TP 4 - Nicolas PIPLARD - E3FIC - 11/09/2020

2.1. Commmande test

Ecrivez un script qui dit si le paramètre passé est un fichier/dossier/n'existe pas :

```
#!/bin/bash
# On check si il y a bien au moin un argument en paramètre
if [ $# -eq 0 ]; then
    echo "Vous devez passer au moin 1 paramètre"
    exit 1
fi
# On bouble le nombre de paramètres et affiche chaque résultat dans le shell
for i in $@
    do
        if [ -f $i ]; then
            echo "$i est un fichier"
        elif [ -d $i ]; then
            echo "$i est un dossier"
        else
            echo "$i n'existe pas"
        fi
done
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./premier.sh test
 fichier caca
test est un dossier
fichier est un fichier
caca n'existe pas
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

Ecrivez un script qui n'affiche que les répertoires :

```
#!/bin/bash
for i in *; do
    if [ -d $i ]; then
        echo "dossier : $i"
    fi
done
```

Ecrivez un script qui n'affiche que lesfichier

```
#!/bin/bash
for i in *; do
    if [ -f $i ]; then
        echo "fichier : $i"
    fi
done
```

Ecrivez un script qui donne le nombre de fichiers et le nombre de répertoires :

```
#!/bin/bash
d=0
f=0
for i in *; do
    if [ -d $i ]; then
        ((d=d+1))
    elif [ -f $i ]; then
        ((f=f+1))
    fi
done
echo "Il y a $d dossier(s)"
echo "Il y a $f fichier(s)"
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./second.sh
Il y a 1 dossier(s)
Il y a 3 fichier(s)
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

2.2. Commande case

En utilisant la structure case, écrire un script qui affiche le menu/demande à l'utilisateur de saisir une option du menu/affiche l'option qu'il a choisie :

version simple demandé :

```
#!/bin/bash
echo "===== Menu ====="
options=("1) Option 1" "2) Option 2" "3) Option 3" "4) Option 4")
for opt in "${options[@]}"; do
    echo $opt
done
echo "Veuillez choisir un option :"
read choice
case $choice in
    1)
        echo "Vous avez choisit l'option 1"
```

```
        ;;
    2)
        echo "Vous avez choisit l'option 2"
        ;;
    3)
        echo "Vous avez choisit l'option 3"
        ;;
    4)
        echo "Vous avez choisit l'option 4"
        ;;
    *) echo "Option invalide";;
esac
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./quatre.sh
===== Menu =====
1) Option 1
2) Option 2
3) Option 3
4) Option 4
5) Quitter
Veuillez choisir un option :
3
Vous avez choisit l'option 3
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./quatre.sh
===== Menu =====
1) Option 1
2) Option 2
3) Option 3
4) Option 4
Veuillez choisir un option :
5
Option invalide
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

version amélioré avec le select, c'est-à-dire que le menu se réaffiche à la fin de chaque traitement :

```
#!/bin/bash
echo "Choisissez une option :"
options=("Option 1" "Option 2" "Option 3" "Option 4" "Quitter")
select opt in "${options[@]}"
do
    case $opt in
        "Option 1")
            echo "Vous avez choisit l'option 1"
            ;;
        "Option 2")
            echo "Vous avez choisit l'option 2"
            ;;
        "Option 3")
            echo "Vous avez choisit l'option 3"
            ;;
        "Option 4")
            echo "Vous avez choisit l'option 4"
            ;;
```

```
        "Quitter")
            break
            ;;
        *) echo "Option invalide";;
    esac
done
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./trois.sh
Choisissez une option :
1) Option 1
2) Option 2
3) Option 3
4) Option 4
5) Quitter
#? 4
Vous avez choisit l'option 4
#? 5
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

En utilisant la structure for, écrire un programme qui donne les valeurs de Y d'une fonction pour les valeurs de X allant de -10 à 10 avec un incrément de 1 :

