



« Examen 2014 Système d'exploitation UNIX »

Exercice 1 :

1. Affichez la liste des fichiers dont les noms comportent la lettre t en deuxième position et qui se termine par un chiffre numérique. La liste affichée doit être triée par ordre alphabétique
2. Faire une copie du fichier **/etc/hosts** dans **Mon_rep** sous le nom **hosts_pers** sans utiliser cp ni ouvrir d'éditeur de texte.
3. Affichez les 15 premières lignes du fichier **/etc/hosts**, et les lignes 15 à 20 ?
4. Vous désirez regrouper dans le répertoire **Mon_rep** tous les fichiers dont le nom contient un caractère minuscule suivi d'un caractère majuscule.
5. Créer le fichier **essai3.txt** qui regroupe les lignes du même n° de ligne des fichiers **essai1.txt** et **essai2.txt**.
6. Créer un répertoire nommé **iset** tel que les autres utilisateurs ne puissent pas lister son contenu mais puissent lire les fichiers qui y sont placés. On obtiendra
\$ ls iset
ls : iset: Permission denied
\$ cat iset/toto
<... le contenu du fichier toto (s'il existe) ...>

➔ Résolution :

Q1/ `ls ?t*[0-9] | sort`

Q2/ `cat /etc/hosts > hosts_pers`

Q3/ `head -n 15 /etc/hosts`
`tail -n +15 /etc/hosts | head -n 6`

Q4/ `cp *[a-z][A-Z]*` #/-- à revoir !!!!

Q5/ `paste essai1.txt essai2.txt > essai3.txt`
ça suppose que les deux fichiers `essai1.txt` et `essai2.txt` ont
le même nombre de lignes, sinon ça va être décalée

Q6/ `mkdir iset`
`chmod g-wx,o-wx iset/`



Exercice 2 :

1. Afficher la liste des fichiers plus vieux que 3 jours situés sous votre répertoire de connexion.
2. Créer un fichier texte quelconque. Écrire une seule commande shell qui recopie le contenu passé en majuscule de ce fichier dans un nouveau fichier nommé "**Titre.txt**".
3. Faire un script shell **users_bash** qui affiche les utilisateurs qui ont le type d'interpréteur **bash**, dont le fichier **/etc/passwd** sera donné en paramètre.
4. Créer un fichier **extremes** (attention, sans extension .txt), qui contient la première ligne de tous les fichiers d'extension **.org** et, à la suite, leur dernière ligne, le tout sans en-tête.
5. Comment faire pour afficher juste les lignes du fichier **extremes** où figure le mot recherché tel quel, et non pas ses variantes ? Par exemple : on cherche le mot «travail», mais pas «travailleur» ou «travailler».
6. Affichez la liste des processus dont vous êtes propriétaire.
7. Lancer une commande de recherche pour afficher tous les liens symboliques mais ayant une faible priorité (égal 10).
8. Modifier la protection tous les fichiers d'extension **.org** en ajoutant le droit de lecture pour le groupe, et le droit d'exécution pour les autres.
9. Écrire un script shell dont le nom est **process** permettant de copier la liste des processus de l'utilisateur exécutant ce script, puis afficher le nom de chaque processus sans doublons.

➔ Résolution :

Q1/ `find ~/ -type f -ctime -3`

Q2/

```
cat > fichier_bidon  
(On le remplit par une seule ligne puis ctrl-D)  
cat fichier_bidon | tr -d [:lower:] > Titre.txt
```

Q3/ `cat > users_bash`
`#!/bin/bash`
`echo `grep /bin/$1 /etc/passwd | cut -d : -f 1``
(ctrl-D)

Q4/ `(head -qn 1 *.org & tail -qn 1 *.org) > extremes`



Q5/ `grep -w MOTIF nom_fichier`

Q6/ `ps -u nom_user`

Q7/ `nice -n 10 ls -l | grep ^l`

Q8/ `chmod g+r,o+x *.org`

Q9/

```
cat > process
#!/bin/bash
echo `ps -f -u $USER | tr -s " " | cut -d " " -f 8 | tail -n +2 | sort | uniq`
(ctrl-D)
```



Exercice 3 :

1. Afficher tous les fichiers réguliers se trouvant sous votre répertoire d'accueil et qui n'ont pas été accédé dans les 10 derniers jours.
2. Affichez la taille et le nom de tous les fichiers et répertoires sous votre répertoire principal.
3. Renommer le répertoire « **rapports** » vers « **rapports2014** » et stocker les messages d'erreurs dans un fichier « **rapport.log** ».
4. Comment appeler la commande pour attribuer à tous les nouveaux fichiers les droits d'accès 'r---w----'.
5. Ecrire une ligne de commande qui permet de créer un fichier « **text3** » qui contiendra les lignes du fichier « **text1** » contenant le mot « **file** ». Ces lignes devront être triées sur le premier champ de chaque ligne.
6. Donner la commande qui affiche le nombre de ces lignes. (Ne comptez pas les lignes vides).

➔ Résolution :

- Q1/ `find ~/ -type f -atime 10`
- Q2/ `ls -l ~/`
- Q3/ `mv rapports rapports2014 2> rapport.log`
- Q4/ `umask 246`
- Q5/ `grep -w file text1 | sort > text3`
- Q6/ `grep -v ^$ text3 | wc -l`



Examen

« Système d'exploitation UNIX »

« Documents non autorisés »

Durée : 2h

Exercice 1:

1. Qu'est-ce qu'un shell ?
2. Écrire une ligne de commande qui compte le nombre de processus actifs dont vous êtes propriétaire.
3. Donner deux commandes pour reprendre l'exécution d'une instruction interrompue par un ^Z.
4. Écrire un script qui permet de créer un processus qui affiche le message « **Vous êtes à l'EMSI Rabat** » à chaque deux minute.
5. Lancer ce script en arrière plan.
6. Repérer le numéro du processus.
7. Essayez d'interrompre l'exécution de ce script mais dans une autre session.
8. Arrêter ce processus.
9. Visualisez l'ensemble des processus tournant sur la machine en arborescence dont vous êtes propriétaire.
10. Faire afficher les informations sur tous les processus non liés à un terminal

