

CF - Contrôle Final (02h00) - Les documents sont interdits (solution)

Nom:-----

Prénom:-----

Groupe:-----

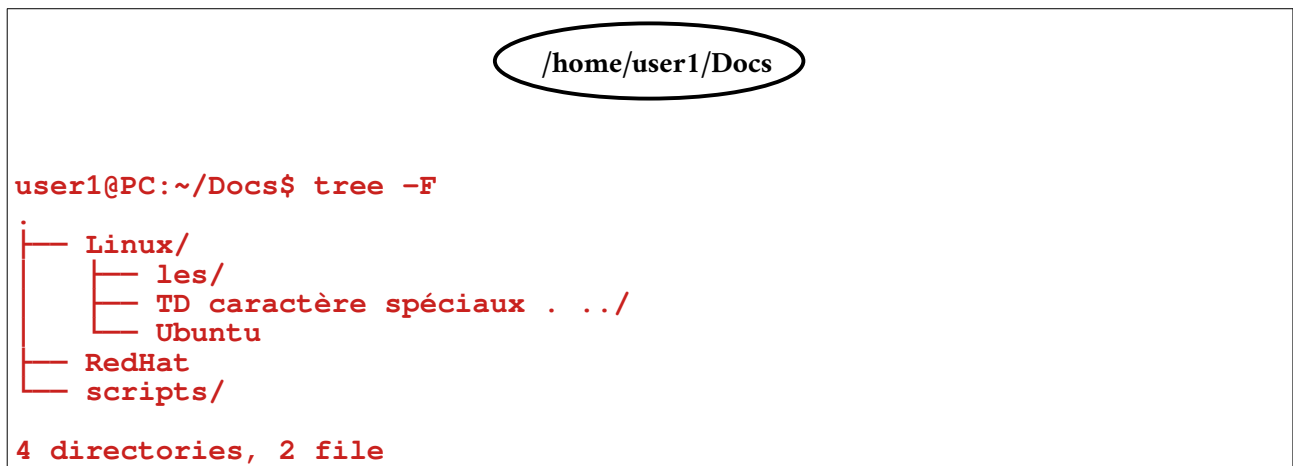
Exercice 01: (les fichiers et les répertoires, les redirections et les filtres; 06points)

Partie A (les fichiers et les répertoires; 12*0,25=03 points)

Scénario A

```
user1@PC:~/Docs$ mkdir -p .././Docs/Linux/"TD caractère spéciaux . .." Linux/les scripts
user1@PC:~/Docs$ touch Linux/Ubuntu RedHat
```

Suite à l'exécution des commandes décrites dans le Scénario A, dessiner l'arborescence des répertoires créés à partir du répertoire `/home/user1/Docs`. Mettez les **répertoires** dans des **cercles** et les **fichiers** dans des **rectangles**. (0.25*6=01.5 point)



Scénario B

```
user1@PC:~/Docs$ rm -rf *
user1@PC:~/Docs$ cp /etc/passwd 'Fichier Users'
user1@PC:~/Docs$ cp ../.bash_history "Histoire de user1"
user1@PC:~/Docs$ cp -rf ../Bureau/ .
user1@PC:~/Docs$ mv *ser? Bureau/
user1@PC:~/Docs$ rm -rf !(Bureau)
```

Suite à l'exécution des commandes décrites dans le Scénario A, dessiner l'arborescence des répertoires créés à partir du répertoire `/home/user1/Docs`. Mettez les **répertoires** dans des **cercles** et les **fichiers** dans des **rectangles**. (0.25*6=01.5 point)

/home/user1/Docs

```
user1@PC:~/Docs$ tree
```

```
├── Bureau
│   ├── Fichier Users
│   └── Histoire de user1
```

1 directory, 2 files

Partie B (les redirections et les filtres; 12*0.25=03 points)

Compléter le **Scénario C** suivant à la place des pointillés.

