

Contrôle continu - 7/11/2023

Durée 1h30. Documents autorisés : une feuille A4 R/V notes de CM et de TP.

Nom et prénom :

Groupe :

Exercice 1 Que font les commandes suivantes ?

a) `find / -name "*.c" -maxdepth 2 -print 2>/dev/null`

Affiche la liste des fichiers dont le nom finit par ".c" à une profondeur maximale de 2 à partir du répertoire '/'.
2>/dev/null redirige les messages d'erreur vers le fichier /dev/null.

b) `find . -name "[Xx]*" -print 2>/dev/null`

Affiche la liste des fichiers dont le nom commence par la lettre 'X' ou par la lettre 'x' à toutes les profondeurs à partir du répertoire courant. 2>/dev/null redirige les messages d'erreur vers le fichier /dev/null.

c) `echo 1 2 3 4 5 6 | tr 123 456`

affiche 4 5 6 4 5 6

d) `chmod 731 foo.txt`

si le fichier `foo.txt` a les droits `-r-x--x---`.

Le fichier `foo.txt` a désormais les droits suivants `-rwx-wx--x`. Le propriétaire peut lire, écrire et exécuter, les membres du groupe peuvent écrire et exécuter et les autres peuvent seulement exécuter.

Exercice 2 Écrire une ligne de commande qui affiche la liste du contenu du répertoire `/mydir` tel que chaque ligne contient la taille du fichier et son nom, et tel que la liste soit triée par taille décroissante des fichiers.

NB : la taille du fichier est donnée par le 5^{ème} champ de la commande `ls -l`.

```
ls -l /mydir | tr -s ' ' | cut -d ' ' -f5,9 | sort -r -n
```

Exercice 3 Soit le fichier `/etc/passwd` dont les trois premières lignes sont comme suit :

```
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
```

On peut remarquer que les champs sont séparés par le caractère ':'.

Écrire une ligne de commande qui crée un fichier nommé `passwd_bis` dans le répertoire `/tmp` où chaque ligne contient le nom de login et l'UID correspondant respectivement aux champs numéros 1 et 3 du fichier `/etc/passwd` et dans lequel le séparateur ':' est remplacé par un espace.

```
cut -d: -f1,3 /etc/passwd | tr ":" " " > /tmp/passwd_bis
```

Exercice 4 Écrire un script qui, pour un répertoire donné dont le nom se trouve dans une variable nommée `rep`, affiche le nom de tous les fichiers ordinaires ainsi que la date de leur dernière modification (champs numéros 6, 7 et 8 de la commande `ls -l`). Il est demandé que le nom des fichiers soient affichés sans le répertoire comme préfixe..

```
for i in $(ls -l $rep | tail +2 | tr -s ' ' | cut -d ' ' -f9 )
do
  if [ -f $rep/$i ]
  then
    ls -l $rep/$i | tr -s ' ' | cut -d ' ' -f6,7,8,9
  fi
done
```

Exercice 5 Écrire un script qui permet de calculer et d'afficher la factorielle d'un nombre. On suppose que la valeur de ce nombre est contenue dans la variable `nombre`.

```
nombre=...
factorielle=1
for i in $( seq 2 $nombre); do
    factorielle=$((factorielle*$i))
done
echo "factorielle de $nombre = $factorielle"
```

Exercice 6 Écrire un script shell qui compare deux répertoires dont les noms sont donnés dans les variables `rep1` et `rep2` et qui affiche le répertoire qui contient le plus de fichiers exécutable par son propriétaire.

```
#!/bin/bash
rep1=$1
rep2=$2
c1=0
for f in $(ls $rep1); do
    if [ -f $rep1/$f ]; then
        car=$(ls -l $rep1/$f | cut -c 4)
        if [ $car = "x" ]; then
            c1=$((c1+1))
        fi
    fi
done
c2=0
for f in $(ls $rep2); do
    if [ -f $rep2/$f ]; then
        car=$(ls -l $rep2/$f | cut -c 4)
        if [ $car = "x" ]; then
            c2=$((c2+1))
        fi
    fi
done
if [ $c1 -gt $c2 ]; then
    echo $rep1
else
    echo $rep2
fi

### version avec grep
c1=$(ls -l $rep1 | tail -n +2 | cut -c 4 | grep 'x' | wc -l)
c2=$(ls -l $rep2 | tail -n +2 | cut -c 4 | grep 'x' | wc -l)
if [ $c1 -gt $c2 ]; then
    echo $rep1
else
    echo $rep2
fi
```

Exercice 7 Écrire un script shell qui recopie les fichiers ordinaires dont la taille est inférieure à 1Mo du répertoire indiqué par la variable `rep1` dans le répertoire indiqué par la variable `rep2`.

```
#!/bin/bash

rep1=$1
rep2=$2
Mo=$((1024*1024))
for f in $(ls $rep1); do
    if [ -f $rep1/$f ]; then
        taille=$(ls -l $rep1/$f | tr -s ' ' | cut -d' ' -f 5)

        if [ $taille -ge $Mo ]; then
            cp $rep1/$f $rep2/
        fi
    fi
done
```