

22113572t PEROT Maël G1-A

1) On utilise « mkdir » pour créer les répertoires, « cd » pour naviguer dans les répertoires et « chmod » pour changer les permissions.

```
22113572t@ST-2019026794:~$ mkdir linux1105
22113572t@ST-2019026794:~$ cd linux1105/
22113572t@ST-2019026794:~/linux1105$ mkdir autres
22113572t@ST-2019026794:~/linux1105$ mkdir executables
22113572t@ST-2019026794:~/linux1105$ cd ..
22113572t@ST-2019026794:~$ chmod 744 linux1105/
22113572t@ST-2019026794:~$ cd linux1105/
22113572t@ST-2019026794:~/linux1105$ chmod 744 autres/
22113572t@ST-2019026794:~/linux1105$ chmod 744 executables/
```

2)

a) L'expression régulière pour les chaîne de caractères commençant par « drwx » est :
drwx.*

b) L'expression régulière pour les chaîne composée de 0 à n occurrences de "exec" est :
(exec)*

c) L'expression régulière pour les chaîne de caractères qui commence par un « l » et finit par un « 5 » et qui ne comporte aucun des caractères spéciaux suivants : « ., - # \ ; , » est :
l[^.,-#\;]*5

3) On va utiliser « ls » avec l'option -d pour afficher seulement les répertoires et un pipeline dans « grep » pour garder ceux qui ne contiennent pas soleil.

```
22113572t@ST-2019026794:~/linux1105$ ls -d | grep [^soleil]
```

4) On va utiliser « ls » pour afficher le contenu de linux1105 et un pipeline dans « grep » pour garder ceux qui contiennent le mot soleil et un autre pipeline dans « wc » avec l'option -l pour compter le nombre de lignes restantes.

```
22113572t@ST-2019026794:~/linux1105$ ls | grep soleil | wc -l
2
```

5) On utilise la commande « mv » pour déplacer un fichier

```
22113572t@ST-2019026794:~/linux1105$ mv vacances/devoir.txt autres/
22113572t@ST-2019026794:~/linux1105$ ls autres/
devoir.txt
```

6) On utilise les commande « history » et on fait un pipeline dans la commande « grep » pour garder les lignes commençant par un espace ou 1 puis un chiffre entre 1 et 9.

```
22113572t@ST-2019026794:~/linux1105$ history | grep '^ [1 ][1-9]'
```

1	start first_site.html
2	cd Documents/
3	ls
4	start first_site.html
5	echo \$NOM
6	echo \$PRENOM
7	echo \$AGE
8	NOM=Mael
9	exit
11	echo \$PRENOM
12	AGE=
13	echo \$AGE
14	man unset
15	unset AGE
16	echo \$AGE
17	ls
18	ls -a
19	cat .profile

7) On va utiliser l'option -i de la commande « diff » pour ignorer la casse

```
22113572t@ST-2019026794:~/linux1105$ diff -i passwd1 passwd2
```

```
1,2d0
<
<
3a2
>
10d8
< lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
16d13
< backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
18a16
```

8) On va modifier les droits avec « chmod » et on peut ensuite exécuter le programme.

```
22113572t@ST-2019026794:~/linux1105$ chmod 750 executables/affiche.sh
22113572t@ST-2019026794:~/linux1105$ ./executables/affiche.sh
```

```
200: 2 2 2 5 5
198: 2 3 3 11
196: 2 2 7 7
194: 2 97
```

9) On va utiliser la commande « ps » avec aux pour avoir plus d'informations qu'on va mettre dans un pipeline vers « grep » pour garder seulement les informations concernant affiche.sh.

```
22113572t@ST-2019026794:~/linux1105$ ps aux | grep affiche.sh
```

```
2211357+ 12740 0.0 0.0 6932 3200 pts/0 T 16:49 0:00 /bin/bash ./executables/affiche.sh
```

10) Pour lancer en arrière plan on rajoute un & à la fin de la commande et on fait la commande « kill -SIGTSTP numProcessus » pour interrompre temporairement. On obtient ce numéro de processus avec la commande « ps ». On utilise ensuite « fg » pour relancer affiche.sh et on fait un contrôle-C pour tuer le programme.

```
22113572t@ST-2019026794:~/linux1105$ ./executables/affiche.sh &
[2] 14006
22113572t@ST-2019026794:~/linux1105$ 200: 2 2 2 5 5
ps 198: 2 3 3 11
```

PID	TTY	TIME	CMD
1849	pts/0	00:00:00	bash
3342	pts/0	00:00:00	oosplash
3382	pts/0	00:00:34	soffice.bin
14006	pts/0	00:00:00	affiche.sh
14010	pts/0	00:00:00	sleep
14011	pts/0	00:00:00	ps

```
22113572t@ST-2019026794:~/linux1105$ 196: 2 2 7 7
194: 2 97
ki192: 2 2 2 2 2 2 3
ll190: 2 5 19
```

```
22113572t@ST-2019026794:~/linux1105$ fg
./executables/affiche.sh
132: 2 2 3 11
130: 2 5 13
kill128: 2 2 2 2 2 2 2
126: 2 3 3 7
14124: 2 2 31
006
122: 2 61
120: 2 2 2 3 5
118: 2 59
^C
```

11) On utilise la commande « cat » pour afficher le contenu du fichier puis on utilise la commande « grep » pour garder les lignes voulues.

```
22113572t@ST-2019026794:~/linux1105$ cat sources_list | grep ^deb-src
deb-src http://deb.debian.org/debian/ bullseye main
deb-src http://security.debian.org/debian-security bullseye-security main contrib
deb-src http://deb.debian.org/debian/ bullseye-updates main contrib
```

12) Même principe que la question précédente.

```
22113572t@ST-2019026794:~/linux1105$ cat sources_list | grep ^[^#]
deb http://deb.debian.org/debian/ bullseye main
deb-src http://deb.debian.org/debian/ bullseye main
deb http://security.debian.org/debian-security bullseye-security main contrib
deb-src http://security.debian.org/debian-security bullseye-security main contrib
deb http://deb.debian.org/debian/ bullseye-updates main contrib
deb-src http://deb.debian.org/debian/ bullseye-updates main contrib
```

13) On va utiliser la commande « awk » pour garder la première, cinquième et sixième colonne et on fait ensuite une redirection de cette commande dans un nouveau fichier.

```

22113572t@ST-2019026794:~/linux1105$ awk -F "-" '{ print $1 $5 $6}' df_txt > new_file
22113572t@ST-2019026794:~/linux1105$ cat new_file
Sys. fich          Uti%  Monté sur
udev               0%    /dev
tmpfs              1%    /run
/dev/sda1          88%    /
tmpfs              0%    /dev/shm
tmpfs              1%    /run/lock
/dev/sda5          1%    /tmp
tmpfs              1%    /run/user/1000

```

14)

```

#!/bin/bash

if [ $# -lt 2 ]
then
    echo "script.sh: pas assez d'arguments"
    exit 1
fi

if [ -f $1 ]
then
    echo "script.sh: l'argument n'est pas un repertoire"
    exit 1
fi

if ![ -d $1 ]
then
    mkdir $1
fi

for i in `seq 2 $#`;
do
    ext=$i
    cp *.$ext $1
done

```