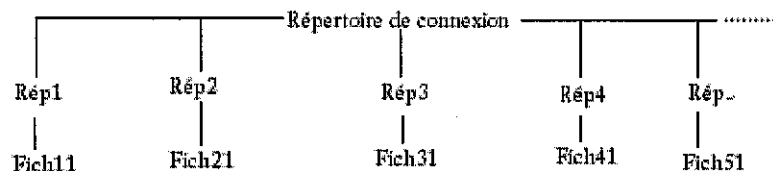
Exercice 2 :

1. Afficher juste les sous-répertoires de /dev.
2. Récupérer les lignes 11 à 17 du fichier **toto** de 21 lignes
3. Afficher la liste des répertoires de connexion de tous les utilisateurs.
4. Quels sont les dix plus gros fichiers de /usr/bin/ ?
5. Comment copier le contenu du fichier /etc/passwd dans un nouveau fichier nommé **Gestion**, en remplaçant les caractères « : » par des « | ».
6. Compter le total des lignes des fichiers dans /dev.
7. Quelle est la commande qui permet de changer la priorité d'un processus au cours de son exécution. Donner un exemple.
8. Changer les modes d'accès de tous les fichiers qui ont une extension '.txt' en r---w--w- (Utiliser le mode symbolique).
9. Copier les fichiers qui commencent et se terminent par une voyelle et de préserver toutes les informations concernant les fichiers dans /dev.
10. Déplacez les fichiers **vert.txt** et **bleu.txt** dans le répertoire **froid**.
11. Affichez page par page le contenu du fichier /etc/services.



Exercice 3 :

1. Ecrire un script qui recherche dans toute mon arborescence tous les fichiers qui n'ont pas été accédés depuis un temps T et dont la taille est supérieure à MIN. (T et MIN sont des arguments).
2. Écrivez le shell script **killprog** qui permet d'envoyer le signal SIGKILL à un processus désigné non pas par son PID mais par son nom.
Par exemple : killprog xterm
3. Faire un script qui crée un certain nombre de répertoires et chaque répertoire possède un fichier, ce nombre étant passé en paramètre



4. On ne s'intéresse ici qu'à des fichiers contenant un mot par ligne. Écrire un script qui compte le nombre de mots contenant une des lettres «b», «c» ou «t», et parmi eux, ceux qui ont au moins deux telles lettres. On donnera aussi le nombre de mots ne contenant aucune voyelle. Cela donnera par exemple :

Dans ce fichier 'le nom du fichier', vous avez :

45 mots contenant «b», «c» ou «t», et parmi eux,

12 contiennent deux de ces lettres au moins.

Il y a aussi 10 mots ne contenant aucune voyelle.

Donner à l'utilisateur de saisir le chemin du fichier.

Examen**« Systèmes d'exploitation 3 »****Durée : 2h****Exercice :**

1. Écrire un script Shell qui affiche le login d'un utilisateur dont on fournit l'UID.
2. Ecrire un script qui lit une ligne de caractères sur son entrée standard et l'écrit sur sa sortie, en passant tous les caractères en majuscules.
3. Écrire une ligne de commande qui compte le nombre de processus actifs dont vous êtes propriétaire.
4. Ecrire un script qui permet de créer un processus qui affiche le message « Vous êtes à l'EMSI RABAT » à chaque deux minute.
5. Lancer ce script en arrière plan.
6. Repérer le numéro du processus.
7. Essayez d'interrompre l'exécution de ce script mais dans une autre session. Que faut-il changer?
8. Arrêter ce processus.
9. Écrire un script Shell dont le nom est **process** permettant de copier la liste des processus de l'utilisateur exécutant ce script, puis afficher le nom de chaque processus sans doublons.
10. Ecrire un script qui recherche dans toute mon arborescence tous les fichiers qui n'ont pas été accédés depuis un temps T et dont la taille est supérieure à MIN. (T et MIN sont des arguments).
11. Ecrire un script Shell qui ajoute à l'intérieur du fichier un commentaire indiquant sa date de dernière modification, dont le chemin du fichier sera donné en paramètre.
12. Écrire un script « hello » qui obéisse à la spécification suivante :
 - Sans argument, il affiche « Hello user ! » en remplaçant « user » par le login de l'utilisateur courant.
 - Avec un argument « A », il affiche « Hello A ! ».
 - Avec plusieurs arguments « A B C D », il affiche « Hello A, B, C and D ! »

De plus, on demande que si le nom du script commence par «bonjour», le texte soit en français et pas en anglais. C'est-à-dire que le comportement doit être le suivant :

Dans ce fichier '**le nom du fichier**', vous avez :

```
$/hello Ghita
```

```
Hello Ghita !
```

```
$/hello Ghita Sara Rania
```

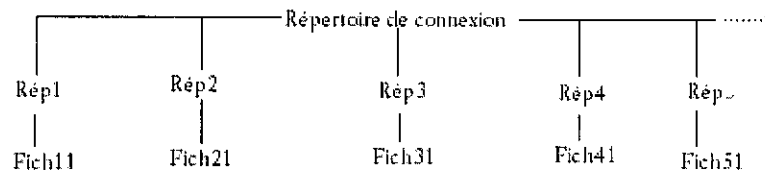
```
Hello Ghita, Sara, Rania!
```

```
$ln -s hello bonjour
```

```
$/bonjour Sara Rania
```

```
Bonjour Sara, Rania !
```

13. Faire un script qui crée un certain nombre de répertoires et chaque répertoire possède un fichier, ce nombre étant passé en paramètre



14. Écrivez le shell script **killprog** qui permet d'envoyer le signal SIGKILL à un processus désigné non pas par son PID mais par son nom.

