TP-LINUX: Premiers pas en script shell

Objectif : Automatiser des procédures de maintenance du système Linux
Utiliser un éditeur de texte (VI ou nano)
Utiliser la console pour des opérations de maintenance

Programmation SHELL-SCRIPT :

Le shell est un outil puissant et intègre un véritable langage de programmation.

Un SCRIPT shell est un programme qui enchaine des commandes du shell.

Outre les commandes que vous connaissez déjà (ls, cd, rm, mv, cp, whoami, pwd, mkdir, chmod, chown, useradd, userdel, groupadd, …), il y a également des possibilités de faire des actions différentes en fonction du résultat d'un TEST, ou de répéter une série d'instructions en BOUCLE.

Le shell autorise également l'utilisation de VARIABLES, c'est-à-dire des mémoires de stockage temporaire d'une information.

Vous retrouverez ces notions dans TOUS les langages de programmation (C, C#, C++, PHP, JavaScript, Python …)

Un SCRIPT shell est un fichier texte contenant des commandes exécutables.

On le créé avec un éditeur de texte (Notepad++, Gedit, Geany, ***vi*** ou ***nano***)

Il faut ensuite donner le droit d'exécution au script avec la commande *chmod,* par exemple avec l'option *+x*

Travail préparatoire :

Nous allons tester une série de commandes ayant comme objectif d'enregistrer une information sur un serveur externe. Pour l'instant, ce sera l'adresse IP du Raspberry et votre nom.

Première commande :

Pour obtenir ses adresses ip : On connait la commande : `ip a`

Cette commande est trop « bavarde ». Pour isoler l'adresse ip du réseau Box, on va « filtrer » pour ne garder que la partie qui nous intéresse : la ligne qui contient par exemple le début de l'adresse IP : 192.168.1.

```
ip a | grep 192.168.1.
```

Le symbole « | » qui s'obtient avec <ALT GR><6> est appelé « tuyau » (*pipe* en anglais). C'est lui qui permet de filtrer.

Testez cette commande.

C'est un bon début mais ce n'est pas suffisant.

Il faut encore filtrer : réduire les espaces, récupérer le 3e mot, recouper pour enlever le '/24' …

A la fin, cela donne :

```
ip a |  grep 192.168.1 | tr -s ' ' | cut -f3 -d' ' | cut -f1 -d'/'
```

Testez cette commande.

NB : Nous cherchons ici à montrer la puissance et la souplesse des commandes shell pour les opérations de maintenance automatisées. La commande « hostname » permet également de connaitre ses adresses ip : `hostname -I` (ou `hostname -i`). Cependant, si l'ordinateur dispose de plusieurs interfaces réseaux, il faudra également utiliser des commandes shell pour isoler l'adresse qui nous intéresse.

Création du Shell Script :

Sur la Console, créez un fichier nommé *enregistrement.bash* en tapant :

nano `enregistrement.bash` ou **vi** `enregistrement.bash`

Pour ***vi***, familiarisez-vous avec les commandes en utilisant la doc fournit en annexe.
Essayez par exemple d'ajouter du texte, enregistrer, rouvrir et modifier le texte.
Pour ***nano***, en bas de l'écran vous verrez les commandes (^ représente la touche CTRL).

Pour stocker notre adresse IP, nous utiliserons une variable, donc le début de notre script sera :

```
#!/bin/bash
#   Essai de script
#   Enregistrement automatique dans une base de donnees externe
#   TP SN Linux
#   Auteur :

adresse=$(ip a |  grep 192.168.1 | tr -s ' ' |cut -f3 -d' ' | cut -f1 -d/)
echo $adresse
```

Explications : En Shell Script, `adresse=$( … )` est un façon de créer une variable nommée « adresse » qui contiendra le **résultat** de la commande écrite dans les parenthèses.
Pour réutiliser cette variable, il faut ajouter le « $ » à son nom : `$adresse`
La commande « echo » permet d'afficher du texte ou le contenu d'une variable.
Vous remarquez qu'il n'y a pas besoin de la déclarer ni de préciser son type (comme en c++). C'est fréquent dans les langages de script.

TRAVAIL :
Ecrivez ce script dans le fichier nommé « enregistrement.bash » et testez le.

Rappels : Il faut lui mettre le droit « x » !!!!! (avec chmod)
et l'exécuter avec la commande `./enregistrement.bash`
Si tout va bien, l'adresse IP de votre Raspberry doit s'afficher.

Suite du script :
Le serveur externe possède une base de données SQL prévue pour ce TP et il n'attend que vous…

Pour accéder au serveur, il nous faut une application « cliente » :
Si vous pensez que c'est utile, mettez à jour la liste des magasins de téléchargement :
`sudo apt update`

Installer le client mysql : `sudo apt-get install mariadb-client`
A l'issue de cette installation, vous disposez de la commande shell « **mysql** » qui permet de se connecter à un serveur SQL (bases de données). Pour l'instant on n'a pas besoin d'en savoir plus …

Ensuite nous reprenons notre script :
Ajouter la ligne qui créée une variable qui contient le nom (hostname) de votre RaspBerry :
`message=$(hostname)`
Et une ligne qui crée une variable « temperature » qui servira plus tard.
`temperature=0`

Créer la variable qui contient le texte de la requête SQL à envoyer au serveur :

```
sql="INSERT INTO connexions VALUES (NULL, NULL, '$adresse', '$message', $temperature)"
```

Et enfin ajouter la commande qui contacte la base de données pour l'enregistrement, avec le nom du serveur, de l'utilisateur, le mot de passe, le nom de la base de données … et la requête :

```
mysql -h 192.168.1.254 -utpCron -pcrontab -D tpRaspCron -e "$sql"
```

Enregistrez et TESTEZ votre script.

Si aucune erreur n'apparait, vous pourrez afficher le contenu de la base de données du serveur par la commande shell :

```
mysql -h 192.168.1.254  -u tpCron -pcrontab -D tpRaspCron -e "SELECT * from connexions"
```

1. Mettez cette commande dans un autre shell-script.

ANNEXE

Utilisation de vi ou de vim (plus récent):

Vi est issu de l'informatique des premiers jours … donc sa manipulation a de quoi surprendre. Mais l'avantage de ce produit : il demande très peu de ressource machine, et il est sur toutes les machines Unix.

Vi connaît 2 modes : le mode **commande** et le mode **insertion**.

En mode commande, Vi attend une commande ou permet le déplacement sur un texte existant.

En mode insertion (commande **i**), on peut taper du texte.

On sort du mode insertion par la touche <ESC>

Commandes nécessaires à la survie :

:q	Quitter Vi sans enregistrer.	:q !	Pour ignorer les modifications.
:x	Quitter et enregistrer	:w	Enregistrer sans quitter
x	Effacer (couper) le caractère à droite du curseur		
i	Insérer du texte	o	Insérer une ligne en dessous
dd	couper la ligne	Y	copier la ligne
p	coller		

Il y en a bien sûr beaucoup d'autres …