

Contrôle continu

Durée 1h30. Documents autorisés : notes de CM et de TP.

Nom et prénom :

Groupe :

Exercice 1

1. La commande `ls -l *.txt` retourne un résultat comme suit :

```
-rw-r--r-- 1 toto user 2838 oct.  9 11:07 nez.txt
-rw-rw-r-- 1 titi user  21 oct.  9 11:33 test.txt
-rw-r--r-- 1 toto user  106 sept. 26 2019 tp2.txt
```

Écrire une ligne de commande pour afficher les fichiers dans l'ordre croissant de leur taille. Par exemple, le résultat sera comme suit :

```
-rw-rw-r-- 1 titi user 21 oct. 9 11:33 test.txt
-rw-r--r-- 1 toto user 106 sept. 26 2019 tp2.txt
-rw-r--r-- 1 toto user 2838 oct. 9 11:07 nez.txt
```

```
ls -al *.txt | tr -s ' ' | sort -n -k5
```

2. La commande `ls -l` dans le répertoire courant produit cet affichage :

```
-rwxr--r-- 1 tutu user 33 19 oct 14:26 hello.sh
-rw-r--r-- 1 toto user 2838 oct.  9 11:07 nez.txt
-rw-r--r-- 1 toto staff 95198 27 nov 2019 svn.pdf
-rw-rw-r-- 1 titi user 21 oct.  9 11:33 test.txt
-rw-r--r-- 1 toto user 106 sept. 26 2019 tp2.txt
-rwxr-xr-x 1 titi staff 3940 1 oct 16:07 tp3.tex
```

Écrire une ligne de commande qui affiche les fichiers du répertoire courant dont la dernière mise à jour s'est effectuée dans l'année en cours. On peut remarquer que seuls ces fichiers ont leur 8^{ème} champ qui indique une heure sous la forme `hh:mm`. Par exemple, pour ce répertoire courant, on obtient :

```
hello.sh
nez.txt
test.txt
tp3.tex
```

```
ls -l | tr -s ' ' | grep ":" | cut -d ' ' -f 9
```

Exercice 2

1. Quelle est la variable d'environnement qui indique le nom de l'utilisateur de la session courante ?

```
$USER
```

2. Utilisez cette variable pour afficher un message de cette forme (ici la variable a pour valeur `toto`) :

```
Bonjour toto !
```

```
echo Bonjour $USER !
```

Exercice 3 Écrire une ligne de commande pour compter le nombre de fichiers dont le nom commence par un chiffre et qui ont l'extension `.txt`.

```
ls [0-9]*.txt | wc -l
```

Exercice 4 Commenter la commande

```
find . -name '*.py' -print
```

Cherche, dans toute l'arborescence sous le répertoire courant, et affiche le chemin des fichiers dont le nom se termine par `.py`

Exercice 5 Écrire une ligne de commande pour compter le nombre de propriétaires différents des fichiers du répertoire courant ayant l'extension `.txt`. Rappelons que dans chaque ligne donnée par la commande `ls -l`, la troisième colonne indique le propriétaire du fichier. Par exemple sur le cas de la question 1.1, le nombre sera 2.

```
ls -l *.txt | tr -s ' ' | cut -d' ' -f3 | uniq | wc -l
```

Exercice 6

1. Supposons que dans le répertoire courant se trouve un fichier `foo.txt` dont les droits sont `-r-x--x---`. À quoi correspondent ces droits ?

Le propriétaire peut lire, exécuter le fichier, les membres de son groupe peuvent seulement exécuter et les autres utilisateurs n'ont aucun droit sur le fichier. Le droit de modification n'est donné à personne, même pas le propriétaire.

2. Quels seront les droits de ce fichier après l'exécution de la commande `chmod 731 foo.txt` ?

Le propriétaire peut lire, écrire et exécuter. Les membres de son groupe peuvent écrire et exécuter et les autres utilisateurs peuvent seulement exécuter.

3. Après l'exécution de cette commande, qui peut changer les droits du fichier `foo.txt` ?

Seul le propriétaire du fichier peut en changer les droits.

4. Donner deux commandes possibles pour enlever les droits d'exécution au groupe à qui appartient le fichier.

```
chmod g-x foo.txt ou chmod 721 foo.txt
```

5. Comment vérifier que le changement de droits s'est opéré avec succès ?

```
ls -l foo.txt et on doit voir des droits comme : -rwx-w--x alors qu'ils étaient : -rwx-wx--x
```

Exercice 7 Écrire un script shell qui crée un fichier `texte_inv.txt` qui a le même contenu que le fichier `texte.txt` mais dans lequel les lignes sont prises dans l'ordre inverse. Par exemple, si le contenu du fichier `texte.txt` est :

```
bonjour tout le monde
et
bonsoir tout le monde
```

alors le fichier `texte_inv.txt` aura le contenu :

```
bonsoir tout le monde
et
bonjour tout le monde
```

Si besoin, on peut utiliser la commande `seq` pour générer une liste décroissante d'entiers, par exemple `seq 5 -1 1` génère la liste `5 4 3 2 1`.

```
#!/bin/bash
fich="texte.txt"
resInv="texte_inv.txt"
taille=$(cat $fich | wc -l)
for i in $(seq $taille -1 1); do
    cat $fich | head -n $i | tail -1 >> $resInv
done
```

Exercice 8 Écrire un script shell qui crée un fichier `texte_rep.txt` à partir d'un fichier `texte.txt` où une ligne numéro i est répétée i fois. On obtient par exemple le résultat suivant :

<u>Contenu du fichier <code>texte.txt</code></u>	<u>Contenu du fichier <code>texte_rep.txt</code></u>
matin	matin
après-midi	après-midi
soir	après-midi
	soir
	soir
	soir

```
#!/bin/bash
fich="texte.txt"
rep="texte_rep.txt"
for i in $(seq $taille); do
  ligne=$(cat $fich | head -n $i | tail -1)
  for j in $(seq $i); do
    echo $ligne >> $rep
  done
done
```

Exercice 9 Écrire un script shell `listedir.sh` qui permet d'afficher le contenu d'un répertoire en séparant les fichiers et les sous-répertoires. Le nom du répertoire est donné au script en ligne de commande. Exemple d'utilisation :

```
./listedir.sh /etc
affiche :
```

```
### fichiers dans /etc
/etc/aliases.db
/etc/auto_home
/etc/auto_master
/etc/bashrc
### répertoires dans /etc
/etc/apache2
/etc/defaults
```

```
#!/bin/bash
rep=$1
FICHIERS=""
REPERTOIRES=""
for nom in $(ls $rep); do
  if [ -f "$rep/$nom" ]; then
    FICHIERS="$FICHIERS\n$nom"
  elif [ -d "$rep/$nom" ]; then
    REPERTOIRES="$REPERTOIRES\n$nom"
  fi
done

echo "### fichiers dans $rep"
echo $FICHIERS
echo "### repertoires dans $rep"
echo $REPERTOIRES
```