Par exemple : killprog xterm

Bonus :

1. Quelle est la différence entre un « Shell » et un « Terminal » ? Ou est-ce la même chose ?
2. Comment consulter le manuel d'aide en ligne ?
3. Qu'est-ce qu'une variable prédéfinie ?
4. Qu'est-ce qu'un Shell script ?
5. Qu'est-ce qu'une redirection ? Au lieu d'avoir ce que l'on veut sur l'écran, on l'a dans un fichier ?

ls : affiche la liste des fichiers

ls -m : Affiche les fichiers en les séparant par une virgule au lieu de les présenter en colonnes.

ls -t : Affiche les fichiers par date, c'est-à-dire en les classant du récent au plus ancien.

ls -lu : Affiche les fichiers par date de dernier accès et indique cette date.

ls -F : Affiche les fichiers par type.

ls -S : Affiche les fichiers triés par ordre de taille décroissante.

ls -X : Affiche les fichiers par type d'extension.

ls -r : Affiche les fichiers en ordre alphabétique inverse.

more fichier : affiche le contenu de fichier

cp : Copie des fichiers dans un répertoire

cp -i : Avertit de l'existence d'un fichier du même nom et demande s'il peut le remplacer ou non.

cp -l : Permet de faire un lien en "dur" entre le fichier source et sa copie

cp -s : Permet de faire un lien "symbolique" entre le fichier source et sa copie

cp -p : Permet lors de la copie de préserver toutes les informations concernant le fichier.

cp -r : Permet de copier de manière récursive l'ensemble d'un répertoire et de ses sous répertoires

mv : Change le nom d'un fichier, déplace un fichier

mv -b : Va effectuer une sauvegarde des fichiers avant de les déplacer

mv -i : Demande pour chaque fichier et chaque répertoire s'il peut ou non le déplacer

mv -u : Demande à "mv" de ne pas supprimer le fichier si la date de modification est la même ou plus récente que son remplaçant.

rm : supprime un fichier

rm -d : Permet de supprimer un répertoire qu'il soit plein ou non.

rm -r : Permet de supprimer un répertoire et ses sous répertoires.

rm -f : Permet de supprimer les fichiers protégés en écriture et répertoires sans confirmation.

ln : Permet de faire un lien "symbolique" ou « dur » entre le fichier source et sa copie

ln -s permet de faire des liens symboliques

du : Affiche la taille d'une arborescence

pwd : Affiche le répertoire courant

man : affiche les pages de manuel de commande

date : affiche la date du système.

grep : recherche d'un motif dans un fichier

grep -c : Compte le nombre de lignes contenant la chaîne

grep -a : Traiter un fichier binaire comme s'il s'agissait de texte

grep -R, -r, --recursive : Lire tous les fichiers à l'intérieur de chaque répertoire, récursivement.

grep -v : afficher les lignes ne contenant pas la chaîne

grep -i : ignorer la casse.

sort : permet de trier par ordre alphabétique les lignes d'un fichier.

sort -b : Ignorer les blancs en début de ligne pendant la recherche de la clé de tri sur chaque ligne.

sort -d : Trier dans l'ordre des répertoires téléphoniques

sort -f : Considérer les minuscules comme leur équivalent en majuscule pendant le tri.

head : afficher les premières lignes

head -n N, --lines N : Afficher les N premières lignes.

head -q, --quiet, --silent : Ne pas afficher les en-têtes mentionnant les noms de fichiers.

head -v, --verbose : Toujours afficher les en-têtes mentionnant les noms de fichiers.

tail : afficher les dernières lignes

tail -n N, --lines N : Afficher les N dernières lignes.

tail -q, --quiet, --silent : Ne jamais afficher l'en-tête avec le nom du fichier

tail -v, --verbose : Toujours afficher l'en-tête avec le nom du fichier

diff : permet de comparer deux fichiers

diff -a : Traiter tous les fichiers comme du texte, et les comparer ligne-à-ligne, même s'ils semblent contenir des données binaires.

diff -b : Ne pas tenir compte des différences concernant des espaces blancs.

diff -B : Ne pas tenir compte des différences qui concernent des lignes blanches.

diff --brief : Indiquer seulement si les fichiers diffèrent, pas les différences elles-mêmes.

find : permet de rechercher un fichier dans une hiérarchie de répertoires.

-name : recherche sur le nom du fichier.

-perm : recherche sur les droits d'accès du fichier.

-links : recherche sur le nombre de liens du fichier.

-user : recherche sur le propriétaire du fichier.

-group : recherche sur le groupe auquel appartient le fichier.

-type : recherche sur le type (d=répertoire, c=caractère, f=fichier normal).

-size : recherche sur la taille du fichier en nombre de blocs (1 bloc=512 octets).

-atime : recherche par date de dernier accès en lecture du fichier.

-mtime : recherche par date de dernière modification du fichier.

-ctime : recherche par date de création du fichier.

chmod : permet de modifier les droits aux fichiers.

umask : définir les droits d'accès par défaut pour les nouveaux fichiers

id : affiche les informations d'identification de l'utilisateur.

whoami : affiche le nom de l'utilisateur.

users : affiche les noms de tous les utilisateurs connectés sur le système.

who : comme users mais avec plus d'informations.

passwd : change le mot de passe actuel.

groups : affiche les groupes auxquels l'utilisateur appartient.

newgrp groupname : changement de groupe.

su username : changement d'identité de l'utilisateur.

lastlog : date de la dernière connexion.

chown : changer le propriétaire d'un fichier

chgrp : changer le groupe d'un fichier

tr : Transposer ou éliminer des caractères substitution ou suppression de caractères sélectionnés.

-d : suppression des caractères sélectionnés

-s : 'aaaaa' dans chaîne1 -> 'a' dans chaîne2

-c : remplace jeu1 par son complément (tous les caractères n'appartenant pas à chaîne1)

uniq : Élimine les lignes dupliquées dans un fichier trié par défaut.

