root 4096 6 déc. 16:31 liens
-rw-r--r-- 1 root root 210 13 déc. 15:55 notes.csv
-rw-r--r-- 1 root root 210 13 déc. 15:52 prenom_tries
root@DEB12Server: ~#cat eleves.txt
Antoine
Azer
Cedric
David
Denis
Fabien
Nicolas
Souphiane
Tiphaine
Xavier
```

- Je réitère la même opération mais sans écraser le contenu du fichier eleves.txt. et j'affiche 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
```

- Je fais un tri sur le contenu du fichier `eleves.txt` et redirige le résultat dans un nouveau fichier nommé `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
```

- Je Supprime les doublons avec la commande `uniq` et redirige le résultat dans un nouveau fichier nommé `sansdoublons.txt` puis j'affiche son contenu dans la console :

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

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@DEB12Server: ~#uniq avecdoublons pasdedoublons
root@DEB12Server: ~#
```

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

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

- Si l'on désire rediriger la sortie standard dans un fichier, le message d'erreur continue de s'afficher dans la console :

```
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
```

- Je redirige 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> erreur.log
root@DEB12Server: ~#ls -l
total 36
-rw-r--r-- 1 root root 146 13 déc. 16:15 avecdoublons
-rw-r--r-- 1 root root 0 13 déc. 16:21 eleves.txt
-rw-r--r-- 1 root root 146 13 déc. 16:13 eleve.txt
-rw-r--r-- 1 root root 65 13 déc. 16:21 erreur.log
-rw-r--r-- 1 root root 73 13 déc. 15:43 etudiants.txt
drwxr-xr-x 2 root root 4096 6 déc. 16:31 liens
-rw-r--r-- 1 root root 210 13 déc. 15:55 notes.csv
-rw-r--r-- 1 root root 73 13 déc. 16:17 pasdedoublons
-rw-r--r-- 1 root root 210 13 déc. 15:52 prenom_tries
-rw-r--r-- 1 root root 73 13 déc. 16:16 sansdoublons
```

- Je redirige la sortie standard ou l'éventuelle sortie d'erreurs dans un seul et même fichier nommé sio1.txt et j'affiche 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 type
root@DEB12Server: ~#
```

- Je redirige la sortie standard ou l'éventuelle sortie d'erreurs à la fin du fichier sio1.txt et j'affiche 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 type
cut: fichier_inexistant.csv: Aucun fichier ou dossier de ce type
```

- J'affiche le résultat d'une commande page par page : cal 2024 | more et touche espace (chainage de 2 commandes à l'aide du caractère « pipe »).

- Je classe 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
```

- Et j'enregistre cette liste dans un fichier nommé prenom_tries.txt :

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

- J'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: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
l1st:x:38:38:Mailng List Manager:/var/l1st:/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
systemd-timesync:x:997:997:systemd Time Synchronization:./usr/sbin/nologin
messagebus:x:100:107:./nonexistent:/usr/sbin/nologin
maxence:x:1000:1000:Maxence,,,:/home/maxence:/bin/bash
vboxadd:x:996:1:./var/run/vboxadd:/bin/false
sshd:x:101:65534:./run/sshd:/usr/sbin/nologin
guest:x:1001:1001:guest,,,:/home/guest:/bin/bash
```

- J'affiche ensuite 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 | uni

```
root@DEB12Server: ~#cut -d : -f 4 /etc/passwd | sort -n | uni
0
1
2
3
```


- J'affiche les commandes commençant par un w, un x, un y ou un z :

```

guest@DEB12Server:/bin$ ls [wxyz]*
w      x86_64
wall  x86_64-linux-gnu-addr2line x86_64-linux-gnu-gcc-nm-12 x86_64-linux-gnu-gprofng xsubpp zegrep
watch x86_64-linux-gnu-ar x86_64-linux-gnu-gcc-ranlib x86_64-linux-gnu-ld xz zfgrep
watchngup x86_64-linux-gnu-as x86_64-linux-gnu-gcov x86_64-linux-gnu-ld.bfd xzcat zforce
wc      x86_64-linux-gnu-c++filt x86_64-linux-gnu-gcov-12 x86_64-linux-gnu-ld.gold xzcmp zgrep
wdctl  x86_64-linux-gnu-cpp x86_64-linux-gnu-gcov-dump x86_64-linux-gnu-lto-dump xzdiff zipdetails
wget   x86_64-linux-gnu-cpp-12 x86_64-linux-gnu-gcov-dump-12 x86_64-linux-gnu-lto-dump-12 xzegrep zless
whatis x86_64-linux-gnu-dwp x86_64-linux-gnu-gcov-tool x86_64-linux-gnu-nm xzfgrep zmore
whereis x86_64-linux-gnu-elfedit x86_64-linux-gnu-gcov-tool-12 x86_64-linux-gnu-objcopy xzgrep znew
which  x86_64-linux-gnu-g++ x86_64-linux-gnu-gold x86_64-linux-gnu-objdump xzless zstd
which.debianutils x86_64-linux-gnu-g++-12 x86_64-linux-gnu-gold x86_64-linux-gnu-ranlib xzmore zstdcat
whiptail x86_64-linux-gnu-gcc x86_64-linux-gnu-gp-archive x86_64-linux-gnu-readelf yes zstdgrep
who     x86_64-linux-gnu-gcc x86_64-linux-gnu-gp-collect-app x86_64-linux-gnu-size updomainname zstdless
whoami  x86_64-linux-gnu-gcc-12 x86_64-linux-gnu-gp-display-html x86_64-linux-gnu-strings zcat zstdmt
word-list-compress x86_64-linux-gnu-gcc-ar x86_64-linux-gnu-gp-display-src x86_64-linux-gnu-strip zcmp
write  x86_64-linux-gnu-gcc-ar-12 x86_64-linux-gnu-gp-display-text xargs zdiff
       x86_64-linux-gnu-gcc-nm x86_64-linux-gnu-gprof xauth zdump

```

- Puis les commandes qui ne commencent pas par une lettre comprise entre a et v :

