

COMPTE RENDU TP1 AUTOMATISATION DES TACHES :

Rappels : commandes de base UNIX et Shell Bash

INTRODUCTION :

Ce TP avait pour objectif de revisiter les commandes de base d'UNIX et les bases du shell Bash, déjà étudiées au premier semestre. Les participants ont travaillé avec la version Linux Debian, nommée "bosc" en salle de TP. L'accent a été mis sur la révision des commandes essentielles, l'organisation des fichiers, les redirections, l'utilisation de variables d'environnement, et d'autres concepts fondamentaux.

Exercice1 : Commande de base

Réponse 01 :

Dans cette section, nous avons revisité des commandes telles que **cd**, **mkdir**, **ls**, **rm**, **cp**, etc. Chaque commande a été expliquée en détail avec une phrase décrivant son utilité, ainsi que le rôle des options courantes.

- **cd** : choose directory
- **mkdir** : créer un répertoire (make directory)
 - p : créer un répertoire parent
- **rmdir** : supprimer un répertoire (remove directory)
- **pwd** : affiche le répertoire de travail
- **man** : manuel d'une commande
- **ls** : liste le contenu du répertoire de travail
 - l : affiche plus de détails
 - a : affiche également les fichiers cachés (commençant par un.)
 - R : affiche les sous répertoires (récursive)
 - 1 : affiche un fichier sur une ligne
- **rm** : remove file, supprimer un fichier
 - i : demander la confirmation de la suppression
 - r : supprimer les sous répertoire également
- **cp** : copier un fichier
 - i : demander la confirmation pour écraser un fichier existant par exemple
 - r : copier le répertoire et sa sous arborescence
 - a : archive
- **mv** : move, déplacer un fichier
 - i : demander la confirmation
- **date** : affiche la date
- **echo** : affiche une ligne de texte
 - n : ne pas effectuer le saut de ligne final

Organisation des fichiers :

Une recommandation a été faite pour une organisation efficace des fichiers, encourageant l'utilisation de sous-répertoires pour chaque module et TP afin d'améliorer l'efficacité à long terme.

Réponse 02 :

J'ai créé un répertoire tp-auto qui sera mon répertoire principal pour cette module. Dedans j'y ai créé un sous répertoire TP01 pour le premier tp.

```
astou@DESKTOP-86U4SBR:~/tp-auto$ mkdir TP01
astou@DESKTOP-86U4SBR:~/tp-auto$ ls
TP01
```

Réponse 03 :

L'utilisation de la commande `man` a été soulignée, avec une démonstration de recherche interactive dans le manuel de `less`.

```

astou@DESKTOP-86U4SBR:~/tp-auto$ man less | grep -i "pattern"
[-o0] logfile] [-p pattern] [-P prompt] [-t tag]
    last search pattern, the terminal bell is rung and forward
    /pattern
        pattern. N defaults to 1. The pattern is a regular expres-
        of the pattern; they modify the type of search rather than
        become part of the pattern:
            Search for lines which do NOT match the pattern.
        ^K    Highlight any text which matches the pattern on the
    ?pattern
        pattern. The search starts at the last line displayed (but
        Search for lines which do NOT match the pattern.
    ESC-/pattern
    ESC-?pattern
        pattern. If the previous search was modified by ^N, the
        search is made for the N-th line NOT containing the pattern.
        matching the current search pattern. If highlighting is al-
    ESC-U Like ESC-u but also clears the saved search pattern. If the
    &pattern
        Display only lines which match the pattern; lines which do
        not match the pattern are not displayed. If pattern is
        which case only lines which match all of the patterns will
        Display only lines which do NOT match the pattern.
troff: <standard input>:657: warning [p 10, 8.0i]: can't break line
troff: <standard input>:668: warning [p 10, 8.8i]: cannot adjust line
        any uppercase letters appear in the search pattern; in other
        words, if a pattern contains uppercase letters, then that
        Like -i, but searches ignore case even if the pattern con-
        command to search for a pattern, jump to a line number, jump
    -ppattern or --pattern=pattern
        ing +/pattern; that is, it tells less to start at the first
        occurrence of pattern in the file.
troff: <standard input>:1157: warning [p 17, 5.2i]: cannot adjust line
        pattern as each character of the pattern is typed in.
    ple, a filename for the :e command, or the pattern for a search
    than a search pattern.