-d : affiche seulement les lignes dupliquées

-u : affiche seulement les lignes non dupliquées (par défaut)

-c : donne le nombre d'exemplaires de chaque ligne

cut : Supprime une partie de chaque ligne d'un fichier par défaut :

-d séparateur : le séparateur des champs

-f listes de champs : les champs séparés.

-c : Sélection sur le rang du caractère.

-b : Sélection sur le no d'octet.

-s (avec -f) : supprime les lignes vides

paste : Regroupe les lignes du même n° de différents fichiers

Les mécanismes de redirections et de tube

< Redirection de l'entrée standard

> Redirection de la sortie standard

>> Redirection de la sortie standard à la fin

| Echange de données entre deux commandes

2> Redirection des erreurs d'une commande

Les caractères spéciaux :

- Les noms de fichiers

* Remplace une suite de caractères

! Négation

? Remplace un caractère quelconque

[Liste] Remplace un des caractères de la liste

- Le contenu de fichier

^ Début de la ligne

\$ Fin de la ligne

. Remplace un caractère quelconque

[Liste] Remplace un des caractères de la liste

[^] Négation

→ **Résolutions :**

Q1/

```
user=`cat /etc/passwd | grep -w $1 | cut -d : -f 1`  
  
if [ -z $user ]  
then  
    echo "Cette UID ne figure pas dans la liste"  
else  
    echo $user  
fi
```

Q2/

```
#!/bin/bash  
  
echo "Entrer votre string : "  
read ch  
  
echo `echo $ch | tr [:lower:] [:upper:]`
```

Q3/

```
#!/bin/bash  
cpt=`ps -f -u $USER | tail -n +2 |wc -l`  
echo "Le nombre de vos processus actif : $cpt"
```

Q4/

```
$ cat >> emsi_rabat  
#!/bin/bash  
  
while true  
do  
    echo "Vous êtes a l'EMSI RABAT"  
    sleep 120  
done  
ctrl-D  
$ chmod u+x emsi_rabat
```

Q5/ \$./emsi_rabat&

Q6/ ps

Q7/ ps -f
kill -2 PID

Q8/ kill -19 PID

Q9/

```
$ cat >> process
#!/bin/bash
echo `ps -f -u $USER | tr -s " " | cut -d " " -f 8 |tail -n +2 | sort | uniq`
```

Q10/

```
#!/bin/bash
echo `find / -size +$1 -atime +$2`
```

Q11/

```
#!/bin/bash
if [ -e $1 ]
then
    datelm=`ls -l $1 | tr -s " " | cut -d " " -f 6,7,8`
    datelm="#" $datelm"
    `echo $datelm >> $1`
else
    echo "Le fichier n'existe pas"
fi
```

Q12/

```
#!/bin/bash
cpt=1
if [ -z $1 ]
then
    var="Hello $USER"
else
    var="Hello "
    for i in $@
    do
        if [ $cpt -eq 1 ]
        then
            var="$var $i"
        elif [ $cpt -lt $# ]
        then
            var="$var, $i"
        elif [ $# -ge 3 ]
        then
            var="$var and $i"
        else
            var="$var , $i"
        fi
        cpt=`expr $cpt + 1`
    done
fi
var="$var !"
echo $var
```

Mme EL HALOUI

M. EL HADRI

EMSI

04/02/2020

311R

Nom & Prénom :

Classe :

Examen UNIX

(Documents autorisés, Durée 2h)

1.

- a) Cocher dans la liste suivante les distributions Linux :
- | | | | | |
|---------------------------------------|---------------------------------|---------------------------------|---------------------------------|-------------------------------|
| <input type="checkbox"/> Open Solaris | <input type="checkbox"/> Ultrix | <input type="checkbox"/> GNU | <input type="checkbox"/> Xorg | <input type="checkbox"/> Slax |
| <input type="checkbox"/> Mint | <input type="checkbox"/> CentOS | <input type="checkbox"/> Redhat | <input type="checkbox"/> Debian | <input type="checkbox"/> AIX |
- b) Trois des noms suivants sont des répertoires se trouvant à la racine du système de fichiers Linux
- | | | |
|------------------------------|---------------------------------|---------------------------------------|
| <input type="checkbox"/> dev | <input type="checkbox"/> system | <input type="checkbox"/> boot |
| <input type="checkbox"/> run | <input type="checkbox"/> users | <input type="checkbox"/> applications |
- c) Trois des noms suivants sont des environnements de bureau :
- | | | |
|---------------------------------|----------------------------------|-------------------------------|
| <input type="checkbox"/> KDE | <input type="checkbox"/> Mate | <input type="checkbox"/> X11 |
| <input type="checkbox"/> Wyland | <input type="checkbox"/> SystemX | <input type="checkbox"/> XFCE |
- d) Trois des propositions suivantes sont des applications trouvées dans la plupart des installations Linux
- | | | |
|-----------------------------------|--------------------------------------|--|
| <input type="checkbox"/> Facebook | <input type="checkbox"/> LibreOffice | <input type="checkbox"/> Google Chrome |
| <input type="checkbox"/> Firefox | <input type="checkbox"/> Mailexp | <input type="checkbox"/> Terminal |
- e) Quatre des propositions suivantes sont des shells qu'on peut utiliser sous GNU-Linux ?
- | | | |
|-------------------------------------|---|-----------------------------------|
| <input type="checkbox"/> Born Shell | <input type="checkbox"/> Bourne Again Shell | <input type="checkbox"/> Z Shell |
| <input type="checkbox"/> Korn Shell | <input type="checkbox"/> C shell | <input type="checkbox"/> Keloxell |
- f) Deux des commandes suivantes redirigent les sorties standard et erreur vers le fichier *out.txt*
- | | |
|---|---|
| <input type="checkbox"/> command >out.txt 2>out.txt | <input type="checkbox"/> command 1&2> out.txt |
| <input type="checkbox"/> command >out.txt 2>&1 | <input type="checkbox"/> command &> out.txt |
| <input type="checkbox"/> command 2+1>out.txt | <input type="checkbox"/> command >& out.txt |
- g) Peut-on exécuter un fichier ayant la permission 766 si on n'est ni propriétaire ni *root*
- ☐ oui ☐ non
- h) Quelle est l'option de la commande *mv* permettant de renommer un répertoire
- ☐ -r ☐ -f ☐ -i ☐ --rename ☐ aucune option spécifique
- i) Laquelle parmi les commandes suivantes permet de copier uniquement les fichiers du répertoire *rep1* sous *rep2* ?
- | | |
|---|--|
| <input type="checkbox"/> cp -f rep1 rep2 | <input type="checkbox"/> cp -f rep1/* rep2 2>/dev/null |
| <input type="checkbox"/> cp -fR rep1/* rep2 | <input type="checkbox"/> cp -r rep1 rep2 2>/dev/null |
- j) Lesquelles parmi les commandes suivantes permettent d'afficher la liste des sous-répertoires se trouvant directement sous la Home Directory :
- | | |
|---|-----------------------------------|
| <input type="checkbox"/> find ~ -type d | <input type="checkbox"/> echo ~/* |
| <input type="checkbox"/> ls -d ~/* | <input type="checkbox"/> ls -d ~/ |
- k) Combien de lignes sont-ils affichés par défaut par la commande *tail*