Scénario C

```
user1@PC:~$ cat notes.csv
A;F;17.5
B;J;18
A;F;17.5
D;L;11
E;M;19
user1@PC:~$ cat 1 2 3
fichier 1
fichier 2
fichier 3
user1@PC:~$
user1@PC:~$ sed 's/;/ /g' notes.csv | sort -n -r -k 3 | uniq | cat -n
    1      E M 19
    2      B J 18
    3      A F 17.5
    4      D L 11
user1@PC:~$ grep -v ^A notes.csv >1 ; cut -d';' -f1,3 notes.csv >>2 2>&1
; HOSTNAME 2>3
user1@PC:~$ cat 1
B;J;18
D;L;11
E;M;19
user1@PC:~$ cat 2
fichier 2
A;17.5
B;18
A;17.5
D;11
E;19
user1@PC:~$ cat 3
HOSTNAME: command not found
user1@PC:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2951 janv. 17 11:54 /etc/passwd
user1@PC:~$ [ -w /etc/passwd ] || echo 1
1
user1@PC:~$ [ -r /etc/passwd ] && echo 2
2
```

Exercice 02: (les droits d'accès; 07 points)

Analyser le scenario suivant et répondre aux questions

Scénario A

```
user1@PC:~$ ls -ld / /home
drwxr-xr-x 26 root root 4096 janv. 12 12:37 /
drwxr-xr-x  7 root root 4096 janv. 17 11:54 /home
user1@PC:~$ groups user1 user2 user3 user4 root
user1 : user1
user2 : user2 sudo user1 user3
user3 : user3 user1
user4 : user4
root  : root
user1@PC:~$ ls -al source/ destination/
destination/:
total 8
drwxrwxrwt 2 user1 user1 4096 janv. 19 08:54 .
drwxr-xr-x 4 user1 user1 4096 janv. 17 14:06 ..
-rw-rw---- 1 user4 user4    0 janv. 17 11:14 fichier4

source/:
total 12
drwxrwxr-- 2 user1 user3 4096 janv. 17 11:00 .
drwxr-xr-x 4 user1 user1 4096 janv. 17 14:06 ..
-r--r--r-- 1 user1 user1    0 janv. 17 11:00 fichier1
-r---w-r-- 1 user2 user1    6 janv. 19 08:41 fichier2
--w-rw-r-- 1 user3 user3    0 janv. 17 11:00 fichier3
```

A: Complétez le tableau suivant en mettant dans chaque groupe ses utilisateurs (0.25*5=01.25 point).

Groupe sudo	Groupe user1	Groupe user2	Groupe user3	Groupe user4
user2	user1 user2 user3	user2	user2 user3	user4

Noter bien: Dans les deux tableaux suivants répondre par oui ou non sur les questions suivantes. Si la réponse est **juste** vous aurez +0.25, si la réponse est **fausse** vous aurez -0.25. La note minimale pour chaque question est **zéro**.

B: Est ce que **user1** peut **modifier** le contenu des fichiers **fichier1** et **fichier2** ? (0.25*2=0.5 point).

C: Est ce que **user3** peut **copier** les fichiers **fichier1** et **fichier2** vers **destination** ? (0.25*2=0.5 point).

D: Est ce que **user4** peut **lire** le contenu des fichiers **fichier1** et **fichier2** ? (0.25*2=0.5 point).

E: Est ce que **user3** peut **déplacer** les fichiers **fichier3** vers **destination** et le **fichier4** vers **source**? (0.25*2=0.5 point).

F: Est ce que **user2** peut **supprimer** les fichiers **fichier3** et **fichier4** ? (0.25*2=0.5 point).

G: Est ce que **user1** peut **supprimer** les fichiers **fichier3** et **fichier4** ? (0.25*2=0.5 point).

Fichiers/Question	fichier1	fichier2	fichier3	fichier4
B	non	oui		
C	oui	non		

D	non	non		
E			oui	non
F			oui	non
G			oui	oui

H: Est ce que l'utilisateur user2 peut exécuter les commandes suivantes ? (0.25*5=0.125 point).

