**453---SYSTEME D'EXPLOITATION - e-learning - Séquence 12**

**Programmation de scripts – 1**

**Légende :**

Explications sur fond blanc

Exercice à accomplir sur fond rouge clair

Une ou plusieurs solutions possibles pour l’exécution de commande sur fond vert clair

Une solution possible pour la rédaction du script sur fond jaune clair

Un interpréteur de commandes ou « shell » permet certes d’activer des commandes de façon interactive, mais aussi d’exécuter des séquences programmées de commandes, ce que l’on appelle des « scripts ».

Dans un système Linux, le fonctionnement de nombreux programmes est d’ailleurs paramétré par des scripts d’initialisation.

Parmi les commandes, certaines sont internes à la programmation du shell (built-in), comme la commande « cd ». D’autres sont externes, c’est-à-dire qu’il s’agit de programmes ou de fichiers binaires exécutables (binaries), comme la commande « ls » (fichier « /bin/ls »).

Comme dans les autres langages de programmation, la programmation d’un script shell peut contenir des instructions de contrôle comme « if », « while », etc.

Par défaut, lorsqu’on exécute un script, un nouveau processus shell est activé (un processus shell fils de celui dans lequel l’exécution est lancée), à moins que l’exécution ne soit lancée avec la commande « . ».

La première ligne de la programmation d’un script shell doit être un « pseudo-commentaire », qui indique à la commande qui lance l’exécution du script que c’est par exemple un shell « bash » qu’elle doit activer pour exécuter le script ; dans ce cas, le pseudo-commentaire s’écrit : « #!/bin/bash ».

Syntaxe du shell :

#!/bin/bash Pseudo-commentaire, au début d’un script, qui indique à la commande qui active le script que c’est un shell bash qu’elle doit activer pour exécuter le script

#!/bin/sh Pseudo-commentaire, au début d’un script, qui indique à la commande qui active le script que c’est un shell sh qu’elle doit activer pour exécuter le script

# Annonce d’un commentaire

; Séparateur de commandes (comme un saut de ligne ou une tabulation)

& Séparateur de commandes, mais la commande qui le précède est exécutée en tâche de fond

. Exécution d’un script shell sans activation d’un nouveau processus shell (une telle activation constituant le comportement par défaut)

« ~/.profile » (ou « ~/.bash\_profile », selon les versions) est un fichier de script qui est exécuté automatiquement en début de session par le shell qui la contrôle.

Une variable d’un shell est une zone dans l’espace mémoire alloué par le système au processus shell. Elle est identifiée par un nom. Une valeur lui est affectée (stockée dans la zone). Tout processus, donc en particulier un shell, possède un espace mémoire dit d’environnement, contenant des variables d’environnement. Cet espace est hérité de processus shell père en processus shell fils.

**Quelques commandes :**

UPTIME(1)

NOM

uptime - Indiquer depuis quand le système a été mis en route

SYNOPSIS

uptime [options]

DESCRIPTION

uptime affiche sur une ligne les informations suivantes : l'heure actuelle, la durée

depuis laquelle le système fonctionne, le nombre d'utilisateurs actuellement connectés,

et la charge système moyenne pour les 1, 5, et 15 dernières minutes.

UNAME(1)

NOM

uname - Afficher des informations sur le système

SYNOPSIS

uname [OPTION]...

DESCRIPTION

Afficher certaines informations concernant le système.

Le comportement est identique si OPTION est omis ou -s est utilisée.

DATE(1)

NOM

date - Afficher ou configurer la date et l'heure du système

SYNOPSIS

date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMJJhhmm[[CC]AA][.ss]]

DESCRIPTION

Afficher dans le FORMAT indiqué ou configurer la date et l'heure du système.

BASH(1)

NOM

bash - Interpréteur de commandes GNU Bourne-Again SHell

SYNOPSIS

bash [options] [chaîne\_de\_commande | fichier]

DESCRIPTION

bash est un interpréteur de commandes (shell) compatible sh qui exécute les

commandes lues depuis l'entrée standard ou depuis un fichier. bash inclut aussi

des fonctionnalités utiles des interpréteurs de commandes ksh et csh.

CHMOD(1)

NOM

chmod - Modifier les bits de comportement (droits ou permissions, etc.) de fichier

SYNOPSIS

chmod [OPTION]... MODE[,MODE]... FICHIER...

chmod [OPTION]... MODE-OCTAL FICHIER

chmod [OPTION]... --reference=FICHIER-R FICHIER

DESCRIPTION

Le programme chmod modifie les bits de comportement (droits ou permissions, etc.)

de fichier de chacun des fichiers indiqués, en suivant l'indication de mode, qui peut

être une représentation symbolique des changements à effectuer, ou un nombre

octal représentant le motif binaire des nouveaux bits de comportement.

MKDIR(1)

NOM

mkdir - Créer des répertoires

SYNOPSIS

mkdir [OPTION]... RÉPERTOIRE...

DESCRIPTION

Créer les RÉPERTOIREs s'ils n'existent pas.

MV(1)

NOM

mv - Déplacer ou renommer des fichiers

SYNOPSIS

mv [OPTION]... [-T] SOURCE CIBLE

mv [OPTION]... SOURCE... RÉPERTOIRE

mv [OPTION]... -t RÉPERTOIRE SOURCE...

DESCRIPTION

Renommer la SOURCE en CIBLE ou déplacer la SOURCE vers le RÉPERTOIRE.

ECHO(1)

NOM

echo - Afficher une ligne de texte

SYNOPSIS

echo [OPTION-COURTE]... [CHAÎNE]...

echo OPTION-LONGUE

DESCRIPTION

Afficher la(les) CHAÎNE(s) en écho sur la sortie standard.

