

Initiation à la programmation shell

- Tubes et redirection
- Créer un script
- Syntaxe du shell
 - variables
 - test
 - structures de contrôle
 - arithmétique

Tubes et redirections

- Rediriger la sortie :
`ls -l > lesfichiers.txt`
- Ajouter à un fichier :
`ls -l >> lesfichiers.txt`
- Rediriger les erreurs :
`mount /dev/fd0 /mnt/floppy 2>erreurs.txt`
- Rediriger une entrée :
`less < lesfichiers.txt`
- Les tubes :
`ps | sort | less`

Programmation interactive

```
$ for fichier in *.jpg
> do
> cible=$(echo $fichier|sed -e "s/jpg/jpeg/")
> mv $fichier $cible
>done
```

Script minimal

```
#!/bin/bash
# Fichier : renomme.sh
for fic in *.jpg
do
  cible=$(echo $fic|sed -e "s/jpg/jpeg/")
  mv $fic $cible
done
```

- Rendre le script exécutable :
`chmod u+x renomme.sh`
- L'exécuter :
`. renomme.sh`

Syntaxe du shell bash

Les variables

```
$ texte = Bonjour
$ echo $texte
Bonjour
$ texte = "Mesdames, Messieurs"
$ echo $texte
Mesdames, Messieurs
```

```
$ texte2 = "Bonjour $texte"
$ echo $texte2
Bonjour Mesdames, Messieurs
$ echo 'Bonjour $texte'
Bonjour $texte
$ texte = 7+8
$ echo $texte
7+8
```

Variables d'environnement

Variable	Description
\$HOME	Répertoire principal du compte
\$PATH	Liste des répertoires où chercher les exécutables
\$PS1	Invite principale du shell
\$PS2	Invite secondaire du shell
\$IFS	Séparateur de champ de saisie. Par défaut, espace, tabulation, retour à la ligne
\$0	Nom du script
\$num	Paramètre numéro <code>num</code> du script
\$#	Nombre de paramètres du script

Test booléens

Utilisation : [-f fichier]

Test	Résultat
ch1 = ch2	Vrai si les chaînes sont identiques
ch1 != ch2	Vrai si les chaînes sont différentes
-n ch	Vrai si la chaîne est non vide
ex1 -eq ex2	Vrai si les deux expressions ont même valeur
ex1 -ne ex2	Vrai si les deux expressions n'ont pas même valeur
! ex	Vrai si ex est faux (négation)
ex1 -ge ex2	Vrai si $ex1 \geq ex2$
ex1 -le ex2	Vrai si $ex1 \leq ex2$

Cours bash - p.9

Test booléens - Fichiers

Test	Résultat
-d fic	Vrai si fic est un répertoire
-e fic	Vrai si fic existe
-f fic	Vrai si fic est un fichier ordinaire
-r fic	Vrai si fic peut être lu
-w fic	Vrai si fic peut être écrit
-x fic	Vrai si fic peut être exécuté
-s fic	Vrai la taille de fic est non nulle

Cours bash - p.10

Structures de contrôle - if

```
if test
then
    instructions
elif test
then
    instruction
else
    instruction
fi
```

Cours bash - p.11

Structures de contrôle - Boucles

```
# For :

for var in valeurs
do
    instructions
done

# While :

while test
do
    instructions
done
```

Cours bash - p.12

Exemple - while

```
#!/bin/bash

foo=1

while [ "$foo" -le 20 ]
do
    echo "Ligne numéro $foo"
    foo=$((foo+1))
done
```

Cours bash - p.13

Structures de contrôle - case

```
case variable in
    modèle ) instructions ;;
    ...
esac

# Exemple :

echo 'Sommes nous le matin ?'
read moment

case moment in
    oui|OUI ) echo "Bonjour !" ;;
    non|NON ) echo "Bonsoir !" ;;
    * ) echo "Réponse incorrecte";;
esac
```

Cours bash - p.14