Les commandes	Les réponses (oui/non)
user2@PC:/home/user1\$ chgrp user3 source/fichier2	oui
user2@PC:/home/user1\$ chgrp user4 source/fichier2	non
user2@PC:/home/user1\$ sudo apt-get update	oui
user2@PC:/home/user1\$ sudo chown user1:user3 source/fichier2	oui
user2@PC:/home/user1\$ chown user1:user3 source/fichier2	non

I: Quelle est la commande pour que l'utilisateur user2 donne les droits 750 à tous ses fichiers et ses répertoires ? (0.25*2=0.5 point).

```
user2@PC:~$ chmod -R 750 ~
```

J: Quelle est la commande pour que l'utilisateur user2 change seulement les droits de ses fichiers à 640 ? (0.25*2=0.5 point).

```
user2@PC:~$ find ~ -type f -exec chmod 640 {} \;
```

K: Quelle la commande pour que l'utilisateur user2 attribue automatiquement les droits 750 pour les nouveaux répertoires créés et les droits 640 pour les nouveaux fichiers créés ? (0.25*2=0.5 point).

```
user2@PC:~$ umask 027
```

CF - Contrôle Final (02h00) - Les documents sont interdits

Nom:-----

Prénom:-----

Groupe:-----

Exercice 03: (les scripts; 05 points)

Un enseignant a besoin d'un script pour:

- ✓ copier un fichier ou un répertoire dans les comptes des étudiants (tous les utilisateurs sauf l'enseignant celui qui exécute le script)
- ✓ évaluer les étudiants après le contrôle final (CF) sur le serveur.

1. Le script a un seul paramètre dans lequel il peut prendre seulement deux valeurs:

«**copier**» pour copier le fichier ou le répertoire dans tous les répertoires Bureau des utilisateurs sauf le compte enseignant (celui qui exécute le script).

«**evaluer**» pour évaluer les étudiants et afficher les notes sur l'écran.

2. Les tâches demandées aux étudiants:

- ✓ Créer un fichier **TP** dans le Bureau (01 point).
- ✓ Ecrire dans le fichier **TP** la phrase «Je suis userx» (userx le nom du compte de l'étudiant) (sur 01 point).

3. Pour faire la copie, l'enseignant doit d'abord tester s'il a les droits wx sur le répertoire Bureau de l'étudiant.

4. Le script informe l'enseignant s'il a oublié le paramètre ou s'il y a une erreur.

Un exemple d'exécution du script dans lequel user1 est l'enseignant et les autres utilisateurs sont les étudiants.

```
user1@PC:~$ ls /home/
user1 user2 user3 user4
user1@PC:~$ ./script
manque le paramètre d'entrée
user1@PC:~$ ./script copier
Introduire le nom du fichier ou du répertoire: cours
user1@PC:~$ ls /home/*/Bureau
/home/user1/Bureau:
/home/user2/Bureau:
cours
/home/user3/Bureau:
cours
/home/user4/Bureau:
cours
user1@PC:~$ ./script evaluer
La note de user2 est: 2/2
La note de user3 est: 1/2
La note de user4 est: 0/2
```

