

Exercice : « Exemples de programmes shell »

- 1- « La boucle while » Écrire un programme shell qui affiche les arguments du programme, dans l'ordre d'apparition (1er argument en premier). Si le programme n'a aucun argument, afficher « sans argument ».
- Indication : Utilisez la commande shift.

```
#!/bin/sh
if [ $# -eq 0 ]; then
    echo "sans argument"
    exit 0
fi
while [ $# -gt 0 ]; do
    echo $1
    shift
done
```

- 2- Écrire un programme shell qui affiche ligne par ligne le contenu d'un fichier dont le nom est fourni en paramètre...

```
#!/bin/sh
exec < $1 # redirection de l'entrée standard
while read ligne ; do
    echo ">$ligne"
done
```

- 3- Écrire un programme shell qui affiche tous les sous-répertoires du répertoire courant, en utilisant une boucle.

```
#!/bin/sh
for rep in *; do    # pour chaque fichier "rep" du repertoire courant
    if [ -d "$rep" ]; then    # si rep est un repertoire
        echo $rep    # affiche rep
    fi
done
```

- 4- «Les conditionnelles imbriquées » Écrire un programme shell qui **accepte 2 paramètres**. Le premier paramètre est +r, -r, +w ou -w, et le deuxième paramètre spécifie une extension de nom de fichiers. En fonction de la valeur du premier paramètre, le programme modifiera les droits du groupe de tous les fichiers du répertoire courant dont l'extension est égale au deuxième paramètre. Pour contrôle, avant chaque modification des droits sur un fichier, le programme affichera le nom du fichier.

Exemple d'utilisation (le script s'appelle droitsfichiers) :
droitsfichiers +r .html

Proposez une nouvelle version de ce programme capable d'accepter trois paramètres. Les trois paramètres spécifient alors :

1. le répertoire dans lequel sont contenus les fichiers dont les droits seront modifiés,
2. les modifications des droits pour le groupe,
3. l'extension des fichiers concernés.

Exemples d'utilisation (le script s'appelle droitsfichiers) :

droitsfichiers .. -w .dat

droitsfichiers perso -r .txt

```
#!/bin/sh
if [ $# -eq 2 ]; then # si le nombre d'arguments est égal à 2 :
    rep="." # variable rep vaut le repertoire courant = "."
    droit=$1 # variable droit vaut le premier argument $1
    ext=$2 # variable ext vaut le deuxième argument $2
elif [ $# -eq 3 ]; then # si le nombre d'arguments est égal à 3 :
    rep=$1 # variable rep vaut le premier argument $1
    droit=$2 # variable droit vaut le deuxième argument $2
    ext=$3 # variable ext vaut le troisième argument $3
else
    echo "erreur : donnez 2 ou 3 arguments" 1>&2
    exit 1
fi

for fich in "${rep}"/*"${ext}"; do # pour chaque fichier "fich" dans
rep/*ext:
    chmod g$droit "$fich" # changer le droit du groupe "g" sur "fich"
    echo $fich
done
```

5- On voudrait écrire un script permettant de facilement changer de façon systématique l'extension d'une série de fichiers. On souhaite par exemple renommer tous les fichiers .htm du répertoire courant en .html

```
#!/bin/sh
if [ $# -ne 2 ]; then
    echo "erreur : donnez 2 arguments." 1>&2
    exit 1
fi
vieux=$1
nouveau=$2
for fich in *"${vieux}"; do
    base=`basename "$fich" "${vieux}"`
    mv "$fich" "$base$nouveau"
if [ $? -ne 0 ]; then
    echo "je ne peux pas renommer \"$fich\" en \"$base$nouveau\"" 1>&2
fi
done
```

La commande basename est ici utilisée pour récupérer dans la variable base le nom du fichier sans son extension représentée par la variable \$vieux. Le nouveau nom du fichier (avec sa nouvelle extension) est donc facile à reconstruire : c'est le nom du fichier base, suivi de la nouvelle extension \$nouveau.

- 6- Écrire un script shell qui prend en paramètre un fichier formaté comme indiqué ci-dessous et affiche tous les livres d'un auteur passé en paramètre.

Exemple de fichier (bibliotheque):

Leon Shklar & Rich Rosen @ Web Application Architecture

Bernard Desgraupes @ LaTeX : apprentissage, guide et référence

Vipul Kashyap & Leon Shklar @ Real World Semantic Web Applications

Ernest E. Rothman & Brian Jepson & Rich Rosen @ Mac OS X for Unix Geeks (Leopard)

Bernard Desgraupes @ Passeport Pour Unicode

Exemple d'appel (en gras l'appel, en italique la réponse du script) :

./script bibliotheque Desgraupes

Desgraupes

Passeport Pour Unicode

LaTeX : apprentissage, guide et référence

./script bibliotheque Shklar

Shklar

Web Application Architecture

Real World Semantic Web Applications

#/bin/bash

PASTROUVE=1

affiche () {

echo \$2

cat \$1 | while read ligne

do

A=`echo \${ligne} | grep -i \$2`

if [\$? -eq 0]

then

PASTROUVE=0

echo "\${ligne}/@"*

fi

done

#BONUS

if [\${PASTROUVE} -eq 1]

then

echo "pas de livre trouvé"

fi

}

7- Écrire un script shell qui affiche un paramètre sur 3 de la ligne de commande.

```
# Tant qu'il y a des arguments à traiter
while [ $# -gt 0 ]
do
    # j'affiche le premier argument
    echo $1
    # Puis j'en passe deux
    shift
    shift
    shift
done
```

8- Écrivez un script qui teste si un nom proposé par l'utilisateur est un login ou pas tant qu'il le souhaite. Si le login existe, le script doit montrer le(s) groupe(s) auquel il appartient.

```
#!/bin/bash
# Script qui teste si la saisie est un user et affiche ses groupes
# Boucle jusqu'à demande de sortie
```

```
echo "Saisir un utilisateur (q pour quitter) : "
while read user
do
    if [ $user = "q" ]
    then
        echo "Byebye..."
        exit 0
    fi
    if [ -z `cat /etc/passwd | grep "^$user:"` ]
    then
        echo "$user n'est pas un utilisateur"
    else
        echo "groupes de " `groups $user`
    fi
    echo "Saisir un utilisateur (q pour quitter) : "
done
```