

SYSTÈME D'EXPLOITATION OPERATING SYSTEM II - CORRECTION

Redirection

Exercice 0

```
marion@marion-Inspiron-7501:~$ ssh lfclxx@marion.szpieg.fr
lfclxx@marion.szpieg.fr's password:
Last login: Thu Jan 28 18:56:58 2021 from 89.115.143.200
lfclxx@SshSzpieg:~$ history -c
lfclxx@SshSzpieg:~$ mkdir tpSE2
lfclxx@SshSzpieg:~$ cd tpSE2/
lfclxx@SshSzpieg:~/tpSE2$
```

Exercice 1 : afficher les droits des fichiers (ls -l)

```
lfclxx@SshSzpieg:~/tpSE2$ cp /home/lfclpartage/tpSE1/documents/
fiche.txt .
lfclxx@SshSzpieg:~/tpSE2$ ls -l
total 0
-rw-r--r-- 1 lfclxx lfcl 0 janv. 28 19:15 fiche.txt
```

« lfclxx » est le **propriétaire** du fichier

« lfcl » est le **groupe rattaché** au fichier

« 0 » indique que c'est un fichier qui prend 0 octet sur le disque dur. C'est donc un fichier vide.

« - » indique que c'est un fichier

« rw- » indique que le **propriétaire** du fichier a le droit de le lire (« r »), d'écrire dedans (« w ») mais pas de l'exécuter

« r-- » indique que le **groupe rattaché** au fichier a le droit de le lire (« r »), mais pas d'écrire dedans, ni de l'exécuter

« r-- » indique que les **autres utilisateurs** au fichier a le droit de le lire (« r »), mais pas d'écrire dedans, ni de l'exécuter

« janv 28 19:15 » correspond à la date et à l'heure de la dernière modification du fichier

« fiche.txt » est le nom du fichier

Exercice 2 : redirection des flux de sortie (> et >>)

- a) `lfclxx@SshSzpieg:~/tpSE2$ echo Bonjour`
Bonjour
b) Cette commande sert à afficher un message à l'écran
- a) b) `lfclxx@SshSzpieg:~/tpSE2$ echo Bonjour > fiche.txt`
`lfclxx@SshSzpieg:~/tpSE2$ ls -l`
total 4
-rw-r--r-- 1 lfclxx lfcl 8 janv. 28 19:42 fiche.txt
c) Avec la commande « `echo Bonjour > fiche.txt` », le message « Bonjour » a été écrit dans le fichier `fiche.txt`. Ça n'est donc plus un fichier vide. 8 correspond au nombre d'octets sauvegardé dans par le fichier `fiche.txt`.
Chaque caractère de « Bonjour » prend un octet et la fin de ligne prend aussi un octet donc $7 + 1 = 8$.
- a) `lfclxx@SshSzpieg:~/tpSE2$ cat fiche.txt`
Bonjour
b) La commande « `cat` » permet d'afficher à l'écran le contenu d'un fichier texte.

Exercice 3 : redirection des flux de sortie (> et >>)

Avec le caractère « > », la sortie redirigée écrase ce qu'il y avait dans le fichier cible, alors qu'avec les caractères « >> », appelés « chevron », le message est rajouté à la fin du fichier fiche.txt, après un saut de ligne.

Exercice 4 : redirection des flux de sortie (> et >>)

```
lfclxx@SshSzpieg:~$ find . -iname "*.txt"
./tpSE1/images/photosVac/photo1.txt
./tpSE1/images/ski/gdbd3.txt
./tpSE1/documents/boulot/rapport.txt
./tpSE1/documents/fiche.txt
```

- « . » signifie qu'on cherche à partir du répertoire courant
- « -iname » le caractère « i » signifie qu'on ignore la casse et le mot « name » indique qu'on fait une recherche par rapport au nom du répertoire ou du fichier. Ignorer la casse signifie ne pas faire de différence entre majuscule et miniscule.