Réalisez le traitement pour la fonction y=x^2 :

```
#!/bin/bash
y=0
for (( x=-10; x<=10; x++ )) do
    ((y=$x**2))
    echo "pour x = $x"
    echo "y = $y"
done
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./five.sh
pour x = -10
y = 100
pour x = -9
y = 81
pour x = -8
y = 64
pour x = -7
y = 49
pour x = -6
y = 36
pour x = -5
y = 25
pour x = -4
y = 16
pour x = -3
y = 9
pour x = -2
y = 4
pour x = -1
y = 1
pour x = 0
y = 0
pour x = 1
y = 1
pour x = 2
y = 4
pour x = 3
y = 9
pour x = 4
y = 16
pour x = 5
y = 25
pour x = 6
y = 36
pour x = 7
y = 49
pour x = 8
y = 64
pour x = 9
y = 81
pour x = 10
y = 100
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

Réécrivez le programme avec la structure répéter .... jusqu'à :

```bash
#!/bin/bash
y=0
x=-10
until [ $x -gt 10 ]; do
    ((y=$x**2))
    echo "pour x = $x"
    echo "y = $y"
    ((x=$x+1))
done
```

Adapter le script afin que les bornes -x et +x soient passées en paramètres au script :

```
#!/bin/bash
y=0
xmin=-10
xmax=10
echo "Saisissez -x :"
read xmin
echo "Saisissez +x :"
read xmax
if [ $xmin -gt $xmax ]; then
    echo "-x ne doit pas être supérieur à +x"
    exit 1
fi
until [ $xmin -gt $xmax ]; do
    ((y=$xmin**2))
    echo "pour x = $xmin"
    echo "y = $y"
    ((xmin=$xmin+1))
done
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./six.sh
Saisissez -x :
50
Saisissez +x :
70
pour x = 50
y = 2500
pour x = 51
y = 2601
pour x = 52
y = 2704
pour x = 53
y = 2809
pour x = 54
y = 2916
pour x = 55
y = 3025
pour x = 56
y = 3136
pour x = 57
y = 3249
pour x = 58
y = 3364
pour x = 59
y = 3481
pour x = 60
y = 3600
pour x = 61
y = 3721
pour x = 62
y = 3844
pour x = 63
y = 3969
pour x = 64
y = 4096
pour x = 65
y = 4225
pour x = 66
y = 4356
pour x = 67
y = 4489
pour x = 68
y = 4624
pour x = 69
y = 4761
pour x = 70
y = 4900
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./six.sh
Saisissez -x :
10
Saisissez +x :
5
-x ne doit pas être supérieur à +x
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

Modifiez le script de façon à ce que l'on puisse passer en paramètre l'incrément :

```
#!/bin/bash
y=0
xmin=-10
xmax=10
increment=1
echo "Saisissez -x :"
read xmin
echo "Saisissez +x :"
read xmax
echo "Saisissez l'increment :"
read increment
if [ $xmin -gt $xmax ]; then
    echo "-x ne doit pas être supérieur à +x"
    exit 1
fi
if [ $increment -le 0 ]; then
    echo "l'increment doit être supérieur à 0"
    exit 1
fi
until [ $xmin -gt $xmax ]; do
    ((y=$xmin**2))
    echo "pour x = $xmin"
    echo "y = $y"
    ((xmin=$xmin+$increment))
done
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./six.sh
Saisissez -x :
-10
Saisissez +x :
10
Saisissez l'increment :
2
pour x = -10
y = 100
pour x = -8
y = 64
pour x = -6
y = 36
pour x = -4
y = 16
pour x = -2
y = 4
pour x = 0
y = 0
pour x = 2
y = 4
pour x = 4
y = 16
pour x = 6
y = 36
pour x = 8
y = 64
pour x = 10
y = 100
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

2.4. Commande if

En utilisant la structure if, écrire un script qui : afficher le menu/demande à l'utilisateur de saisir une option du menu/affiche à l'utilisateur l'option qu'il a choisie

```
#!/bin/bash
echo "===== Menu ====="
options=("1) Option 1" "2) Option 2" "3) Option 3" "4) Option 4")
for opt in "${options[@]}"; do
    echo $opt
done
echo "Veuillez choisir un option :"
read choice
if [ $choice == 1 ]; then
    echo "Vous avez choisi l'Option $choice"
elif [ $choice == 2 ]; then
    echo "Vous avez choisi l'Option $choice"
elif [ $choice == 3 ]; then
    echo "Vous avez choisi l'Option $choice"
elif [ $choice == 4 ]; then
    echo "Vous avez choisi l'Option $choice"
else
    echo "Votre choix n'est pas valide"
```

