

TP: Programmation Shell

Exercice N°1 :

Ecrivez un script qui convertit en minutes et secondes un temps exprimé en secondes (le temps est passé en paramètre).

Exercice N°2 :

Ecrivez un script qui affiche, pour tous les utilisateurs passés en arguments du script, le nom de l'utilisateur, le répertoire de connexion et le shell.

Exercice N°3 :

Ecrivez un script qui enregistre dans un fichier les lignes saisies au clavier, et qui affiche le nombre de lignes qui ont été enregistrées.

Exercice N°4 :

Créez la commande copier. La commande reçoit en argument deux noms de fichiers, la source et la destination. Le script se termine et affiche un message d'erreur si l'une des conditions suivantes est réalisée :

- Le nombre d'arguments est incorrect.
- Le fichier source n'existe pas ou il n'est pas copiable (pas d'accès en lecture).
- Le fichier source n'est pas un fichier ordinaire.
- Le fichier destination existe.
- Le répertoire de destination, que l'on peut connaître avec la commande `dirname`, n'est pas accessible en écriture.
- La copie a échoué.

Quelques Notion sur : La gestion de fichiers

L'objectif de la gestion de fichier est de savoir lire des lignes ou des mots, créer ou mettre à jour des fichiers, accéder à un fichier à l'aide de son descripteur.

- la mise à jour d'un fichier par la redirection `>>`
- la redirection avec `<<`
- la lecture d'un fichier dans une boucle avec la commande `read`

Exemple de création d'une fiche :

```
#!/bin/sh
echo "nom ? \c"
read nom
echo "telephone ? \c"
read telephone
echo "${nom}:${telephone}" >> agenda
```

```
#more agenda
toto:23-11-71-43
titi:98-33-65-78
```

Suppression et modification d'enregistrements dans un fichier

On peut réaliser cette tâche grâce à des commandes tels que : grep, tail, head ou sed

Exemple de suppression

On souhaite supprimer la ou les ligne(s) qui ne concernent pas toto.

```
# cat agenda
toto:23-11-71-43
titi:98-33-65-78
```

```
#grep -v "^toto :" agenda > /tmp/tempo$$
#mv /tmp/tempo$$ agenda
```

Exemple de modification

On peut utiliser la commande grep (vu précédemment) pour ce genre de tâche mais la commande sed est adaptée à ce type d'opération. Pour plus d'informations faites man sed

```
# sed `s/titi/tata/` agenda > tmp/tempo$$
# mv /tmp/tempo$$ agenda
```

```
# cat agenda
```

```
tata:98-33-65-78
```

Exercice N°5 :

Réaliser un programme de gestion d'agenda téléphonique. Chaque enregistrement contient le nom d'une personne et son numéro de téléphone. Les deux champs sont séparés par une virgule. Le programme affiche le menu suivant :

1. Ajouter une nouvelle fiche
 2. Rechercher une fiche (avec une partie du nom)
 3. Détruire une fiche
 4. Modifier une fiche
 5. Lister l'annuaire
 6. Fin
- Votre choix ? :

Correction

EXERCICE 1:

```
#!/bin/bash
#arith.h

minutes=`expr $1 \/ 60`
secondes=`expr $1 \% 60`
echo "Résultat $minutes minutes et $secondes secondes"
```

EXERCICE 2:

```
#!/bin/bash

for nom
do
grep "^$nom" /etc/passwd | cut -d: -f1,6,7
done
```

EXERCICE 3:

```
#!/bin/bash
#arith.h
integer i=0
while
echo "Entrer un texte ou <FIN> pour quitter:\c"
read ligne
do
if [ "$ligne" = "fin" -o "$ligne" = "FIN" ]
then
break
fi
echo "$ligne" >> fichligne
let i=i+1
done
echo "le nombre de lignes saisies: $i"
```

EXERCICE 4:

```
#!/bin/bash
# copier proprement

if [ $# -ne 2 ]
then
echo "ERREUR: nombre arguments incorrect"
exit 1
fi

if [ ! -r $1 ]
then
echo "ERREUR: la source n'est pas accessible"
exit 2
fi

if [ ! -f $1 ]
then
echo "ERREUR: la source n'est pas un fichier"
exit 3
```

```

fi

if [ ! -e $2 ]
then
echo "ERREUR: la destination n'existe pas"
exit 4
fi

rep_dst=`dirname $2`
if [ -w $rep_dst ]
then
echo "ERREUR: le répertoire n'est pas accessible"
exit 5
fi

if cp $1 $2 > /dev/null
then :
else
echo "ERREUR: la copie a echouée"
exit 6
fi

```

EXERCICE 5:

```

#!/bin/sh
#agenda

ajouter () {
echo "-----AJOUT D'UNE NOUVELLE FICHE-----"
echo "Nom ?"
read nom
echo "Tel ?"
read tel
echo "$nom, $tel" >> agenda.txt
}

rechercher() {
echo "-----rechercher une fiche-----"
echo "Nom (ou une partie du nom)?"
read nom
grep "$nom" agenda.txt
}

lister() {
echo "-----lister les fiches-----"
sort agenda.txt | more
}

detruire() {
echo "-----destruction d'une fiche-----"
echo "Nom (ou une partie du nom)?"
read nom
grep -v "$nom" agenda.txt > buffer
mv buffer agenda.txt
}

modifier() {
echo "-----modifier une fiche-----"
echo "Nom ? "
read nom
if grep "^$nom," agenda.txt

```

```

then
    echo "Nouveau telephone ?"
    read tel
    grep -v "^$nom" agenda.txt > buffer
    echo "$nom, $tel" >> buffer
    mv buffer agenda.txt
fi
}

pause(){
echo "appuyer sur return pour continuer"
read bouton
}

#-----
while :
do
echo "1- Ajouter une nouvelle fiche"
echo "2- Rechercher une fiche(entrer une partie du nom) "
echo "3- Detruire une fiche"
echo "4- Modifier une fiche"
echo "5- Lister l'annuaire"
echo "6- Fin"
echo "choix ?"
read choix
case "$choix" in
1) ajouter ;;
2) rechercher
    pause
    ;;
3) detruire ;;
4) modifier ;;
5) lister
    pause
    ;;
6) exit 1 ;;
*) echo "choix incorrect" ;;
esac
done

```