Bash – un shell courant

Commandes courantes :

- ls (lister un répertoire)
- cd (aller dans un répertoire)
- mkdir (créer un répertoire)
- cat (afficher un fichier)
- less (visionner un fichier)
- mv (renommer un fichier)

Les quotes

message='Bonjour tout le monde'
echo 'Le message est : $message'

message='Bonjour tout le monde'
echo "Le message est : $message"

message=`pwd`
echo "Vous êtes dans le dossier $message"

Enregistrer une variable

read prenom nom

echo $prenom $nom

read -p ‘entrez votre nom :’ nom

echo $nom

Scripts

#!/bin/bash

(#! appellé “sha-bang”)

```
#!/bin/bash
let "a = 5"
let "b = 2"
let "c = a + b"
echo $c
```

Les paramètres

./variables.sh param1 param2 param3

$# : contient le nombre de paramètres ;

$0 : contient le nom du script exécuté (ici ./variables.sh) ;

$1 : contient le premier paramètre ;

$2 : contient le second paramètre ;

… ;

```
#!/bin/bash
echo "Vous avez lancé $0, il y a $# paramètres"
echo "Le paramètre 1 est $1"
```

Exercice

- Écrire un script avec votre nom comme paramètre, qui écrit : Bonjour « votre nom »

Les tableaux

```
#!/bin/bash
tableau=('valeur0' 'valeur1' 'valeur2')
tableau[5]='valeur5'
echo ${tableau[*]}
```

Reprendre le script de tout à l'heure, mais en rajoutant un 2ème paramètre, copier les paramètres dans un tableau, puis afficher le tableau (Bonjour « prénom » « nom »!)

Les tests - if

```
#!/bin/bash

read -p ‘qui es-tu ?’ nom
if [ $nom = "Bruno" ]; then
        echo "Salut Bruno !"
else
        echo "Je ne vous connais pas!"
fi
```

Les tests - elif

```
if [ test ]
then
	echo "Le premier test a été vérifié"
elif [ autre_test ]
then
	echo "Le second test a été vérifié"
elif [ encore_autre_test ]
then
	echo "Le troisième test a été vérifié"
else
	echo "Aucun des tests précédents n'a été vérifié"
fi
```

Les tests - !

```
if [ $nom != "Bruno" ]
then
        echo "tu ne t’appelle pas Bruno"
fi


if [ ! -e fichier ]
then
        echo "Le fichier n'existe pas"
fi
```

Les tests - && et ||

&& : signifie « et » ; (ou -a)

|| : signifie « ou ». (ou -o)

```
#!/bin/bash
read -p "Si vous etes d'accord entrez o ou oui : " reponse
if [ ! $reponse = "o"] && [ ! $reponse = "oui" ]; then
    echo "Non, je ne suis pas d'accord !"
else
    echo "Oui, je suis d'accord"
fi
```

alternative

```
#!/bin/bash
read -p "Si vous etes d'accord entrez o ou
oui : " reponse
if [ ! $reponse = "o" -a ! $reponse = "oui" ];
then
    echo "Non, je ne suis pas d'accord !"
else
    echo "Oui, je suis d'accord"
fi
```

Conditions multiples

```
#!/bin/bash
case $1 in
        "Chien" | "Chat" | "Souris")
                echo "C'est un mammifère"
                ;;
        "Moineau" | "Pigeon")
                echo "C'est un oiseau"
                ;;
        *)
                echo "Je ne sais pas ce que c'est"
                ;;
esac
```

Les boucles

```
#!/bin/bash
while [ -z $reponse ] || [ $reponse != 'oui' ]
do
        read -p 'Dites oui : ' reponse
done
```

Boucler sur une valeur

```
#!/bin/bash
for variable in 'valeur1' 'valeur2' 'valeur3'
do
	echo "La variable vaut $variable"
done
```

ou bien

```
#!/bin/bash
for animal in 'chien' 'souris' 'moineau'
do
	echo "Animal en cours d'analyse : $animal"
done
```

En pratique

```
#!/bin/bash
for fichier in `ls`
do
        mv $fichier $fichier-old
done
```

Les boucles – exercice

faire un script de multirenommage qui ne va renommer que les fichiers correspondant au paramètre saisi. si aucun paramètre saisi, il faudra le demander

Les boucles – for classique

```
#!/bin/bash
for i in `seq 1 10`;
# écrire seq 1 2 10 pour aller de 2 en 2
do
        echo $i
done
```