- l) Que affiche la commande : `head -n2 f1 f2`
(f1 et f2 étant deux fichiers textes se trouvant sous le répertoire courant)
- ☐ Les deux premières lignes de chaque fichier passé en argument
 - ☐ Uniquement les deux premières lignes du fichier f1
 - ☐ Un message d'erreur
- m) Que affiche la commande : `head -2 f1 <f2`
(f1 et f2 étant deux fichiers textes se trouvant sous le répertoire courant)
- ☐ Les deux premières lignes de f1 suivies par les deux premières lignes de f2
 - ☐ Uniquement les deux premières lignes du fichier f1
 - ☐ Un message d'erreur
- n) Laquelle parmi les lignes de commandes suivantes exécute command2 uniquement si command1 réussit
- | | |
|--|---|
| <input type="checkbox"/> <code>command1 command2</code> | <input type="checkbox"/> <code>command1 & command2</code> |
| <input type="checkbox"/> <code>command1 && command2</code> | <input type="checkbox"/> <code>command1 command2</code> |
- o) La commande suivante :
- ```
rm -f f1
rm: cannot remove 'f1': Permission denied
```
- échoue parce qu'on :
- |                                                                       |                                                            |
|-----------------------------------------------------------------------|------------------------------------------------------------|
| <input type="checkbox"/> n'a pas le droit w sur le répertoire courant | <input type="checkbox"/> n'est pas administrateur          |
| <input type="checkbox"/> n'a pas le droit w sur le fichier f1         | <input type="checkbox"/> n'est propriétaire du fichiers f1 |
- p) Deux des commandes suivantes permettent d'arrêter le système ?
- |                                                       |                                                   |
|-------------------------------------------------------|---------------------------------------------------|
| <input type="checkbox"/> <code>shutdown -h now</code> | <input type="checkbox"/> <code>halt-sys</code>    |
| <input type="checkbox"/> <code>poweroff</code>        | <input type="checkbox"/> <code>kill system</code> |
- q) Cocher dans la liste suivante les lignes de commandes syntaxiquement incorrectes ?
- |                                                          |                                                   |
|----------------------------------------------------------|---------------------------------------------------|
| <input type="checkbox"/> <code>cp -l ./-l</code>         | <input type="checkbox"/> <code>cp \* .</code>     |
| <input type="checkbox"/> <code>cp -i 'rep/x' -f .</code> | <input type="checkbox"/> <code>cp a* ~/ -R</code> |
- r) Deux des commandes suivantes permettent d'afficher la liste des fichiers se trouvant sous /bin dont le nom commence par la lettre 'a' :
- |                                                  |                                                         |
|--------------------------------------------------|---------------------------------------------------------|
| <input type="checkbox"/> <code>ls /bin a*</code> | <input type="checkbox"/> <code>ls /bin/a*</code>        |
| <input type="checkbox"/> <code>ls a* /bin</code> | <input type="checkbox"/> <code>(cd /bin ; ls a*)</code> |
- s) Laquelle des commandes suivantes permet d'afficher la taille et l'espace disponible dans la partition racine :
- |                                            |                                               |
|--------------------------------------------|-----------------------------------------------|
| <input type="checkbox"/> <code>free</code> | <input type="checkbox"/> <code>df /</code>    |
| <input type="checkbox"/> <code>du /</code> | <input type="checkbox"/> <code>fdisk /</code> |
- t) Dans le but de changer son mot de passe, un utilisateur ( qui n'est pas administrateur) peut :
- ☐ utiliser la commande `passwd` même s'il ne se rappelle pas de son mot de passe actuel
  - ☐ utiliser la commande `passwd` uniquement s'il connaît son mot de passe actuel
  - ☐ modifier directement son fichier `.password` se trouvant dans sa *Home Directory*
- u) La commande : `find a*`
- ☐ est syntaxiquement incorrecte
  - ☐ cherche et affiche tous les fichiers dont le nom commence par la lettre 'a' se trouvant dans l'arborescence du répertoire courant
  - ☐ affiche un message d'erreur si aucun fichier ou répertoire commençant par 'a' n'a été trouvé directement dans répertoire courant
- v) Voulant exécuter la commande « `sudo ls /root` » un utilisateur reçoit le message suivant (après avoir introduit son mot de passe) :
- test n'apparaît pas dans le fichier sudoers. Cet événement sera signalé.*



Cela veut dire :

- ☐ l'utilisateur en question n'est pas autorisé à utiliser la commande *sudo*
- ☐ le mot de passe introduit ne correspondait pas
- ☐ la commande « `sudo ls /root` » est syntaxiquement incorrecte

2. Que fait chacune des commandes suivantes :

a) `mkdir -p ~/a/b`

.....  
.....

b) `cat f g > h`

.....  
.....

c) `rm -f *~`

.....  
.....

d) `pkill -STOP vmstat`

.....  
.....

e) `ls -lait /bin`

.....  
.....

f) `chmod g-w,o=g prog`

.....  
.....

g) `echo 'aaa42bSS6a' | tr -s '[:lower:]' '[:upper:]'`

.....  
.....