```

Réponse 04 :

Nous avons exploré diverses options de la commande **ls** pour afficher le contenu du répertoire courant de différentes manières, notamment en listant les fichiers cachés, en affichant des informations détaillées, en utilisant un format plus lisible, et en effectuant des listes récursives et chronologiques.

- **ls** : Affiche la liste des fichiers et répertoires.

```

astou@DESKTOP-86U4SBR:~$ ls
tp-auto

```

- **ls -a** : Affiche la liste des fichiers et répertoires.

```

astou@DESKTOP-86U4SBR:~$ ls -a
.  .bash_logout  .lessht  .sudo_as_admin_successful
.. .bashrc      .profile  tp-auto

```

- `ls -l` : Affiche une liste détaillée avec des informations telles que les droits, le nombre de liens, la taille, etc.

```
astou@DESKTOP-86U4SBR:~$ ls -l
total 4
drwxr-xr-x 3 astou astou 4096 Jan 24 14:45 tp-auto
astou@DESKTOP-86U4SBR:~$ |
```

- `ls -lh` : Affiche une liste détaillée avec une taille de fichier lisible.

```
astou@DESKTOP-86U4SBR:~$ ls -lh
total 4.0K
drwxr-xr-x 3 astou astou 4.0K Jan 24 14:45 tp-auto
astou@DESKTOP-86U4SBR:~$ |
```

- `ls -R` : Affiche la liste de manière récursive, descendant dans les sous-répertoires.

```
astou@DESKTOP-86U4SBR:~$ ls -R
.:
tp-auto

./tp-auto:
TP01

./tp-auto/TP01:
astou@DESKTOP-86U4SBR:~$ |
```

- `ls -lt` : Affiche la liste par date de modification.

```
astou@DESKTOP-86U4SBR:~$ ls -lt
total 4
-rw-r--r-- 1 astou astou 0 Jan 24 15:20 azer
drwxr-xr-x 3 astou astou 4096 Jan 24 14:45 tp-auto
astou@DESKTOP-86U4SBR:~$ |
```

- `ls -lU` : Affiche la liste par date d'accès.

```
astou@DESKTOP-86U4SBR:~$ ls -lU
total 4
drwxr-xr-x 3 astou astou 4096 Jan 24 14:45 tp-auto
-rw-r--r-- 1 astou astou 0 Jan 24 15:20 azer
-rw-r--r-- 1 astou astou 0 Jan 24 15:35 trez
astou@DESKTOP-86U4SBR:~$
```

- `ls -F` : Affiche la liste avec des indications sur le type de fichier (répertoire, lien symbolique, exécutable).

```
astou@DESKTOP-86U4SBR:~$ ls -F
azer tp-auto/ trez
astou@DESKTOP-86U4SBR:~$ |
```

- **ln -i** : Affiche la liste avec le numéro inode associé à chaque fichier.

```
astou@DESKTOP-86U4SBR:~$ ls -i
10557 azer 9755 test 358 tp-auto 9755 trez
astou@DESKTOP-86U4SBR:~$ ln trez tut
astou@DESKTOP-86U4SBR:~$ ls -i
10557 azer 9755 test 358 tp-auto 9755 trez 9755 tut
astou@DESKTOP-86U4SBR:~$ |
```

Exercice 2 : Redirections

Cette section a couvert l'utilisation des redirections avec la commande **echo** pour créer et modifier des fichiers, ainsi que l'utilisation des commandes **wc** pour compter le nombre de mots et de lignes.

Réponse1 : Crée un fichier "dd" contenant le texte "Bonjour".

```
astou@DESKTOP-86U4SBR:~$ echo Bonjour >dd
astou@DESKTOP-86U4SBR:~$ cat dd
Bonjour
```

Réponse2 : Ajoute le texte "Hello" à la fin du fichier "dd".

```
astou@DESKTOP-86U4SBR:~$ echo Hello >>dd
astou@DESKTOP-86U4SBR:~$ cat dd
Bonjour
Bonjour
Hello
astou@DESKTOP-86U4SBR:~$ |
```

Réponse3 :

La commande **wc** permet de compter les mots, les caractères et les lignes des fichiers. Et l'option **-l** permet de compter uniquement les lignes au sein d'un fichier.

