

**Module 1 – Atelier Pratique – entrées – sorties – variables – if**  
**Série 1 Théorique et Pratique - Solution**  
**Scripts Linux - 420-127-LG**

Nom :  
Prénom

---

Si vous reconnaissez votre solution. Je vous en remercie d'avance au nom de l'open source.

1 Que permet un script Linux ?

D'automatiser des tâches qui seraient autrement longue à effectuer (à taper) ou à rendre une opération plus facile pour l'utilisateur.

2 Que permet l'instruction « exit 0 » à la fin d'un script ?

Termine l'exécution d'un script et renvoie la valeur 0 au programme appelant.

3 Encerclez la bonne réponse.

3.1 Quelle instruction retrouve-t-on généralement au début d'un script ?

- a) /bin/bash
- b) bash
- c) #/bin/bash
- d) #!/bin/bash
- e) Aucune instruction particulière

3.2 Si le script se nomme « exer32 », que doit-on réaliser pour pouvoir rendre ce script exécutable ?

- a) exec exer32
- b) chmod +x exer32
- c) ./exer32
- d) aucune de ces réponses.

3.3 Si je tape « ./exer33 », je

- a) Sauvegarde le fichier exer33
- b) Exécute le fichier exer33
- c) Charge le fichier exer33
- d) Compile le fichier exer33
- e) Aucune de ces réponses.

4 Comment puis-je faire afficher la phrase « Bonjour le monde » à l'écran en bash ?

echo "Bonjour tout le monde"

5 Nommez deux types de shell qui existe dans le monde Linux ?

1. c-shell 2.bash

6 Si le script se nomme « calc » et que je tape « ./calc 1 2 allo ». Si j'écris la ligne suivante dans le script « echo \$3 ». Que sera-t-il affiché à l'écran ?

allo

- 7 J'ai une variable nommée \$Var qui contient la valeur 5. Comment puis-je faire afficher à l'écran le résultat de la variable \$Var multipliée par 10.  
`echo $(( $Var * 10 ))`
- 8 J'ai une variable nommée \$Resultat qui contient la valeur 100. Comment puis-je faire afficher la phrase : La variable \$Resultat = 100  
`echo "La variable \ $Resultat = $Resultat"`
- 9 j'ai une variable \$Var qui vaut 9. Donnez 2 façons de faire afficher le résultat du carré de la variable \$Var.  
`echo $(( $Var * $Var ))` ou `echo $(( Var**2 ))`
- 10 J'aimerais lire une valeur à partir du clavier, ne pas faire afficher ce que je tape et donner 10 secondes à l'utilisateur pour entrer la valeur sans quoi le script continue son exécution. Quelle est l'instruction à faire ?  
`read -s -t 10`
- 11 Comment puis-je tester si une variable est vide en utilisant un « if » ?  
`if [ -z $Var ]; then`
- 12 Écrivez la condition nécessaire pour vérifier l'existence d'un fichier.  
`if [ -f $1 ]; then`

## Exercices Pratiques Script Linux

1. Réalisez un script bash nommé « lc » qui affichera le contenu du répertoire avec la couleur. (Souvenez-vous de la commande : `ls -al --colors`)

```
#!/bin/bash  
  
ls -al -colors  
  
exit 0
```

2. Faire un script bash nommé « exer2 » qui permet de faire afficher la date du jour sur une ligne et sur la ligne suivante les usagers branchés au système. Souvenez-vous des commandes « date » et « who ».

```
#!/bin/bash  
  
date  
Who  
  
exit 0
```

3. Réalisez un script qui permet de copier un fichier source vers un fichier destination. Le programme demandera le fichier source à l'utilisateur ainsi que le nom du fichier destination. Voici un exemple:

```
schasse@Kepler:~/Stephane/Cours/420-KAC/Hiver 2010/exemple_Script$ ./exemple1
*** Utilitaire de copie de fichier ***
Entrez le nom du fichier source à copier: fichier_test
Entrez le nom du fichier destination : fichier_copie
Copie de fichier_test vers fichier_copie réalisée avec succès !
```