3. Sur le système Fedora 31, donner les lignes de commandes permettant de :

a) Afficher le manuel de la commande `less`

.....

b) Lister tous les fichiers du répertoire `/bin` dont le nom se termine par un chiffre

.....

- c) Copier toutes les entrées (fichiers et sous-répertoires) du répertoire courant qui se terminent par '.c' ou '.cpp' sous /run/media/emsi/cle\_usb  
.....
- d) Déplacez le fichier ba et le répertoire bl se trouvant dans le répertoire courant vers le répertoire tmp de la Home Directory  
.....
- e) Créer dans votre Home Directory un lien d'appelant etc vers le répertoire /etc  
.....
- f) Afficher les dix plus gros fichiers de /usr/bin  
.....
- g) Afficher la liste de tous les fichiers se trouvant dans la Home Directory qui n'ont pas été accédés dans les 30 derniers jours  
.....
- h) Chercher dans /usr/share toutes les images (jpg, gif, png) dont la taille dépasse 1Mo  
.....
- i) Récupérer dans le fichier tata les lignes 11 à 17 du fichier toto.  
.....
- j) Tuer tous les processus appartenant à l'utilisateur zayd.  
.....
- k) Afficher la liste des utilisateurs ayant au moins un processus s'exécutant dans le système. La liste doit faire apparaître aussi le nombre de processus correspondant à chaque utilisateur.  
.....

Indication : Utiliser pour cela les commandes : ps aux, cut, sort, tail et uniq  
Ci-après un exemple de sortie de la commande : ps aux

| USER    | PID  | %CPU | %MEM | VSZ   | RSS   | TTY | STAT | START | TIME | COMMAND              |
|---------|------|------|------|-------|-------|-----|------|-------|------|----------------------|
| root    | 1    | 0.1  | 0.1  | 1672  | 268   | ?   | Ss   | 11:54 | 0:07 | systemd              |
| geoclue | 107  | 0.0  | 0.1  | 1454  | 1228  | ?   | Ssl  | 11:55 | 0:00 | /usr/libexec/geoclue |
| dnsmasq | 1373 | 0.0  | 0.0  | 2236  | 2532  | ?   | S    | 11:55 | 0:00 | /usr/sbin/dnsmasq    |
| root    | 1451 | 0.0  | 0.1  | 14064 | 12128 | ?   | Sl   | 11:55 | 0:00 | gdm-session-worker   |
| fedx    | 1469 | 0.0  | 0.1  | 2024  | 1264  | ?   | Ss   | 11:55 | 0:00 | systemd --user       |

Les colonnes sont séparées par des espaces

- l) Comment copier le contenu du fichier /etc/passwd dans un nouveau fichier nommé users, en remplaçant les caractères « : » par des « | ».  
.....
- m) Changer les modes d'accès de tous les fichiers qui ont une extension '.txt' en rw--w----





|                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                   |                                |                   |
|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|-------------------|
| <b>Matière :</b> Système d'Exploitation 3                                                             |                                                                                                                                                                                                                                                                                                                                                                   | <b>Filière :</b> IIR           | <b>Niveau :</b> 3 |
| <b>Date de l'épreuve :</b> 04/02/2021                                                                 |                                                                                                                                                                                                                                                                                                                                                                   | <b>Durée de l'épreuve :</b> 2H |                   |
| <b>Documents autorisés</b><br>Oui <input checked="" type="checkbox"/><br>Non <input type="checkbox"/> | <b>Matériels autorisés :</b><br>Oui <input type="checkbox"/> : (Préciser).....AUCUN.....<br>Non <input checked="" type="checkbox"/>                                                                                                                                                                                                                               |                                |                   |
| Pour ne pas oublier, inscrivez votre nom et prénom avant de commencer:                                | <b>Nom :</b> .....<br><b>Prénom :</b> .....<br><b>Classe :</b> .....                                                                                                                                                                                                                                                                                              |                                |                   |
| Directives Générales :                                                                                | <ul style="list-style-type: none"><li>- <b>Répondre directement sur le questionnaire</b></li><li>- Les réponses doivent être brèves, précises et clairement présentées</li><li>- Calculatrices, téléphones portables et objets connectés <b>NON</b> autorisés</li><li>- Interdiction de toute forme d'échange verbal, écrit ou matériel entre candidats</li></ul> |                                |                   |

## 1. Entourer la bonne réponse (4pts)

1.1. Quatre des propositions suivantes sont des Shells qu'on peut utiliser sous GNU-Linux, lesquelles ?

- A. Korn Shell
- B. Born Shell
- C. Keloxell
- D. BourneAgain Shell
- E. Z Shell
- F. C Shell

1.2. Peut-on exécuter un fichier ayant la permission 766 si on n'est ni propriétaire ni root ?

- A. Oui
- B. Non

1.3. Lorsqu'un processus enfant se termine avant la fermeture du processus parent, lequel des éléments suivants est vrai ?