Réponse4 : Liste les fichiers de /etc et redirige le nombre total de fichiers dans le fichier "/tmp/astou.txt".

```
astou@DESKTOP-86U4SBR:~$ ls /etc | wc /tmp/astou.txt
110 110 1014 /tmp/astou.txt
astou@DESKTOP-86U4SBR:~$ wc /tmp/astou.txt
110 110 1014 /tmp/astou.txt
```

Réponse5 : Utilise un tube pour rediriger le nombre total de fichiers de /etc dans le fichier "/tmp/astou.txt".

```
astou@DESKTOP-86U4SBR:~$ ls /etc | wc
    110      110     1014
astou@DESKTOP-86U4SBR:~$ |
```

Exercice3 : Trouvez les informations suivantes et notez les sur votre compte-rendu

J'ai appris à trouver des informations telles que l'adresse IP, le nom de la machine, la mémoire vive installée, et l'espace disque disponible.

Réponse1 :

- Commande : `ip -a`
- Résultat : Affiche l'adresse IP de la machine.

```
astou@DESKTOP-86U4SBR:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:c8:f6:cb brd ff:ff:ff:ff:ff:ff
    inet 172.19.40.112/20 brd 172.19.47.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fec8:f6cb/64 scope link
        valid_lft forever preferred_lft forever
astou@DESKTOP-86U4SBR:~$ |
```

Activer Wind
Accédez aux para

Réponse2 :

- Fichier : `/etc/resolv.conf`
- Résultat : Affiche les informations de configuration DNS, y compris l'adresse IP du DNS principal.

```
astou@DESKTOP-86U4SBR:~$ cat /etc/resolv.conf | grep -i nameserver
nameserver 172.19.32.1
astou@DESKTOP-86U4SBR:~$ |
```

Réponse3 :

- Commande : `uname`
- Résultat : Affiche le nom de la machine.

```
astou@DESKTOP-86U4SBR:~$ uname
Linux
astou@DESKTOP-86U4SBR:~$ |
```

Réponse4 :

- Commande : `free`

- Résultat : Affiche la mémoire vive totale et utilisée.

```
astou@DESKTOP-86U4SBR:~$ free
              total        used        free      shared  buff/cache   available
Mem:           5763052       523792       5165096        2504       297624       5239260
Swap:          2097152           0       2097152
```

Réponse5 :

- Commande : pwd
- Résultat : Affiche le répertoire de connexion de l'utilisateur

```
astou@DESKTOP-86U4SBR:~$ pwd
/home/astou
astou@DESKTOP-86U4SBR:~$ |
```

Réponse5 :

- Commande : df
- Résultat : Affiche l'espace disque disponible sur le répertoire de connexion.

```
astou@DESKTOP-86U4SBR:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
none            2881524         4    2881520   1% /mnt/wsl
none           124045308 107536596  16508712  87% /usr/lib/wsl/drivers
/dev/sdc        1055762868    370616 1001688780   1% /
none            2881524        96    2881428   1% /mnt/wslg
none            2881524         0    2881524   0% /usr/lib/wsl/lib
rootfs          2878228        2056    2876172   1% /init
none            2878228         0    2878228   0% /dev
none            2881524         4    2881520   1% /run
none            2881524         0    2881524   0% /run/lock
none            2881524         0    2881524   0% /run/shm
none            2881524         0    2881524   0% /run/user
tmpfs           2881524         0    2881524   0% /sys/fs/cgroup
none            2881524         76    2881448   1% /mnt/wslg/versions.txt
none            2881524         76    2881448   1% /mnt/wslg/doc
drvfs           124045308 107536596  16508712  87% /mnt/c
drvfs           976629756  40299508  936330248   5% /mnt/d
drvfs           15728640   4723016  11005624  31% /mnt/g
astou@DESKTOP-86U4SBR:~$ |
```

Exercice4 : Commande find

Afficher le nombre de fichiers dans un répertoire et ses sous-répertoires:

Réponse1 :

```
astou@DESKTOP-86U4SBR:~$ find /usr -size +250k|
```

Réponse2 :

```
astou@DESKTOP-86U4SBR:~$ find /var -newer ~ -type f -exec basename {} \;
find: '/var/spool/cron/crontabs': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/private': Permission denied
find: '/var/log/private': Permission denied
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/cache/private': Permission denied
astou@DESKTOP-86U4SBR:~$
```

Réponse3 :

```
astou@DESKTOP-86U4SBR:~$ find /usr/include -type f -name "*.h" -exec grep -H "include" {} +
/usr/include/iproute2/bpf_elf.h:#include <asm/types.h>
astou@DESKTOP-86U4SBR:~$
```

Activer Windows

Exercice5 : Méta-caractères

Cette section a introduit l'utilisation de méta-caractères en utilisant un exemple en Python. Les participants ont compris comment créer et manipuler des fichiers en utilisant des scripts.

