


Compte-rendu TP2

Exercice 1

1. Bash cherche les commandes successivement dans 
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin jusqu'à les trouver
2. La variable d'environnement HOME est celle utilisée si *cd* est utilisé sans argument
3. LANG correspond à la langue utilisé et l'encodage (FR et UTF-8). PWD contient le répertoire courant.

OLDPWD contient le répertoire précédent.




SHELL correspond à l'emplacement du shell et là où se trouvent les commandes.

_ contient le répertoire de la commande *printenv*

5. La commande BASH "ouvre" un nouvel environnement. La variable locale MY_VAR n'existe plus car elle n'était créée que localement dans la première console, la deuxième BASH ne la connaît pas.

En revenant dans la première console avec *exit*, on retrouve la valeur de MY_VAR.

6. La variable MY_VAR étant cette fois une variable d'environnement, tous les BASH la connaîtront car elle n'est plus locale.

7. *echo 'Bonjour à vous deux ' \$NOMS* 

8. *unset* va effacer la variable du système, alors que donner une valeur vide ne supprime pas cette variable. Elle existe toujours mais ne contient rien.

9. *echo '\$HOME = \$HOME*

Programmation BASH

Exercice 2

Mot de passe :

```
#!/bin/bash
```

```
PASSWORD='1234' read -s -p 'Saisissez votre mot de passe : ' mdp
```

```
if [ $PASSWORD = $mdp ]
```

```
then
```

```
echo 'GG mec '
```



fi

users :

```
#!/bin/bash
```

```
if [ $# != 1 ]
```

```
then
```

```
    echo "Utilisation : users.sh nomdutilisateur"
```

```
else
```

```
if cut -d: -f1 /etc/passwd | sort -r | grep -q $1 ; then
```

```
echo "Nom d'utilisateur présent"
```

```
else
```

```
echo "Nom d'utilisateur non présent"
```

```
fi
```

```
fi
```

isnumber :

```
#!/bin/bash
```

```
function is_number() {
```

```
re='^[+-]?[0-9]+([.][0-9]+)?$'
```

```
if ! [[ $1 =~ $re ]] ; then
```

```
    return 1
```

```
else
```

```
    return 0
```

```
fi }
```

```
is_number $1
```

```
if [ $? = 0 ]
```

then

```
echo 'Cest un réel'
```

else

```
echo 'Ce nest pas un réel'
```

fi

facto:

```
#!/bin/bash
```

```
num=$1
```

```
fact=1
```

```
while [ $num -gt 1 ]
```

```
do
```

```
fact=$((fact * num))
```

```
num=$((num - 1))
```

```
done
```

```
echo $fact
```

jeu

```
#!/bin/bash
```

```
correct=$RANDOM
```

```
corr=$((correct % 1000))
```

```
echo $corr
```

```
rep=0
```

```
while [ $rep != $corr ]
```

```
do
```

```
read -p "Entrez votre supposition : " rep
```

```
if [ $rep -gt $corr ]  
  
then echo "Vous etes au dessus !"  
  
elif [ $rep -lt $corr ]  
  
then
```

```
    echo "vous etes en dessous !"
```

```
else
```

```
    echo "gg"
```

```
fi
```

```
done
```

Statistique

```
#!/bin/bash
```

```
function is_entier() {
```

```
    if [ $((($1%1)) -eq 0 ]; then  
        return 1  
    else  
        return 0  
    fi
```

```
}
```

```
min=$1
```

```
max=$1
```

```
total=$#
```

```
while (($#)); do
```

```
    if [ $1 -lt 100 -a $1 -gt -100 ];then  
        is_entier $1
```

```
        if [ $? -eq 1 ]; then

            somme=$((somme+$1))

            if [ $1 -gt $max ]

            then

                max=$1

            elif [ $1 -lt $min ]

            then

                min=$1

            fi

        else

            echo "non"

        fi

    else

        echo "Nombre pas conforme, veuillez recommencer"

        exit

    fi

shift

done
```

```
moyenne=$((($somme/$total))
```

```
echo "moyenne : $moyenne"
```

```
echo " min : $min"
```

```
echo " max : $max"
```