On ne vérifie pas l'existence du fichier destination pour l'instant.

```
#!/bin/bash

read -p "Entrez le nom du fichier destination" Destination
read -p "Entrez le nom du fichier source" Source

if [ -f $Source ]; then
    cp$Source $Destination
    echo "Fichier $Source copié correctement."
else
    echo "Le fichier source $Source inexistant"
endif

exit 0
```

4. Reprenez le script précédent (#3) et ajoutez les éléments nécessaires pour vérifier si les fichiers sources et destination existe. Dans le cas que le fichier source n'existe pas, il faut envoyer un message d'erreur à l'utilisateur et terminer le script. Dans le cas que le fichier destination existe, il faut demander à l'utilisateur si il veut écraser le fichier destination. Si l'utilisateur ne veut pas écraser le fichier destination qui existe déjà, alors le script se termine.

```
#!/bin/bash
#Auteur: Daniel Quevillon
#Date: 27 octobre 2014

echo "**** Utilitaire de copie de fichier ****"
read -p "Entrez le nom du fichier source à copier: " FichierSource
read -p "Entrez le nom du fichier destination: " FichierDestination
if [ -e $FichierSource ]; then
  if [ -e $FichierDestination ]; then
    echo -n "Le fichier destination existe, voulez-vous l'écraser (y/n)?"
    read Ecraser
    if [ $Ecraser = "y" -o $Ecraser = "Y" ]; then
      cp ./FichierSource ./FichierDestination
      echo "Copie de $FichierSource vers $FichierDestination réalisée avec succes !"
    fi
  else
    cp ./FichierSource ./FichierDestination
    echo "Copie de $FichierSource vers $FichierDestination réalisée avec succes !"
  fi
else
  echo "Le fichier source n'existe pas."
fi
```

5. Réaliser un script qui se nommera "list" et qui permettra les fonctionnalités suivantes :

- Lorsque l'on tape : **list**  
Le script doit faire afficher le contenu du répertoire.  
(Commande ls -al)
- Lorsque l'on tape : **list -a nom\_fichier**  
Le script doit faire afficher le contenu du fichier nommé « nom\_fichier » passé en paramètre. (Commande cat)

- Lorsque l'on tape : **list -d nom\_fichier**  
Le script doit détruire le fichier passé en paramètre. (Commande rm)

Vous devez aussi planifier les possibilités suivantes :

- L'utilisateur entre un nombre plus grand de paramètre.
- Tout autre paramètre que -a ou -d affiche la syntaxe à l'écran.

Dans les deux cas précédents, la syntaxe de la commande doit être affichée.

```
#!/bin/bash

if [ $# -eq 2 ]; then
  case $1 in
    -a) cat $2
        echo
        ;;
    -d) rm $2
        ;;
    *) echo "parametre invalide: syntaxe = list [-a,-d] fichier"
        esac
  else
    if [ $# -eq 0 ]; then
      ls -al
    else
      echo "parametre invalide: syntaxe = list [-a,-d] fichier"
    fi
  fi
fi
exit 0
```

6. Écrire un script bash nommé « calc » qui accepte 2 chiffres en paramètre et une opérande (+, -, /, x), effectue l'opération et affiche le résultat. Ainsi, si on tape :

calc 3 + 5  
On reçoit : 8

calc 3  
On reçoit : Nombre invalide de paramètre

calc 1 - 7  
On reçoit : -6

calc 3 5 6  
On reçoit : Nombre invalide de paramètre

calc 1 / 0  
On reçoit : erreur, ne peut diviser par zéro

calc 10 x 7  
On reçoit : 70

Attention c'est le caractère x et non l'étoile \*.

```
#!/bin/bash
if [ $# -lt 3 ]; then
    echo "Nombre invalide de paramètre"
else
    case $2 in
        +)
            echo "$(($1 + $3))"
            ;;
        -)
            echo "$(($1 - $3))"
            ;;
        x)
            echo "$(($1 * $3))"
            ;;
        /)
            echo "$(($1 / $3))"
            ;;
    esac
fi
exit 0
```