Réponse1 :

Commande: (voir le script Python dans l'énoncé)

Résultat: Le script crée 16 fichiers, chaque fichier contenant un nombre et suivi d'un saut de ligne

```
astou@DESKTOP-86U4SBR:~$ python3 genf.py 16
astou@DESKTOP-86U4SBR:~$ ls
aer  cvb  f0  f10  f12  f14  f2  f4  f6  f8  genf.py  tp-auto  tut
azer dd  f1  f11  f13  f15  f3  f5  f7  f9  test  trez  wc
astou@DESKTOP-86U4SBR:~$ ls -l -h
total 84K
lrwxrwxrwx 1 astou astou 7 Jan 24 16:12 aer -> tp-auto
-rw-r--r-- 1 astou astou 0 Jan 24 15:20 azer
-rw-r--r-- 1 astou astou 8 Jan 24 16:14 cvb
-rw-r--r-- 1 astou astou 22 Jan 24 16:15 dd
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f0
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f1
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f10
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f11
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f12
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f13
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f14
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f15
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f2
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f3
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f4
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f5
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f6
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f7
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f8
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f9
-rwxr-xr-x 1 astou astou 121 Jan 28 22:31 genf.py
```


On constate que le script a créé 16 fichiers de f0 à f15, tous ces fichiers à l'intérieur contiennent le numéro qui est attribué. Par exemple le fichier f15 contient à l'intérieur 15, le fichier f12 contient 12 ainsi de suite. La taille des fichiers varie entre 2 et 3 octet en fonction du nombre de digit, f10 à f15 les fichiers sont de 3 octets et les fichiers à 1 digit font 2 octet.

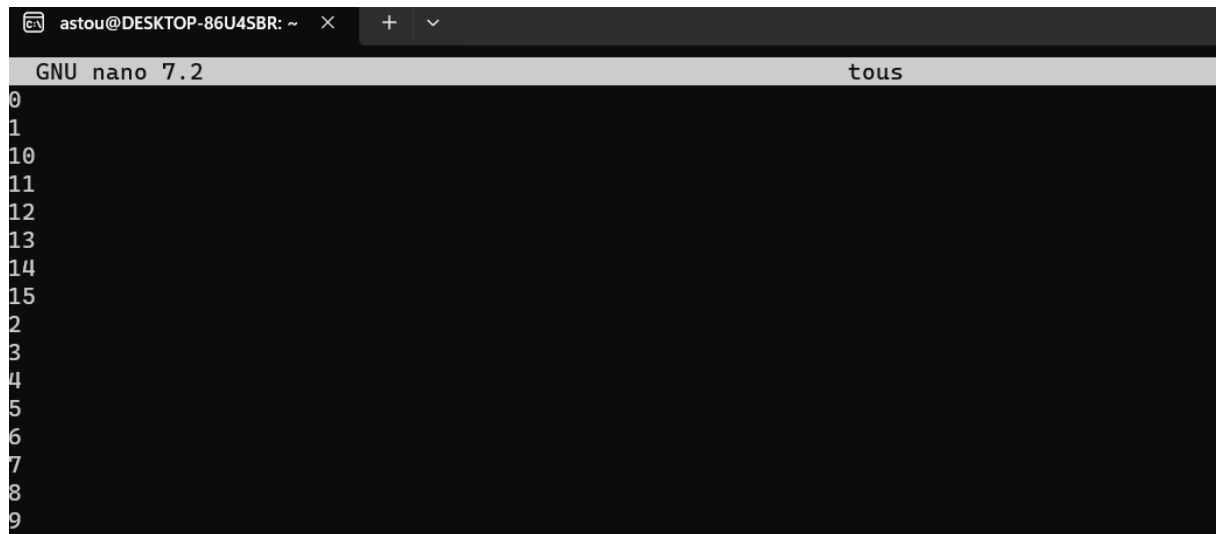
Réponse2 :

Commande: `cat f* > tous`

Résultat: Concatène tous les fichiers générés par le script Python en un seul fichier "tous".

J'ai concaténé tous les fichiers générés par le script Python en un seul fichier "tous".

```
astou@DESKTOP-86U4SBR:~$ cat f* > tous
```



Le fichier tous contient désormais la concaténation de tous les fichiers précédemment créés.