Exercice 5 : redirection des flux de sortie (>)

```
lfclxx@SshSzpieg:~$ find . -iname "*.txt" > result.txt
lfclxx@SshSzpieg:~$ cat result.txt
./tpSE1/images/photosVac/photo1.txt
./tpSE1/images/ski/gdbd3.txt
./tpSE1/documents/boulot/rapport.txt
./tpSE1/documents/fiche.txt
./result.txt
```

1. a) Voir copie d'écran ci-dessus.
b) La sortie est redirigée dans un fichier, donc rien n'est affiché à l'écran.
2. a) Voir copie d'écran ci-dessus.
b) Le fichier « result.txt » a été créé par la commande « find », et il contient le résultat de cette commande.

Exercice 6 : redirection des flux de sortie (>)

1.

```
lfclxx@SshSzpieg:~$ find /home/ -iname "*.txt"
find: '/home/mri4a79': Permission non accordée
find: '/home/mri4a4': Permission non accordée
find: '/home/marion': Permission non accordée
find: '/home/lfcl14': Permission non accordée
find: '/home/mri4a21': Permission non accordée
find: '/home/eleve59': Permission non accordée
find: '/home/eleve8': Permission non accordée
find: '/home/mri4a69': Permission non accordée
```
2. La différence avec l'exercice 4 est qu'on fait la recherche dans le répertoire /home/.
3. On a toutes ces lignes « Permission non accordée » car on n'a pas les droits pour regarder le contenu de ces répertoires qui appartiennent à d'autres utilisateurs du serveur.

Exercice 7 : redirection des flux de sortie (>)

1.

```
lfclxx@SshSzpieg:~$ find /home/ -iname "*.txt" > result2.txt
find: '/home/mri4a79': Permission non accordée
find: '/home/mri4a4': Permission non accordée
find: '/home/marion': Permission non accordée
find: '/home/lfcl14': Permission non accordée
find: '/home/mri4a21': Permission non accordée
```
2. Ces noms de fichier ont été redirigés à cause du « > »

3. Les informations sur les fichiers trouvés par la commande « find » sont dans le fichier « result2.txt ».
4. Avec la commande « cat » :

```
lfclxx@SshSzpieg:~$ cat result2.txt
/home/lfclxx/result2.txt
/home/lfclxx/tpSE1/images/photosVac/photo1.txt
/home/lfclxx/tpSE1/images/ski/gdbd3.txt
/home/lfclxx/tpSE1/documents/boulot/rapport.txt
/home/lfclxx/tpSE1/documents/fiche.txt
/home/lfclxx/result.txt
/home/lfclpartage/tpSE2/fiche.txt
/home/lfclpartage/tpSE1/images/photosVac/photo1.txt
/home/lfclpartage/tpSE1/images/ski/gdbd3.txt
/home/lfclpartage/tpSE1/documents/boulot/rapport.txt
/home/lfclpartage/tpSE1/documents/fiche.txt
```

Exercice 8 : redirection des flux de sortie (>)

1.

```
lfclxx@SshSzpieg:~$ find /home/ -iname "*.txt" > result2.txt 2> erreurs.txt
lfclxx@SshSzpieg:~$
```
2. Il n'y a aucun résultat à l'écran car les deux flux de sortie sont redirigés. Le flux de sortie standard est redirigé vers le fichier result2.txt, et le flux de sortie d'erreur est redirigé vers le fichier erreurs.txt.