Le contenu du script avec des parties manquantes.

```
user1@PC:~$ cat -n script
1  [ ]/bin/bash
2  [ ]ensemble contient les utilisateurs sauf
3  [ ]celui qui exécute le script
4  [ ]ensemble=`ls /home/ | grep -v [ ]`
5  [ ]if [ [ ] = copier ]
6  [ ]then
7  [ ] [ ] "Introduire le nom du fichier ou du répertoire: " obj
```

```

8      for user in $ensembles
9      █
10     # L'enseignant va tester si il a les droits wx
11     if [ -█ /home/$user/Bureau -█ -█ /home/$user/Bureau ]
12     then
13         cp -rf "████" /home/$user/Bureau
14     else
15         echo "vérifier les droits wx du Bureau de $user"
16     fi
17     █
18     █
19     if [ █ = evaluer ]
20     then
21         for user in $ensembles
22         █
23             note=0
24             if [ █ /home/$user/Bureau/TP ]
25             then
26                 █ note=$note+1
27                 grep █ Je suis $user█ /home/$user/Bureau/TP
28                 if [ $? -█ 0 ]
29                 then
30                     █ note=$note+1
31                 fi
32             fi
33             echo "La note de $user est: █/2"
34             █
35     # On test si le paramètre est vide
36     else
37         if [ █ █ ]
38         then
39             echo "manque le paramètre d'entrée"
40         else
41             echo "les mots clés sont: 'copier' et 'evaluer'"
42         fi
43     fi
44 fi

```

Mettez dans le tableau suivant la partie manquante du script selon les demandes.

La ligne du script	Compléter par	La partie manquante
1 (0.25 point)	les caractères manquants	#!
2,3 (0.25 point)	un caractère	#
4 (0.25 point)	une variable du système	\$LOGNAME ou \$USER...
5,19 (0.25 point)	une variable paramètre	"\$1"
7 (0.5 point)	une commande et une option	read -p
9,22 (0.25 point)	un mot clé parmi les mot clés de la structure for	do
11 (0.75 point)	trois options dans l'ordre	-w -a -x ou -x -a -w
13 (0.25 point)	une variable	\$obj
17,34 (0.25 point)	un mot clé parmi les mot clés de la structure for	done
18 (0.25 point)	un mot clé parmi les mot clés de la structure if	else

24 (0.25 point)	une option	-f
26,30 (0.25 point)	une commande	let
27 (0.25 point)	un caractère	Les doubles quotes ""
28 (0.25 point)	une option	-eq
33 (0.25 point)	une variable	\$note
37 (0.5 point)	un optione et une variable paramètre	-z \$1

Exercice 04: (les processus; 02 points)

Dans chaque point, cocher toutes les affirmations correctes. Si l'affirmation sélectionné est **juste** vous aurez +0.25, si l'affirmation sélectionné est **fausse** vous aurez -0.25. La note minimale pour chaque point est **zéro**.

1.	<ul style="list-style-type: none"> • Un processus est un programme en cours d'exécution (en mémoire) • Un processus a une identité PID (Process ID) • Un processus n'a pas l'identité de son parent PPID (Parent Process ID) • Un processus n'a pas UID et GID de l'utilisateur qui a lancé le processus • Un processus n'a pas une priorité • Un processus a une priorité
2.	<ul style="list-style-type: none"> • La commande ps sans options permet d'afficher tous les processus en cours de l'exécution • La commande ps permet d'afficher les processus en cours d'exécution lancés par l'utilisateur depuis la console actuelle • Les informations affichées par ps sont PID TTY TIME CMD • ps -f affiche tous processus • ps -f affiche les même processus de la commande ps sans option avec plus d'information comme PID, PPID et UID • ps -f affiche les même processus de la commande ps sans option avec plus d'information comme PRI et NI
3.	<ul style="list-style-type: none"> • La commande ps -l affiche tous processus • La commande ps -l affiche tous processus avec plus d'informations comme PRI et NI • ps -f affiche les même processus de la commande ps sans option avec plus d'information techniques comme PRI et NI • ps avec l'option -U permet d'afficher les processus d'un utilisateur • On peut utiliser la commande ps avec seulement une seule option • PRI sa valeur varie entre 20 et -20 ; Plus la valeur est élevée plus le traitement est ralenti
4.	<ul style="list-style-type: none"> • NI priorité du processus ; Un nombre plus élevé signifie priorité inférieure • PRI priorité du processus ; Un nombre plus élevé signifie priorité supérieure • ps -e affiche tous les processus en cours d'exécution de tous les utilisateurs • ps -e n'existe pas