SYSTÈME D’EXPLOITATION UNIX PROGRAMMATION SCRIPTS

SOLUTION DES EXERCICES P1

Exercice 1

1) Affichez les shells disponibles sur votre système

2) Affichez votre shell actuel

3) Affichez votre shell par défaut

4) Basculez d’un shell à un autre puis fermer chaque shell ouvert

Changez votre shell par défaut

5)Vérifiez la disponibilité des éditeurs de texte et familiarisez-vous avec l’un d’eux

Solution ex 1

1) cat /etc/shells

2) shell actuel : echo $0

3) shell par defaut : echo $SHELL

4) sh

bash

zsh

exit

exit

exit

Changez votre shell par défaut : **chsh**

Ou **lchsh**

5)vim, emacs, nano, gedit, ne.

Exercice 1

(suite)

6) En utilisant les commandes tail et head, écrire un script qui permet de réordonner les lignes d'un fichier "losange" contenant un losange dont les lignes sont inversées.

7) Ré-écrivez votre script en une seule ligne.

cat losange.txt

****

*****

***

*

*

***

*****

****

Solution :

Par Saad Alami 3iir8

#!/bin/bash

a=$((cat $1 | wc -l))

a=$((a / 2))

b=$((a - 1))

tail -n $a $1 ; tail -n +2 $1 | head -n $b

En un seule commande :

(tail -4 losange ; tail –n+2 losange | head –n 3)

> losange.tmp ; mv losange.tmp losange

Programmation shell scripts

Exercice 2

Quel est le résultat de la commande suivante :

$ # echo bonjour

Avec une seule commande d'affichage, affichez sur deux lignes le message suivant:

Bonne

Journée

Ecrire un script qui affiche la date en permanence, chaque écriture écrasant la précédente.

Ecrire un script qui affiche sous forme de tableau dix étudiants, et leur classement à l'ordre inverse (en utilisant printf et des espaces).

Etudiant1 10

Etudiant2 9

….

- Etudiant10 1

Solution ex2

1) Rien c’est un commentaire

2) echo -e "bonne\njournée"

3)

```
#!/bin/bash
while true
  do
        echo -n -e "`date` \r"
  done
```

4)

```
#!/bin/bash
for i in {1..10}
   do
        ((j=11-i))
        printf "Etudiant%-2d%10d\n" $i $j
done
```

Mécanismes de base

Exercice 3

Identifiez les fichiers d'initialisation de sessions propres à votre compte.
Créez un alias permanent pour remplacer la commande "ls -la" par "ls"
Écrivez un script qui compte à rebours de 5 à 0, avec une pause d'une seconde à chaque affichage. (en rouge)
Explorez la variable PS1 et redéfinissez de manière permanente votre propre prompte en utilisant des couleurs et en affichant les informations suivantes :

<user>@<host>/<pwd>_<time>_$_

Solution Ex3

1) ls -la

2) echo “alias ls=‘ls –la’ “ >> ~/.bashrc

3) #!/bin/bash

c0="\e[00m"

c1="\e[31;01m"

for i in {5..0}

do

echo -e "\rSuppression dans ${c1}${i}s${c0}\a\c"

sleep 1

done

echo -e "\nSuppression en cours"

4) PS1='\e[01;34m\u@\h \e[01;32m\W\e[0m\e[35m\t\e[31m\$\e[0m'

Mécanismes de base (suite)

Variables, read, Shell expansion, variables de position, variables speciales, set, shift

Exercices 4

Ecrire un script qui affiche le répertoire de travail en cours, en remplaçant le répertoire personnel par le symbole ~ (en préfixe)

Ecrire un script qui prend comme e-argument une adresse mail et affiche le nom de login correspondant.

En utilisant la commande mv, changez l'extension de tous les fichiers d'un répertoire donné en 1e argument, d'une extension donnée en 2e argument à une autre en 3e argument.