Exercice 9 : permission et droit des fichiers

1.

```
lfclxx@SshSzpieg:~/tpSE2$ ls -l fiche.txt
-rw-r--r-- 1 lfclxx lfcl 18 janv. 28 19:50 fiche.txt
```
2.

```
lfclxx@SshSzpieg:~/tpSE2$ chmod u-w fiche.txt
```
3. La commande « echo » échoue car comme on le voit à la 3^e commande, l'utilisateur lfclxx, qui est propriétaire du fichier fiche.txt n'a pas le droit d'écrire dedans (il n'y a plus de « w » après le 1^{er} « r »).
4. La commande « chmod » permet de modifier les droits d'accès à un fichier.

```
lfclxx@SshSzpieg:~/tpSE2$ ls -l fiche.txt
-r--r--r-- 1 lfclxx lfcl 18 janv. 28 19:50 fiche.txt
lfclxx@SshSzpieg:~/tpSE2$ echo fin >> fiche.txt
-bash: fiche.txt: Permission non accordée
```

Exercice 10 : permission et droit des fichiers

1.

```
lfclxx@SshSzpieg:~/tpSE2$ chmod u+w,u-r fiche.txt
lfclxx@SshSzpieg:~/tpSE2$ ls -l fiche.txt
--w-r--r-- 1 lfclxx lfcl 18 janv. 28 19:50 fiche.txt
lfclxx@SshSzpieg:~/tpSE2$ echo fin >> fiche.txt
lfclxx@SshSzpieg:~/tpSE2$ cat fiche.txt
cat: fiche.txt: Permission non accordée
lfclxx@SshSzpieg:~/tpSE2$ chmod u+r fiche.txt
lfclxx@SshSzpieg:~/tpSE2$ ls -l fiche.txt
-rw-r--r-- 1 lfclxx lfcl 22 janv. 28 22:35 fiche.txt
lfclxx@SshSzpieg:~/tpSE2$ cat fiche.txt
Bonsoir
Boa tarde
fin
```
2. La ligne « echo fin >> fiche.txt » fonctionne ici car à la première ligne, on a autorisé le propriétaire du fichier fiche.txt à écrire dedans (« u+w »).
3. La première commande « cat fiche.txt » échoue car la première commande « chmod » a retiré les droits de lecture au propriétaire de fiche.txt (« u-r »), alors que la 2^e fois, grâce à la 5^e commande, les droits d'écriture ont été remis au propriétaire (« u+r »).

Exercice 11 : permission et droit des fichiers

1. Taper les commandes suivantes :
2. Taper ensuite la commande « ./monpgr.sh »

- On obtient un message d'erreur. C'est dû au fait que le propriétaire de monpgr.sh n'a pas le droit d'exécuter le programme.
- Il faut changer les droit avec la commande « chmod » :
Proposer une solution pour éviter l'erreur. La mettre en place puis lancer de nouveau le fichier exécutable.

```
lfclxx@SshSzpieg:~/tpSE2$ chmod u+x monpgr.sh
lfclxx@SshSzpieg:~/tpSE2$ ls -l monpgr.sh
-rwxr--r-- 1 lfclxx lfcl 78 janv. 28 22:49 monpgr.sh
lfclxx@SshSzpieg:~/tpSE2$
```

Exercice 12 : permission et droit des fichiers

```
1. lfclxx@SshSzpieg:~$ ls -l /home/lfclpartage/tpSE2/
total 24
-rw-r--r-- 1 marion prof 34 janv. 31 20:46 fiche2.txt
-rw-r--r-- 1 marion prof 42 janv. 31 20:48 fiche3.txt
-rw-r--r-- 1 marion prof 35 janv. 31 14:07 fiche.txt
-rwxr-xr-x 1 marion prof 77 janv. 31 20:49 marion2prog.sh
-rwxr-xr-- 1 marion lfcl 77 janv. 31 20:51 marion3prog.sh
-rwxr-xr-- 1 marion prof 76 janv. 28 17:32 marionprog.sh
lfclxx@SshSzpieg:~$ id
uid=1163(lfclxx) gid=1002(lfcl) groupes=1002(lfcl)
```

- a) `lfclxx@SshSzpieg:/home/lfclpartage/tpSE2$./marionprog.sh`
-bash: ./marionprog.sh: Permission non accordée
On ne peut pas exécuter le programme marionprog.sh, car en tant que « lfclxx », on fait partie de l'ensemble « other », qui n'a pas le droit d'exécuter le script :

```
-rwxr-xr-- 1 marion prof 76 janv. 28 17:32 marionprog.sh
```

