

LES SCRIPTS LINUX

Introduction :

Pour exécuter un script, il faut que le fichier soit exécutable : `chmod u+x nomscript`.

Pour lancer un script, faire `./nomscript`

Le script doit toujours commencer par `#!/bin/bash`

Caractère particuliers :

Caractère	Signification
#	Commentaire
;	Séparateur de commande
::	fin de liste de commande pour case
"	Banalisation partielle
'	Banalisation totale
\	Banalisation du caractère suivant
{}	Regroupe des commandes

Les arguments :

Le passage d'argument se fait par `nomscript arg1 arg2 arg3`.

Les arguments ainsi passés ont pour nom `$1, $1,$3`.

`$*` désigne l'ensemble des arguments.

`$#` c'est le nombre d'arguments.

Affectation d'un argument : `nomargument=valeurargument`

Code retour : 0 = succès de la commande.

Exemple :

Créer script à l'aide de vi contenant les lignes suivantes :

```
#!/usr/bin/env bash
```

```
echo "$$ : c'est le PID du script"
echo "$* : c'est la liste des arguments passé au script"
echo "$# : c'est le nombre d'argument passé à la commande"
echo "$1 : c'est le premier argument"
echo "$? : c'est le retour de l'echo (résultat commande) précédent (premier argument)"
echo "$0 : c'est le mon du script"
```

Rendre exécutable le script :

```
cyrille@cyrille:~$ chmod u+x script
```

Lancer le script en lui passant 4 arguments :

```
cyrille@cyrille:~$ ./script 04 12 156 398
```

```
6066 : c'est le PID du script
```

```
04 12 156 398 : c'est la liste des arguments passé au script
```

```
4 : c'est le nombre d'argument passé à la commande
```

```
04 : c'est le premier argument
```

```
0 : c'est le retour de l'écho précédent (premier argument). 0 = réussite de la commande
```

```
./script : c'est le mon du script
```

```
cyrille@cyrille:~$
```

Les tests :

Opérateur de test des chaînes	Signification
-gt	supérieur
-ge	supérieur ou égal
-eq	égal
-ne	non égal
-le	Inférieur ou égal
-lt	Inférieur
-z	Variable vide

Exemple :

```
if test nombre1 -gt nombre2
```

Opérateur de test de chemin d'accès	Signification
-f	vrai si fichier existe
-d	vrai si répertoire existe
-r	vrai si accès en lecture autorisé
-w	vrai si accès en écriture autorisé
-x	vrai si accès en exécution autorisé

Exemple :

```
if test -f /mnt:floppy
```

Les instructions :**While (tant que) :**

Tant que la conditions n'est pas vraie, on effectue les commandes.

```
while [conditions à valider]
do
    commandes à effectuer
done
```

if (si) :

Si la condition est vraie, on exécute la commande1 sinon on exécute la commande2.

```
if [conditions à valider]
then commande1 à effectuer
else
    commande2 à effectuer
fi
```

For (pour) :

Pour la variable allant de 0 à n on exécute la commande.

```
for variable
do
    commandes à effectuer
done
```

Read :

Permet de lire les saisies utilisateurs.

Exemple 1 utilisation read:

Créer le fichier script1

```
#!/bin/bash
echo Faites une saisie
read REPONSE
echo Vous avez saisie : $REPONSE
```

Rendre executable le script :

```
cyrille@cyrille:~$ chmod u+x script1
```

Lancer le script :

```
cyrille@cyrille:~$ ./script1
```

Faites une saisie

premier script

Vous avez saisie : premier script

```
cyrille@cyrille:~$
```

Exemple de script :

Le script ci dessous demande à l'utilisateur de saisir le répertoire de recherche afin de rechercher les fichiers exécutables pour les comptabiliser et les afficher.

```
#!/usr/bin/env bash

echo "LANCEMENT DU SCRIPT $0 ..."
sleep 1

#Saisie utilisateur
echo "# Veuillez entrer un nom de répertoire #"

echo "# et valider par la touche \"Enter\". #"

read REPERTOIRE

#Recherche les fichiers executables et stockage du resultat
RESULTAT="$(find ${REPERTOIRE} -type f -perm -111)"

#ls -l pour chaque fichier et on les compte

COMPTEUR=0
for fichier in $RESULTAT
do
    echo "#####"
    echo "# fichier : ${fichier}"
    echo "# ls -l : $(ls -l ${fichier})"
    echo "#####"
    echo ""
    echo ""

    (( COMPTEUR = COMPTEUR +1 ))
done

#Affichage du nombre de fichier exécutables

echo "#####"
echo "Nombre de fichiers executables : ${COMPTEUR}"
echo "#####"
exit 0
```

Résultat script :

J'ai indiqué comme répertoire /home/cyrille/

```
cyrille@cyrille:~$ ./script2
LANCEMENT DU SCRIPT ./script2 ...
# Veuillez entrer un nom de répertoire #
# et valider par la touche "Enter". #
/home/cyrille/

#####
# fichier : /home/cyrille/.openoffice.org2/user/config/hatching_en-US.soh
# ls -l : -rwxrwxrwx 1 cyrille cyrille 5238 2007-07-10 17:13
/home/cyrille/.openoffice.org2/user/config/hatching_en-US.soh
#####

#####
# fichier : /home/cyrille/.openoffice.org2/user/config/palette_en-US.soc
# ls -l : -rwxrwxrwx 1 cyrille cyrille 5271 2007-07-10 17:13
/home/cyrille/.openoffice.org2/user/config/palette_en-US.soc
#####

#####
# fichier : /home/cyrille/.openoffice.org2/user/config/modern_en-US.sog
# ls -l : -rwxrwxrwx 1 cyrille cyrille 6840 2007-07-10 17:13
/home/cyrille/.openoffice.org2/user/config/modern_en-US.sog
#####

#####
# fichier : /home/cyrille/.openoffice.org2/user/config/styles_en-US.sod
# ls -l : -rwxrwxrwx 1 cyrille cyrille 1708 2007-07-10 17:13
/home/cyrille/.openoffice.org2/user/config/styles_en-US.sod
#####

#####
Nombre de fichiers exécutables : 4
#####
cyrille@cyrille:~$
```