```
fi
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./six.sh
===== Menu =====
1) Option 1
2) Option 2
3) Option 3
4) Option 4
Veuillez choisir un option :
2
Vous avez choisi l'Option 2
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./six.sh
===== Menu =====
1) Option 1
2) Option 2
3) Option 3
4) Option 4
Veuillez choisir un option :
0
Votre choix n'est pas valide
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

Construisez un script qui permet de n'afficher que les enregistrements dont la valeur est supérieure à 10 :

```
#!/bin/bash
while IFS=' ', read -r word number
do
    if [ $number -gt 10 ]; then
        echo $number
    fi
done < data.txt
```

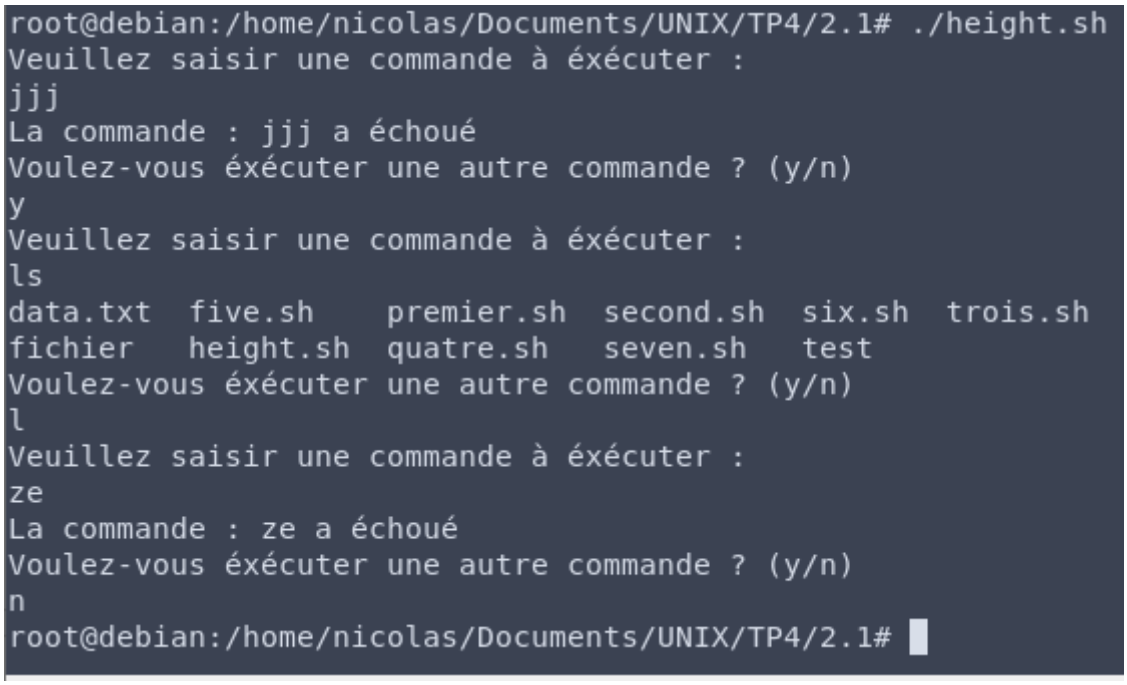
```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./seven.sh
25
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

2.5. Commande until

En utilisant la structure until...do...done, écrire un script qui : demande à un utilisateur de saisir une commande / exécute la commande ou affiche un message d'erreur si la commande ne s'est pas déroulée. / répète cette opération tant que l'utilisateur le désire :

```
#!/bin/bash
response=y
until [ $response == "n" ]; do
    echo "Veuillez saisir une commande à éxécuter :"
    read commande
    $commande 2>/dev/null
```

```
    # 2>/dev/null permet d'éviter d'avoir un message d'erreur si la commande
n'existe pas:ou tout autre message d'erreur.
    if [ $? -ne 0 ]; then
        echo "La commande : $commande a échoué"
    fi
    # On demande à l'utilisateur si il veut continuer
    echo "Voulez-vous éxécuter une autre commande ? (y/n)"
    read response
done
```



2.6. Commande while

En utilisant la structure while, écrire un script qui : Tant que l'utilisateur n'a pas tapé 9 : / affiche un menu / demande à l'utilisateur de saisir une option du menu :

1 Afficher la date (date)"

2 Afficher le nombre de personnes connectées (who)"

3 Afficher la liste des processus (ps)"

9 Quitter"

