Unix : Utilisation et programmation

Pr. Hajar IGUER,
hajar.iguer@uic.ac.ma

Unix®
Operating System

Installer un nouveau Shell

- La commande à utiliser:
  - sudo apt-get install ksh
- Une fois le nouveau Shell installé, pour l'utiliser on utilise la commande suivante:
  - chsh //change shell
  - /bin/ksh
- Débogage d'un fichier script
  - bash –x fichier.sh

Variables: Affectation de valeur

- Pour l'affectation des valeurs aux variables du Shell, on utilise:
  - nom_variable=valeur
  - var1= test
  - var2 = 45
  - MY_Name ='' G.Info''
- Précéder une apostrophe ' d'un antislash\ afin de l'afficher comme valeur de la variable
- Pour le renvoi de la valeur d'une variable:
  - $nom_variable

Afficher la valeur d'une variable

- Utiliser la commande echo pour afficher un message ou la valeur d'une variable
  - echo $msg //affiche la valeur d'une variable
  - echo –e $msg //affiche le message avec les retours à la ligne
- Il faut utiliser le symbole \n pour le retour à la ligne

Afficher la valeur d'une variable

- En utilisant les simples quotes ‘ ’
  - msg=‘test’ ; echo ‘Afficher $msg’
  - Pas d’affichage de contenu de la variable
  - En utilisant les double quotes " "
  - msg=‘test’ echo "Afficher $msg "
  - Affiche du contenu de la variable après analyse
- En utilisant les back quotes ` `
  - msg=`ls`; echo "Afficher $msg"
  - Exécution de la commande et affichage du résultat

Lire une variable

- La commande read donne la possibilité à l'utilisateur de saisir du texte
  - read var
  - read var1 var2
  - read –p 'Saisissez votre texte' var3
  - read –p 'Saisissez votre texte' –n 3 var4
  - read –p 'Saisissez votre mdp' –s pass

Variable d'environnement

- Une variable n'existe et n'est utilisée que dans le script où elle a été définie.
- Une variable d'environnement sont des variables utilisables dans n'importe quel programme.
- Elles sont aussi appelés variables globales
- Pour afficher la liste des variables, on utilise:
  - env

Variable Path: Exemple

- La variable $PATH:
  - Afin d'exécuter un ficher shell depuis n'importe quel emplacement, il faut rajouter son emplacement au niveau de la variable d'environnement $Path ou le mettre dans un des emplacements déjà référencés
- $ echo $PATH
  - /bin:/usr/bin:/usr/local/bin:/usr/bin/X11:/usr/X11R6/bin:/home/rossignol_d/bin

Variables d'environnement

- EDITOR: le nom de l'éditeur de texte
- PWD: chemin courant
- OLDPWD: l'ancien chemin courant
- SHELL: nom du Shell
- Expr: Evaluation d'expressions numériques
- HOSTNAME: Indique le nom de l'ordinateur

Variables d'environnement

- CDPATH: liste des chemins d'accès utilisés par la commande cd lors d'un changement de répertoire en relatif
- TMOUT: nombre de secondes d'inactivité provoquant un logout
- HOME: indique le répertoire d'accueil et correspond à l'argument par défaut de la commande cd

Déclaration variable

- Pour ajouter une variable d'environnement:
  - nom_var=valeur
  - export nom_var
  - export nom_var=valeur
  - declare –x nom_var

Alias

- Les alias permettent de:
  - Donner une équivalence entre une ligne de commande Linux et une chaine de caractères.
  - Définir des abréviations ou de nouveaux noms pour des commandes.
  - alias nom_alias=‘ commandes ’

- La commande alias permet d’afficher la liste des alias définis.

Variables des paramètres

- Les scripts bash peuvent accepter des paramètres. Pour appeler le script:
  - ./fich.sh param1 param2 param3

`$0`	Le nom de la commande (i.e. : du script)
`$1`, `$2`, etc.	Le premier, deuxième, etc, argument passés au script.
`$*`	La liste de tous les arguments passés au script.
`$#`	Le nombre d'arguments passés au script.
`$?`	Le code de retour de la dernière commande lancée.
`$!`	Le numéro de process de la dernière commande lancée en tâche de fond.
`$$`	Le numéro de process du shell lui-même.

Opérations mathématiques

- Afin d'effectuer des opérations mathématiques sur les variables d'un script, on utilise:
  - let a=7
  - let b=8
  - let c=a+b
  - echo $c

A=6
B=2
Expr $A*$B

Exemples

- $ vi test3

```
#!/bin/bash
# display user information from the system.
echo "User info for userid: $USER"
echo UID: $UID
echo HOME: $HOME
```

- $ vi test4

```
#!/bin/bash
# testing variables
days=10
guest="Katie"
echo "$guest checked in $days days ago"
days=5
guest="Jessica"
```

- $ vi test5

```
#!/bin/bash
# using the backtick character
testing=`date`
echo "The date and time are: " $testing
```

- $ vi test6

```
#!/bin/bash
# An example of using the expr command
var1=10
var2=20
var3=`expr $var2 / $var1`
echo The result is $var3
```

Exercice 1

- **Traduisez en script Shell l'algorithme suivant :**
- entier nb
- réel pht, ttva, pttc
- **Début**
- écrire "Entrez le prix hors taxes :"
- lire pht
- écrire "Entrez le nombre d'articles :"
- lire nb
- écrire "Entrez le taux de TVA :"
- lire ttva
- pttc ← nb * pht * (1 + ttva)
- écrire "Le prix toutes taxes est : ", pttc
- **Fin**

Exercice 2

- Ecrire un programme qui demande à l'utilisateur de taper **4** entiers et qui affiche leur moyenne.

- Refaire le même exercice en passant les 4 chiffres en paramètres du script.

Exercice 3

- Ecrivez un programme qui affiche la décomposition en base 10 d'un nombre de quatre chiffres lu au clavier. Par exemple, le nombre **4235** sera affiché sous la forme :
  - **5 + 3*10 + 2*10^2 + 4*10^3**
- Refaire le même programme en passant le chiffre à décomposer en paramètre du script