- A. Le processus fils devient orphelin
- B. Le processus parent disparaît
- C. Si le processus parent ne gère pas SIGCHLD, le processus fils devient un zombie
- D. Aucune de ces réponses

1.4. Quelle est l'option de la commande mv permettant de renommer un répertoire

- A. -r
- B. -f
- C. -i
- D. --rename
- E. Aucune option spécifique

1.5. Qu'est ce qui est chargé en mémoire au démarrage du système ?

- A. Noyau
- B. Shell
- C. Commandes
- D. Script

1.6. Trois des propositions suivantes sont des applications trouvées dans la plupart des installations Linux

- A. Facebook
- B. LibreOffice
- C. Google Chrome
- D. Firefox
- E. Mailexp
- F. Terminal

1.7. Laquelle des valeurs suivantes pour la colonne STAT de la commande « ps » n'est pas vraie ?

- A. Statut R signifie Running
- B. Statut S signifie Sleeping
- C. Statut E signifie Exited
- D. Statut Z signifie Zombie

1.8. La commande suivante : `rm -f f1`

`rm: cannot remove 'f1': Permission denied`

Échoue parce qu'on :

- A. N'a pas le droit w sur le répertoire courant
- B. N'est pas administrateur
- C. N'a pas le droit w sur le fichier f1
- D. N'est pas propriétaire du fichiers f1

## 2. Exercices d'interprétation : (8pts)

2.1. Lançons trois commandes en pipe de la manière suivante :

`$ command1 | command2 | command3`

2.1.1. Si `command1` correspond à la commande '`ps`' et qu'aucun processus n'est lancé dans le shell courant avant ce pipe, quel serait le nombre de processus qui seront listés par `ps` ? (N.B. bien sûr ils ne seront pas affichés sur la sortie standard puisque la sortie de `command1` passe en pipe vers l'entrée de `command2`)

Réponse : (0.5pts) .....

Justification : (0.5pts)

2.1.2. Supposons que `command1` est remplacée par `find . -name 'fich*' -print` et que son affichage sur la console est le suivant :

```
/home/moi/tp/fich1.txt
find: '/home/moi/tp/fich2': Permission non accordée
/home/moi/tp/fich3
find: '/home/moi/tp/fich4': Permission non accordée
```

2.1.2.1. Que signifie le point juste après `find` ? (0.5pts)

2.1.2.2. Donnez le résultat final affiché sur la sortie standard si **command2** est remplacée par **grep 'fich'** et **command3** est remplacée par **wc -l** :

Réponse :(0.5pts) .....

Justification :(0.5pts)

2.1.2.3. Remplaçons cette fois-ci la **command2** par **cut -d 'o' -f 2** et **command3** par **tr '/e' 'da'** ; Quel serait le résultat affiché ?

Réponse :(0.5pts)

Justification :(0.5pts)

2.1.2.4. Quel serait le contenu du fichier **fich5.txt** si l'on ajoute une redirection à la fin du pipe de la manière suivante :

**\$ find . -name 'fich\*' -print | grep 'fich' | wc -l 2> fich5.txt**

.....(0.5pts)

2.2. Cochez les droits par défaut des fichiers et des dossiers créés après chacune des commandes suivantes :(1.5pts)

|           | Fichiers simples         |                          |                          |                          |                          |                          |                          |                          |                          | Répertoires              |                          |                          |                          |                          |                          |                          |                          |                          |
|-----------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
|           | r                        | w                        | x                        | r                        | w                        | x                        | r                        | w                        | x                        | r                        | w                        | x                        | r                        | w                        | x                        | r                        | w                        | x                        |
| umask 002 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| umask 222 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| uamsk 020 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

2.2.1. Que représentent les lettres suivantes :(0.75pts)

**r**: pour les fichiers ordinaires .....; pour les dossiers :.....

**w**: pour les fichiers ordinaires .....; pour les dossiers :.....

**x**: pour les fichiers ordinaires .....; pour les dossiers :.....

2.2.2. Pourquoi y-a-t'il trois combinaisons **rwX** pour chaque fichier ? Qu'est ce qu'elles représentent ?(0.75pts)

2.3. Soit un fichier **fich6** qui a les droits d'accès suivants **-rw-r--r--**

2.3.1. Est-ce que **fich6** est un fichier ordinaire ou bien un dossier ? (0.5pts)

2.3.2. Que se passera-t-il si on exécute la commande suivante : (0.5pts)

**\$ chmod a+x fich6**

### 3. Exercices de production : (8pts)

3.1. Donnez la commande qui permettra de changer les droits de **fich7.sh** en **-rwx-r--r--** quelque soit ses droits d'accès initiaux. (0.5pts)

.....

3.2. Donnez la commande qui changera le propriétaire de **fich7.sh** en **user2** (0.5pts)

.....

3.3. Comment afficher uniquement les fichiers du répertoire courant qui sont des liens symboliques ? (1pts)

.....

3.4. Comment afficher le nombre d'utilisateurs déclarés dans le fichier **/etc/passwd** ayant **zsh** comme Shell de login. (1pts)

.....

3.5. Donner la syntaxe pour compter le nombre de lignes qui contiennent le mot « *file* » dans la page de man de la commande **less**(1pts)

.....

3.6. Donner la syntaxe pour afficher la liste des utilisateurs dont l'**UID** varie entre 700 et 999 dans **/etc/passwd**. (1pts)

.....

3.7. Chercher dans **/usr/share** toutes les images (jpg, gif, png) dont la taille dépasse 1Mo (1pts)

.....

3.8. Récupérer dans le fichier **tata** les lignes 21 à 37 du fichier **toto**. (1pts)

.....

3.9. Comment copier le contenu du fichier **/etc/passwd** dans un nouveau fichier nommé **users**, en remplaçant les caractères « : » par des « | ». (1pts)

.....



**Examen Final SE LINUX : Durée 2 h 00**

*Note : L'énoncé de l'épreuve se compose de 4 pages.*

**N. B. Répondez dans le tableau vers la fin de l'épreuve.**

Nom : .....

Prénom : .....

Note ...../20

Classe : .....

