

Introduction au Bash

IFT-1214

Université de Montréal

2

Introduction

Principes

- Plus qu'une invite de commande
- Langage de programmation
- Interprété
 - Non compilé
 - Traduit au fur et a mesure de son exécution
- Permet
 - L'encapsulation de suites de commandes dans un même programme
 - Simplification de certaines tâches quotidiennes ou répétitives

Introduction

Documentation

- Learning the bash shell (2nd édition)
par : Cameron Newham & Bill Rosenblatt
Collection Unix Shell Programming
Editions : O'REILLY
- Pages des manuels Unix
 - man bash
 - info bash
- Divers sites web
 - Mots clefs google : "bash programming"

3

Introduction

Plan

- Principes de base
- Structures de contrôle
- Guillemets
- Arithmétique et booléens
- Lecture de l'entrée standard
- Fonctions
- Interception des signaux
- Passage d'arguments et valeurs de retour
- Paramétrage de l'environnement

4

Principes de Base

Introduction au Bash

6

Principes de Base

Squelette

- Un script est un fichier texte dont la première ligne contient le chemin du shell par lequel il doit être exécuté

– Fichier **script.sh** :

```
#!/bin/bash
...
code du script
...
```

- Le nom du fichier n'a pas d'importance, par contre, il doit être exécutable :

– Chmod : `biotope% chmod +x script.sh`

Principes de Base

Premier Script

- L'exécution se fait en tapant le nom du fichier comme l'appel d'une commande. Par contre :
 - Si l'on est pas dans le répertoire dans lequel est le fichier, Il faut donner le chemin (relatif ou complet)
 - Même si l'on est dans le même répertoire, suivant la configuration de l'environnement, il faut parfois précéder le nom du fichier de **./**

script.sh

```
#!/bin/bash
echo Premier script en bash
```

```
biotope% chmod +x script.sh
biotope% ./script.sh
Premier script en bash
```

7

Principes de Base

Variables

- Une variable peut contenir un nombre ou une chaîne de caractères. La déclaration se fait de la façon suivante : **variable=contenu**
- La lecture d'une variable se fait avec **\$variable**
- Une chaîne avec espaces se met entre " "

script.sh

```
#!/bin/bash
salutation="Premier script en bash"
echo $salutation
```

```
biotope% ./script.sh
Premier script en bash
```

8

Principes de Base

Autres

- Une variable peut être utilisée n'importe quand
- Un commentaire commence par un **#**

```
script.sh
```

```
#!/bin/bash
# ceci est une ligne de commentaire

compteur=3 # initialisation de la variable compteur à 3
echo Valeur du compteur : $compteur # affichage
```

```
biotope% ./script.sh
Valeur du compteur : 3
```

Principes de Base

Encapsulation

```
#!/bin/bash
echo Script de nettoyage des fichiers inutiles
echo Suppression des fichier de sauvegarde ...
rm -f *.bak
echo Suppression des fichier de traces ...
rm -f *.log
echo Fait.
```

```
biotope% ./script.sh
Script de nettoyage des fichiers inutiles
Suppression des fichier de sauvegarde ...
Suppression des fichier de traces ...
Fait.
```

Structures de Contrôle

Introduction au Bash

Structures de Contrôle

If

```
#!/bin/bash

if [ -f /tmp/archive.tar ];

then
    # L'archive existe, on décompresse
    tar -xvf /tmp/archive.tar
    echo Terminé.

else
    # L'archive n'existe pas, on donne un message d'erreur
    echo Pas de fichier archive.tar dans /tmp
fi
```

Structures de Contrôle

If

• Tests sur les fichiers

- Existence : -e
- Répertoire : -d
- Fichier : -f
- Non vide : -s
- Lisible : -r
- Modifiable : -w
- Exécutable : -x

```
#!/bin/bash

if [ -x script.sh ];
then
    echo Fichier exécutable
else
    echo Fichier non exécutable
fi
```

```
biotope% ./script.sh
Fichier exécutable
```

13

Structures de Contrôle

If

• Comparaison d'entiers

- Egalité : -eq
x = y : [x -eq y]
- Inégalité : -ne
x ≠ y : [x -ne y]
- Supériorité : -gt
x > y : [x -gt y]
- Infériorité : -lt
x < y : [x -lt y]

```
#!/bin/bash
var1=15
var2=9
if [ $var1 -gt $var2 ];
then
    echo $var1 supérieur à $var2
else
    echo $var1 inférieur ou égal à $var2
fi
```

```
biotope% ./script.sh
15 supérieur à 9
```

14

Structures de Contrôle

If

• Comparaison d'entiers

- Egalité : -eq
x = y : [x -eq y]
- Inégalité : -ne
x ≠ y : [x -ne y]
- Supériorité : -gt
x > y : [x -gt y]
- Infériorité : -lt
x < y : [x -lt y]

```
#!/bin/bash
var1=15
var2=21
if [ $var1 -gt $var2 ];
then
    echo $var1 supérieur à $var2
else
    echo $var1 inférieur ou égal à $var2
fi
```

```
biotope% ./script.sh
15 inférieur ou égal à 21
```

15

Structures de Contrôle

If

```
#!/bin/bash
var1=15
var2=21
if [ $var1 -gt $var2 ];
then
    echo $var1 supérieur à $var2
else
    if [ $var1 -eq $var2 ];
    then
        echo $var1 est égal à $var2
    else
        echo $var1 est inférieur à $var2
    fi
fi
```

```
biotope% ./script.sh
15 est inférieur à 21
```

16