Solution ex4

1) echo ${PWD/$HOME/"~"}

2) #!/bin/bash

adresse=$1

echo ${adresse%@*}

3) #!/bin/bash

for f in *."$1"

 do

 mv $f ${f%"$1"}"$2"

 done

– – –

Solution ex4

Solution 2 : Nisrine Zaim 3iir8

```
#!/bin/bash
for i in $1/*
   do
      if [ -f $i -a `echo $i | grep .$2$` ]
       then
            p1=`echo $i | cut -d "." -f 1`
            p2="$p1$3“
            mv $i $p2
      fi
   done
```

Solution ex4

Solution 3 : Souleiman Kassou 3iir8

```
#!/bin/bash
for something in $1/*
    do
      if [ -f something ]
            then
             mv $something ${something/.*/.$2}
      fi
     done
```

Exercices 4

(suite 1)

4. Vous souhaitez écrire un script qui prend obligatoirement un paramètre d'une longueure de 5 caractères minimum. Ecrire une expression qui permet de vérifier la syntaxe. En cas d'erreur, elle doit afficher la syntaxe correcte et quitter avec un code 1.

Solution ex4

(suite 1)

```
#!/bin/bash
echo "Saisir le parametre: "
read param
if [[ ${#param} -lt 5 ]] ; then
echo "Erreur! Veuillez entrer un Chaine de plus de 5 caracteres !"
  exit 1
else
  echo "On peut continuer"
  echo "Vous avez saisi $param :          longeur = ${#param}  caracteres"
 exit 0
fi
```

Exercices 4

suite (2)

5. Ecrire un script bash qui demande la saisie au clavier d'un nom complet et affiche séparément :

   Le prénom

   Le nom de famille

   Le/s prénoms secondaires

Solution ex4

suite (2)

```
#par Ben Aghmouche Yahya 3iir5
echo "enter you name "
read name
n=`echo $name | cut -d " " -f1 `
m=`echo $name | cut -d " " -f2 `
p=`echo $name | cut -d " " -f3 `
if [ -z "$p" ]
then
echo -e "your first name : $n«
echo -e " your last name:  $m"
else
echo -e "your first name : $n your middle name : $m
\n your last name : $p«
fi
```

Solution ex4

suite (2)

Solution 2

```
read -p "enter full name "

fullnamefirst="${fullname%% *} "

last="${fullname##* }"

middle="${fullname#$first} "

middle="${middle%$last}«

echo "first name : " $first

echo "middle name : " $middle

echo "last name : " $last
```

— — —

Fonctionnement interactif

Redirections, tableaux, getopts, tee, xargs

Exercices 5

1. Ecrire un script qui demande la saisie d'un nom de fichier complet en option -f (avec son chemin d'accès) et affiche successivement le nom seul et le chemin seul.
2. En utilisant xargs dans un script, cherchez le texte donné en premier argument du script dans tous les fichiers du répertoire donné en deuxième argument, en coloriant le texte recherché dans les lignes de résultat.

– – –

Solution Ex 5.1

```
#!/bin/bash
usage() { echo "Usage: $0 [ -f FILENAMe ]" 1>&2}
getopts ":f:" optcase $opt in
f) [ ! -f $OPTARG ] && echo "$OPTARG n'est pas un fichier" 1>&2 && exit 2
echo "Le fichier est: ${OPTARG##/}"
echo "Le chemin est : ${OPTARG%/}"
;;
:)echo "L'option -$OPTARG nécessite un argument" >&2 usage && exit 1
;;
*) echo "Option -$OPTARG non supporte"
usage 1>&2 && exit 3
;;
esac
```

Solution Ex 5.2

Solution 1

```
#!/bin/bash
[ $# -ne 2 -o ! -d $2 ] && echo "USAGE: $(basename $0) <motif> <rep>" 1>&2 && exit 1
ls $2/* | xargs grep  -rnwH "$1" 2>/dev/null
```

Solution 2 :

```
#!/bin/bash
#pour affichage avec couleur
[[ -z $1 || ! -d $2 ]] && echo "USAGE $(basename $0) chaine rep" && exit 1
res=$(ls $2 | xargs grep $1 2>/dev/null)
echo -e ${res//$1/"\e[35;1m$1\e[0m"}
```

Exercice 5

(suite)

3. Ecrire un script bash qui prend des options -h ou --host et -p ou --port pour télécharger une page web en utilisant curl ou wget

Solution Exercice 5.3

```
#!/bin/bash
while [ "$1" != "" ]; do
case $1 in
-p | --port ) shift
port=$1
;;
-h | --host ) shift
host=$1
;;
--help ) echo "Usage $(basename $0) [-h|--host host] [-p|--port
        port] "
exit;;
*) echo "Usage $(basename $0) [-h|--host host] [-p|--port port] "
exit 1
;;
esac
Shift
done
: ${host:="http://localhost"}
port=${port:-80}
echo "$host:$port "
wget "$host:$port"
```