Réponse3 :

- Commande: `wc -c tous`
- Résultat: Affiche la taille en octets du fichier "tous".
- Commande: `wc -l tous`
- Résultat: Affiche le nombre de ligne du fichier "tous".

```
astou@DESKTOP-86U4SBR:~$ wc -l tous
16 tous
astou@DESKTOP-86U4SBR:~$ wc -c tous
38 tous
astou@DESKTOP-86U4SBR:~$
```

On constate que le fichier tous contient 16 lignes et a pour taille 38 octets.

Réponse4 :

Commande: `grep "1" tous | wc -l`

Résultat: Affiche le nombre de lignes dans "tous" contenant le chiffre 1.

```
astou@DESKTOP-86U4SBR:~$ grep 1 tous | wc -l
7
astou@DESKTOP-86U4SBR:~$ |
```

Réponse5 :

Commande : `cut -d: -f1 /etc/passwd | sort`

Résultat : Affiche la liste des noms de login depuis le fichier /etc/passwd triés alphabétiquement.

```
astou@DESKTOP-86U4SBR:~$ cut -d: -f1 /etc/passwd | sort
_apt
astou
backup
bin
daemon
games
irc
list
lp
mail
man
news
nobody
proxy
root
sync
sys
systemd-network
uucp
www-data
astou@DESKTOP-86U4SBR:~$ |
```

Réponse6 :

Commande: `ls /usr/include/std*.h`

Résultat: Affiche les noms des fichiers dans /usr/include qui commencent par "std" et se terminent par ".h".

```
astou@DESKTOP-86U4SBR:~$ find /usr/include -name 'std*.h'
/usr/include/stdtimu.h
/usr/include/stdastoiu.h
astou@DESKTOP-86U4SBR:~$ find /usr/include | grep '/usr/include/std' | grep '.hs'
astou@DESKTOP-86U4SBR:~$ find /usr/include | grep '/usr/include/std' | grep '.h'
/usr/include/stdtimu.h
/usr/include/stdastoiu.h
astou@DESKTOP-86U4SBR:~$ |
```

Exercice06 : Variables d'environnement en shell (bash)

Réponse1 :

Afficher la liste des variables

```
astou@DESKTOP-86U4SBR:~$ find /home/astou -type f | wc -l
31
astou@DESKTOP-86U4SBR:~$ |
```

Commande : `script.sh`

- Résultat : Le script est exécuté avec succès, confirmant l'ajout temporaire de "outils" au PATH.
2. Modification permanente du PATH dans le fichier ~/.bashrc:
- Commande : `echo 'export PATH=$PATH:$HOME/outils' >> ~/.bashrc`
 - Résultat : Le PATH est modifié de manière permanente dans le fichier de configuration du shell.

EXERCICE 7 - Propagation des variables d'environnement :

Une séquence de commandes a été exécutée pour observer la propagation des variables d'environnement entre différents shells.

1. Commandes exécutées et observations :
- Lignes 0-9 du script Python ont été exécutées avec des commentaires expliquant chaque étape.

Que se passe-t-il lors de la première commande (ligne 0) ?

- La commande `echo $ZORGLUB` tente d'afficher la valeur de la variable d'environnement `$ZORGLUB`. Cependant, cette variable n'est pas définie (n'existe pas), donc elle affichera probablement une ligne vide ou le message "variable d'environnement vide" (selon le shell utilisé).

La commande `bash` (ligne 4) ouvre un nouveau shell, qui hérite des variables de l'ancien. Que s'affiche-t-il à la ligne 11 ? Expliquer pourquoi.

- Lors de l'ouverture du nouveau shell avec `bash` à la ligne 4, ce shell hérite des variables d'environnement du shell parent (celui qui a exécuté le script). Cependant, lorsque le sous-shell se termine avec `exit` à la ligne 9, il ne transmet pas les modifications de variables à son shell parent.
- À la ligne 11, lorsque nous essayons d'afficher `$TRUCBIS`, cela affiche "hoho". Cela confirme que la variable `TRUCBIS` est définie dans le sous-shell, mais sa valeur n'est pas transmise au shell parent (ligne 10).