```

guest@DEB12Server:/bin$ ls [!a-v]*
if      who x86_64-linux-gnu-gcc-ar x86_64-linux-gnu-gp-display-text xauth zegrep
VBoxAudioTest whoami x86_64-linux-gnu-gcc-ar-12 x86_64-linux-gnu-gprof xsubpp zfgrep
VBoxClient word-list-compress x86_64-linux-gnu-gcc-nm x86_64-linux-gnu-gprofng xz zforce
VBoxControl write x86_64-linux-gnu-gcc-nm-12 x86_64-linux-gnu-ld xzcat zgrep
VBoxDRMClient x86_64 x86_64-linux-gnu-gcc-ranlib x86_64-linux-gnu-ld.bfd xzcmp zipdetails
w      x86_64-linux-gnu-addr2line x86_64-linux-gnu-gcov x86_64-linux-gnu-ld.gold xzdiff zless
wall  x86_64-linux-gnu-ar x86_64-linux-gnu-gcov-12 x86_64-linux-gnu-lto-dump xzegrep zmore
watch x86_64-linux-gnu-as x86_64-linux-gnu-gcov-dump x86_64-linux-gnu-lto-dump-12 xzfgrep znew
watchngup x86_64-linux-gnu-c++filt x86_64-linux-gnu-gcov-dump-12 x86_64-linux-gnu-nm xzgrep zstd
wc      x86_64-linux-gnu-cpp x86_64-linux-gnu-gcov-tool x86_64-linux-gnu-objcopy xzless zstdcat
wdctl  x86_64-linux-gnu-cpp-12 x86_64-linux-gnu-gcov-tool-12 x86_64-linux-gnu-objdump xzmore zstdgrep
wget   x86_64-linux-gnu-dwp x86_64-linux-gnu-gcov-tool-12 x86_64-linux-gnu-ranlib xzless zstdless
whatis x86_64-linux-gnu-elfedit x86_64-linux-gnu-gold x86_64-linux-gnu-readelf yes zstdmt
whereis x86_64-linux-gnu-g++ x86_64-linux-gnu-gp-archive x86_64-linux-gnu-size updomainname
which  x86_64-linux-gnu-g++-12 x86_64-linux-gnu-gp-collect-app x86_64-linux-gnu-strings zcat
which.debianutils x86_64-linux-gnu-gcc x86_64-linux-gnu-gp-display-html x86_64-linux-gnu-strip zdiff
whiptail x86_64-linux-gnu-gcc-12 x86_64-linux-gnu-gp-display-src xargs zdump

```

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.

```

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 *comm*
-rw-r--r-- 1 guest guest 0 13 d c. 16:49 '#commentaire'

```

- Les simples quotes ('...') prot gent un ensemble de caract res. Dans l'exemple, on cr e un fichier dont le nom contient des espaces.

```

guest@DEB12Server:~$ touch 'Fichier dont le nom contient des espaces'
guest@DEB12Server:~$ ls -l *espace*
-rw-r--r-- 1 guest guest 0 13 d c. 16:50 'Fichier dont le nom contient des espaces'
guest@DEB12Server:~$

```

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

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

4. Autres caractères.

- Le point-virgule sépare les commandes.

```
guest@DEB12Server:~$ echo "Mon répertoire de connexion : $HOME"
Mon répertoire de connexion : /home/guest
guest@DEB12Server:~$ echo "voici la date :"; date
voici la date :
ven. 13 déc. 2024 16:52:19 CET
guest@DEB12Server:~$
```

- 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 : sam. 14 déc. 2024 01:09:24 CET
```

- Et pour finir je saisi les commandes echo date et echo `date` :

```
guest@DEB12Server:~$ echo date
date
guest@DEB12Server:~$ echo `date`
sam. 14 déc. 2024 01:10:10 CET
guest@DEB12Server:~$
```