```
#!/bin/bash
choice=0
while [ $choice -ne 9 ]; do
    echo "===== Menu ====="
    options=("1) Afficher la date (date)" "2) Afficher le nombre de personnes
connectées (who)" "3) Afficher la liste des processus (ps)" "9) Quitter")
    for opt in "${options[@]}"; do
        echo $opt
    done
    echo "Veuillez choisir une option :"
    read choice
    case $choice in
```

```
        1)
            date
            ;;
        2)
            who
            ;;
        3)
            ps
            ;;
        9)
            ((choice=9))
            ;;
        *) echo "Option invalide";;
    esac
done
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./nine.sh
===== Menu =====
1) Afficher la date (date)
2) Afficher le nombre de personnes connectées (who)
3) Afficher la liste des processus (ps)
9) Quitter
Veuillez choisir une option :
1
Fri 11 Sep 2020 11:47:23 AM CEST
===== Menu =====
1) Afficher la date (date)
2) Afficher le nombre de personnes connectées (who)
3) Afficher la liste des processus (ps)
9) Quitter
Veuillez choisir une option :
2
nicolas  tty2         2020-09-11 08:56 (tty2)
===== Menu =====
1) Afficher la date (date)
2) Afficher le nombre de personnes connectées (who)
3) Afficher la liste des processus (ps)
9) Quitter
Veuillez choisir une option :
3
  PID TTY          TIME CMD
 2235 pts/0    00:00:00 su
 2236 pts/0    00:00:00 bash
 4694 pts/0    00:00:00 su
 4759 pts/0    00:00:00 su
 4760 pts/0    00:00:00 bash
 8012 pts/0    00:00:00 nine.sh
 8037 pts/0    00:00:00 ps
===== Menu =====
1) Afficher la date (date)
2) Afficher le nombre de personnes connectées (who)
3) Afficher la liste des processus (ps)
9) Quitter
Veuillez choisir une option :
4
Option invalide
===== Menu =====
1) Afficher la date (date)
2) Afficher le nombre de personnes connectées (who)
3) Afficher la liste des processus (ps)
9) Quitter
Veuillez choisir une option :
9
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```

2.7. Commande select

Vous allez à l'aide de la fonction select réaliser un menu à quatre options pour un utilisateur.

Le script doit boucler tant que l'utilisateur n'a pas choisi de quitter.

Option 1 : Afficher la liste des utilisateurs connectés

Option 2 : Afficher la liste des processus

Option 3 : Afficher à l'utilisateur son nom, son UID, son GID, son TTY

Option 4 : Quitter

```
#!/bin/bash
echo "Choisissez une option :"
options=("Option 1" "Option 2" "Option 3" "Quitter")
select opt in "${options[@]}"
do
    case $opt in
        "Option 1")
            who
            ;;
        "Option 2")
            ps
            ;;
        "Option 3")
            GID=`id -g $USER`
            TTY=`tty`
            echo "Nom : $USER - UID : $UID - GID : $GID - TTY : $TTY"
            ;;
        "Quitter")
            break
            ;;
        *) echo "Option invalide";;
    esac
done
```

```
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1# ./ten.sh
Choisissez une option :
1) Option 1
2) Option 2
3) Option 3
4) Quitter
#? 3
Nom : nicolas - UID : 0 - GID : 1000 - TTY : /dev/pts/0
#? 2
  PID TTY          TIME CMD
 2235 pts/0    00:00:00 su
 2236 pts/0    00:00:00 bash
 4694 pts/0    00:00:00 su
 4759 pts/0    00:00:00 su
 4760 pts/0    00:00:00 bash
 8597 pts/0    00:00:00 ten.sh
 8611 pts/0    00:00:00 ps
#? 1
nicolas  tty2         2020-09-11 08:56 (tty2)
#? 4
root@debian:/home/nicolas/Documents/UNIX/TP4/2.1#
```