**DOCUMENT A RENDRE**

**Partie 1 : QCM**

1) Parmi la liste suivante, cochez les distributions Linux que vous connaissez :

- ☐ Ubuntu      ☐ BeOS      ☐ Mint      ☐ Arch      ☐ Solaris  
☐ Open Suse      ☐ CentOS      ☐ Kali      ☐ Debian      ☐ OSX

2) Les commandes suivantes permettent de lancer le script bash: myscript se trouvant dans le répertoire courant. L'une d'elles exige que le fichier myscript ait le droit d'accès x

- ☐ source ./myscript  
☐ bash myscript  
☐ ./myscript  
☐ . myscript

3) Deux des commandes suivantes permettent d'afficher la liste des fichiers se trouvant sous /lib et dont le nom commence par la lettre 'a' :

- ☐ ls /lib a\*      ☐ ls /lib/a\*  
☐ ls a\* /lib      ☐ (cd /lib ; ls a\*)

4) Un parmi les choix suivants ne correspond pas à l'expression régulière : ^w[!m-r].\*f\$

- ☐ woof      ☐ warproof  
☐ windproof      ☐ weatherproof

5) L'une des commandes suivantes affiche la liste des fichiers du répertoire courant modifiés dans les dernières 24h :

- ☐ find . -type f -mmin -24  
☐ find . -type f -mtime -1  
☐ find . -type f -mtime  
☐ find . -type -cmin 120

6) Cochez dans la liste suivante la commande indiquant au système de ne plus octroyer (positionner) la permission « w » pour les catégories « group » et « other » concernant les nouveaux fichiers et répertoires :

- ☐ chmod go-w \*      ☐ umask -w  
☐ umask 022      ☐ umask go-w \*

- 7) Sous vi, pour sauvegarder le fichier et quitter l'éditeur on peut taper en mode commande :
- ☐ :s
  - ☐ :sq
  - ☐ :wq
  - ☐ <CTR +S> <CTR + Q>
  - ☐ :x
- 8) Si la valeur « umask » est « 027 », quelle sera la valeur octale du mode d'accès accordé par défaut aux nouveaux scripts créés ?
- ☐ 027
  - ☐ 640
  - ☐ 750
  - ☐ 664
- 9) Quelle commande affichera les informations de processus en cours en temps réel ?
- ☐ ps current
  - ☐ top
  - ☐ ps -aux
  - ☐ process
- 10) La commande : `x=x*2`
- ☐ Affecte à x le double de sa valeur si la variable x existe et déclaré de type entier
  - ☐ Toujours affecte à la variable x la chaîne 'x\*2'
  - ☐ Affiche un message d'erreur si la variable x n'existe pas ou sa valeur est nulle
  - ☐ Affecte 0 à x si la variable x n'existe pas ou sa valeur est nulle
- 11) Pour tester dans un script bash la présence d'arguments, on peut écrire :
- ☐ if ( \$# != 0 ) then ... fi
  - ☐ if [ \$# != 0 ] then ... fi
  - ☐ if [[ \$# -ne 0 ]]; then ... fi
  - ☐ if [ -n \$\* ]; then ... fi
- 12) Existe-t-il une différence entre le résultat des deux commandes ci-après :
- echo 'Je suis sous \$HOME' et echo "Je suis sous \$HOME"
- ☐ OUI
  - ☐ NON

## **Partie 2 :**

### **1- Que fait chacune des commandes suivantes :**

a- `ls -l /bin/[aA]*`

.....

b- `find /tmp/Disc -name *.sql -o -name *.c -mtime 0 -size +1k`

.....

c- `mv *.c ..`

.....

d- `rm -f ~/*# 2>> F_LOG`

.....



e- :25,30s/abc/ABC/gc (Sous vi)

f- cp ../premier site ../copie

g- ps -aux | tail -n +2 | cut -d " " -f1 | sort | uniq -c

Indication : Ci-après la sortie de la commande : ps -aux

( USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND )

h- ps -ax | grep : | wc -l > proc

## 2- Sur le système Linux, donner les lignes de commandes permettant de :

a- Dans quel cas ne peut-on pas faire de lien physique ? Que faut-il faire ?

b- Quel est l'effet de chmod sur un lien ?

c- Comment afficher le nombre de lignes contiennent le mot «*Chaine*» dans la page de man de grep ?

d- Lister le contenu de tous les répertoires existants sous la racine, en faisant afficher les informations techniques des fichiers (on n'oubliera pas d'afficher les fichiers cachés)

e- Afficher le fichier /etc/passwd en remplaçant les caractères / par des X.

f- Après avoir tapé la commande la commande ps, vous obtenez l'affichage suivant :

| UID | PID  | PPID | PRI | NI | VSZ  | RSS  | STAT | TTY   | TIME  | COMMAND     |
|-----|------|------|-----|----|------|------|------|-------|-------|-------------|
| 70  | 903  | 900  | 10  | 0  | 2676 | 1580 | S    | pts/0 | 0 :00 | bash        |
| 70  | 924  | 903  | 8   | 0  | 6016 | 3764 | S    | pts/0 | 0 :02 | xfig TP1_1. |
| 50  | 928  | 903  | 9   | 0  | 8660 | 6040 | R    | pts/0 | 0 :14 | lyx TP1.lyx |
| 0   | 1310 | 903  | 16  | 0  | 3032 | 1164 | S    | pts/0 | 0 :00 | ps al       |

Quel est le numéro du processus xfig ? Quelle est la commande correspondant à son père ?

g- Créer un fichier Hebdo (attention, sans extension .txt), qui contient la première ligne de tous les fichiers d'extension .txt et, à la suite, leur dernière ligne, le tout sans en-tête

h- Faire afficher uniquement les lignes 5, 6 et 7 de fichier Hebdo

- i- Afficher la liste des fichiers plus vieux que 3 jours situés sous votre répertoire de connexion.

- j- Comment afficher le nombre de fichiers n'appartenant pas à root dans /etc.

### **Parie 3 : Bash scripting**

- 1) Ecrire un script qui teste si le nombre donné en argument est inférieur à 10, supérieur à 100, ou compris entre 10 et 100.

- 2) Écrivez un script qui détermine le nombre de fichiers ordinaires puis des répertoires existant dans un répertoire passé en argument.

- 3) Ecrire un script qui affiche le login et le Shell d'un utilisateur déclaré dans le fichier **/etc/passwd** dont on fournit l'UID en argument.

*Bon courage*