PRINTF(1)

NOM

printf - Formater et afficher des données

SYNOPSIS

printf FORMAT [PARAMÈTRE]...

printf OPTION

DESCRIPTION

Afficher le(s) PARAMÈTRE(s) selon le FORMAT ou exécuter en fonction

de l'OPTION :

--help Afficher l'aide-mémoire et quitter

--version Afficher le nom et la version du logiciel et quitter

FORMAT contrôle la sortie de la même manière que l'instruction printf du langage C.

**Quelques fichiers :**

« ~/.profile » (ou « ~/.bash\_profile », selon les versions) :

fichier de script de démarrage du shell bash spécifique d’un utilisateur

« /etc/profile » :

fichier de script de démarrage commun à l’ensemble des utilisateurs

**LABORATOIRE – Programmation de scripts – 1**

**Dans la distribution Debian GNU/Linux :**

Dans la suite, le nom de l’utilisateur ordinaire sera représenté par le mot « user ».

Connectez-vous dans un terminal

Syntaxe du shell :

#!/bin/bash Pseudo-commentaire, au début d’un script, qui indique à la commande qui active le script que c’est un shell bash qu’elle doit activer pour exécuter le script

#!/bin/sh Pseudo-commentaire, au début d’un script, qui indique à la commande qui active le script que c’est un shell sh qu’elle doit activer pour exécuter le script

# Annonce d’un commentaire

; Séparateur de commandes (comme un saut de ligne ou une tabulation)

& Séparateur de commandes, mais la commande qui le précède est exécutée en tâche de fond

. Exécution d’un script shell sans activation d’un nouveau processus shell (une telle activation constituant le comportement par défaut)

Affichez le manuel de la commande uptime

man uptime

Dans ~, enregistrez sous le nom « script1 » un fichier de script qui affiche la date et l’heure système, puis la charge système, puis toutes les informations sur le système

Pour faire cela, lancez d’abord un éditeur « gedit » avec en argument le nom « script1 »

gedit script1

Tapez ensuite le texte du script dans l’éditeur gedit, puis enregistrez

#!/bin/bash

# Un premier script

date # affiche la date et l’heure système

uptime # affiche la charge système

uname -a # affiche toutes les informations sur le système

Exécutez un dump de script1 sur le terminal

cat script1

Affichez script1 en filtrant

more script1

less script1

Visualisez script1 à l’aide de plusieurs éditeurs

nano script1

emacs script1

joe script1

gedit script1

Affichez le manuel de la commande bash

man bash

Lancez un shell bash qui exécute script1

bash script1

bash < script1

Lancez un shell sh qui exécute script1

sh script1

Lancez un shell bash qui exécute script1, en affichant la phase d’interprétation et la trace des commandes

bash -x script1

Affichez le manuel de la commande chmod

man chmod

Listez script1 afin de visualiser les permissions actuelles

ls -l script1

Modifiez les permissions sur le fichier script1 de façon à ce que tout le monde reçoive la permission de l’exécuter

chmod +x script1

chmod a+x script1

Listez script1 afin de visualiser la modification des permissions

ls -l script1

Exécutez script1 sous forme d’une commande à partir du répertoire courant

./script1

Syntaxe du shell :

. Exécution d’un script shell sans activation d’un nouveau processus shell (une telle activation constituant le comportement par défaut)

Exécutez script1 dans le shell courant, et non dans un shell fils, sous forme d’une commande à partir du répertoire courant

. ./script1

Créez un répertoire de binaires exécutables sous le dossier personnel de l’utilisateur courant

mkdir /home/user/bin

Déplacez script1 dans /home/user/bin

mv script1 bin

Variables d’environnement :

PS1 Prompt

HOME Chemin absolu du répertoire de connexion

PATH Chemins des répertoires où sont recherchées les commandes (séparés par « : »)

Syntaxe du shell :

= Affectation à une variable d’une valeur

$variable la variable est évaluée

Modifiez transitoirement la valeur de la variable PATH de façon à ce qu’elle comprenne aussi le chemin « /home/user/bin »

PATH=$PATH:/home/user/bin

Exécutez script1 sous forme d’une commande

script1

Syntaxe du shell :

`commande` la commande est interprétée

$(commande) la commande est interprétée

"..." Annulation de la signification des caractères spéciaux, sauf : « ` », « $ » et « \ »

'...' Annulation de la signification de tous les caractères spéciaux

NOM=VALEUR crée la variable NOM et lui affecte la valeur VALEUR

Créez une variable nommée : « CONFIG\_SHELL » et affectez-lui comme valeur le chemin absolu du fichier système profile : « /etc/profile »

CONFIG\_SHELL=/etc/profile

Affichez le manuel des commandes echo et printf

man echo

man printf

Avec echo, affichez la valeur de la variable CONFIG\_SHELL

echo $CONFIG\_SHELL

Avec echo, affichez la valeur de la variable CONFIG\_SHELL, avec alerte

echo -e "\a" $CONFIG\_SHELL

Utilisez la variable CONFIG\_SHELL afin de lister, en utilisant un format d’affichage long, le fichier système profile

ls -l $CONFIG\_SHELL

Avec printf, affichez la valeur de la variable CONFIG\_SHELL, cadrée à droite sur 20 positions, avec passage à la ligne consécutif

printf "%20s\n" $CONFIG\_SHELL

Avec printf, affichez la valeur de la variable CONFIG\_SHELL, cadrée à gauche sur 20 positions, avec passage à la ligne consécutif

printf "%-20s\n" $CONFIG\_SHELL

Avec printf, affichez la chaîne de caractères « Les poires dorées », cadrée à droite sur 30 positions, avec passage à la ligne consécutif

printf "%30s\n" 'Les poires dorées'

Déconnectez-vous

exit