En résumé, l'ouverture d'un sous-shell hérite des variables du shell parent, mais les modifications dans le sous-shell ne sont pas répercutées sur le shell parent après la fermeture du sous-shell.

```
astou@DESKTOP-86U4SBR:~$ find /home/astou -type f | wc -l
31
astou@DESKTOP-86U4SBR:~$ |
```

EXERCICE8 :

- **find /chemin/du/repertoire -type f** : Recherche tous les fichiers dans le répertoire spécifié et ses sous-répertoires.
- **|** : Passe la liste des fichiers trouvés à la commande suivante.
- **wc -l** : Compte le nombre de lignes dans la sortie de la commande **find**, ce qui équivaut au nombre total de fichiers.

```
astou@DESKTOP-86U4SBR:~$ find /home/astou -type f | wc -l
31
astou@DESKTOP-86U4SBR:~$ |
```

Avec cette commande, nous avons obtenu le nombre de fichier demandé.

Exercice9 : Révisions sur les tubes

1. **Différence entre tee et cat :**
 - **cat** : Concatène et affiche le contenu des fichiers.
 - **tee** : Lit depuis l'entrée standard et écrit à la fois sur l'entrée standard et dans un ou plusieurs fichiers.
2. **Commandes suivantes :**
 - **\$ ls | cat** : Affiche le contenu du répertoire courant en utilisant **cat** qui, dans ce cas, n'ajoute pas de valeur significative.

```

astou@DESKTOP-86U4SBR:~$ ls | cat
aer
azer
cvb
dd
f0
f1
f10
f11
f12
f13
f14
f15
f2
f3
f4
f5
f6
f7
f8
f9
genf.py
outils
test
tous
tp-auto
trez
tut
wc
astou@DESKTOP-86U4SBR:~$ |

```

- **\$ ls -l | cat > liste** : Affiche le contenu détaillé du répertoire courant avec **ls -l** et écrit la même sortie dans un fichier appelé "liste".

```

astou@DESKTOP-86U4SBR:~$ ls -l | cat > liste
astou@DESKTOP-86U4SBR:~$ cat liste
total 92
lrwxrwxrwx 1 astou astou 7 Jan 24 16:12 aer -> tp-auto
-rw-r--r-- 1 astou astou 0 Jan 24 15:20 azer
-rw-r--r-- 1 astou astou 8 Jan 24 16:14 cvb
-rw-r--r-- 1 astou astou 22 Jan 24 16:15 dd
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f0
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f1
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f10

```

- **\$ ls -l | tee liste** : Affiche le contenu détaillé du répertoire courant avec **ls -l** et écrit la même sortie dans un fichier "liste".

```

astou@DESKTOP-86U4SBR:~$ ls -l | tee liste
total 92
lrwxrwxrwx 1 astou astou 7 Jan 24 16:12 aer -> tp-auto
-rw-r--r-- 1 astou astou 0 Jan 24 15:20 azer
-rw-r--r-- 1 astou astou 8 Jan 24 16:14 cvb
-rw-r--r-- 1 astou astou 22 Jan 24 16:15 dd
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f0
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f1
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f10
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f11
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f12
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f13
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f14
-rw-r--r-- 1 astou astou 3 Jan 28 22:35 f15
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f2
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f3
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f4
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f5
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f6
-rw-r--r-- 1 astou astou 2 Jan 28 22:35 f7

```

- `$ ls -l | tee liste | wc -l` : Affiche le contenu détaillé du répertoire courant, écrit la même sortie dans "liste", et compte le nombre total de lignes (fichiers et répertoires) avec `wc -l`.

```

astou@DESKTOP-86U4SBR:~$ ls -l | tee liste | wc -l
30
astou@DESKTOP-86U4SBR:~$ |

```

Explication :

- **cat** est utilisé pour afficher le contenu des fichiers ou de l'entrée standard.
- **tee** est utilisé pour diviser le flux, envoyant une copie à la sortie standard et une ou plusieurs copies à des fichiers spécifiés.
- Les commandes avec **ls -l** génèrent une liste détaillée du contenu du répertoire courant.
- La redirection (`>`) écrit la sortie dans un fichier spécifié.
- Les tuyaux (`|`) dirigent la sortie d'une commande vers une autre.
- **wc -l** compte le nombre de lignes dans la sortie, donnant ainsi le nombre total de fichiers ou répertoires dans la liste.

CONCLUSION

En conclusion, ce TP a permis de consolider les connaissances sur les commandes de base d'UNIX et le shell Bash. Les participants ont également acquis des compétences dans l'organisation des fichiers, les redirections, l'utilisation de variables d'environnement, la commande `find`, les méta-caractères, et d'autres concepts essentiels pour la manipulation efficace du système d'exploitation. Ces compétences seront précieuses dans le cadre de tâches plus avancées en administration système et en programmation.