Il n'y a pas de « x », donc on ne peut pas l'exécuter.

```
b) lfclxx@SshSzpieg:/home/lfclpartage/tpSE2$ ./marion2prog.sh
hello word
README tpRx1 tpRx2 tpSE1 tpSE2
M. Peuriere et Mme Szpieg sont...
```

Cette fois-ci on a le droit d'exécuter ce programme, car on fait partie de l'ensemble « other », qui a le droit d'exécuter le script :

```
-rwxr-xr-x 1 marion prof 77 janv. 31 20:49 marion2prog.sh
```

Cette fois-ci, il y a le « x », donc on peut l'exécuter.

```
c) lfclxx@SshSzpieg:/home/lfclpartage/tpSE2$ ./marion3prog.sh
hello word
README tpRx1 tpRx2 tpSE1 tpSE2
... les meilleurs profs de NSI bien sûr!
```

Cette fois-ci on a le droit d'exécuter ce programme, car en tant que « lfclxx », on fait partie du groupe « lfcl » rattaché au fichier, qui a le droit d'exécuter le script :

```
-rwxr-xr-- 1 marion lfcl 77 janv. 31 20:51 marion3prog.sh
```

Les utilisateurs appartenant au groupe lfcl ont le droit d'exécuter le programme.

Exercice 13 : fin de TP

```
1. lfclxx@SshSzpieg:~$ tree .
.
├── erreurs.txt
├── result2.txt
├── result.txt
├── tpSE1
│   ├── documents
│   │   ├── boulot
│   │   └── rapport.txt
│   │   └── fiche.txt
│   ├── images
│   │   ├── photosVac
│   │   └── photo1.txt
│   │   └── ski
│   │       └── gdbd3.txt
├── tpSE2
│   ├── fiche.txt
│   └── monpgr.sh
7 directories, 9 files
```

puis :

```
lfclxx@SshSzpieg:~$ mv erreurs.txt tpSE2/
lfclxx@SshSzpieg:~$ mv result.txt tpSE2/
lfclxx@SshSzpieg:~$ mv result2.txt tpSE2/
lfclxx@SshSzpieg:~$ tree .
.
├── tpSE1
│   ├── documents
│   │   ├── boulot
│   │   └── rapport.txt
│   │   └── fiche.txt
│   ├── images
│   │   ├── photosVac
│   │   └── photo1.txt
│   │   └── ski
│   │       └── gdbd3.txt
├── tpSE2
│   ├── erreurs.txt
│   ├── fiche.txt
│   ├── monpgr.sh
│   ├── result2.txt
│   └── result.txt
7 directories, 9 files
```

2. `lfclxx@SshSzpieg:~$ history > /home/lfclxx/tpSE2/SE2hist.txt`
`lfclxx@SshSzpieg:~$ tree .`

```
.
├── tpSE1
│   ├── documents
│   │   ├── boulot
│   │   │   ├── rapport.txt
│   │   │   └── fiche.txt
│   │   └── images
│   │       ├── photosVac
│   │       │   └── photo1.txt
│   │       └── ski
│   │           └── gdbd3.txt
│   └── tpSE2
│       ├── erreurs.txt
│       ├── fiche.txt
│       ├── monpgr.sh
│       ├── result2.txt
│       ├── result.txt
│       └── SE2hist.txt
└── 7 directories, 10 files
```

La commande suivante :« `history > /home/lfclxx/tpSE2/SE2hist.txt` » redirige tout ce qu'on a tapé pendant le TP dans le fichier texte SE2hist.txt, dont le chemin absolu est /home/lfclxx/tpSE2/SE2hist.txt. Cette sauvegarde vous permettra ultérieurement de regarder les commandes que vous avez tapées.

3. Taper « `exit` » pour quitter le serveur « `marion.szpieg.fr` ».