

Examen court de système d'exploitation

Département d'informatique – IUT Villetaneuse

Jeudi 20 mars 2008 – 1H

Remarques : tous les documents de cours, td/tp sont autorisés. Le barème est indicatif.

1 Langage de commande bash - 10 pts

Écrire un programme Bash qui prend en paramètre deux entiers i et j ($i \leq j$) et qui affiche des X pour former un « chapeau » dont le sommet est sur la i ème colonne et dont le coté droit va jusqu'à la j ème colonne, en commençant à la ligne 1 du terminal texte. Voici par exemple l'affichage produit par `port-cerin:$ /bin/bash chapeau.sh 8 13`

```
|-----  
|          X          <= ligne 1  
|         X X        <= ligne 2  
|        X  X         ''  
|       X   X         ''  
|      X    X         ''  
|     X     X         ''  
|    X      X         <= ligne 6  
|port-cerin:$  
|
```

(N'affichez ni les `<= ligne` ni les tirets qui servent à matérialiser le coin supérieur gauche de votre écran d'ordinateur). Vous utiliserez les commandes suivantes :

`tput cup m n`

Envoie la commande écran qui place le curseur en ligne m colonne n . Par exemple, `tput cup 0 0` place le curseur en position (0,0) (le coin en haut à gauche de l'écran, position de curseur appelée 'home')

`tput clear`

Envoie la commande d'effacement de l'écran au terminal courant.

Dans un premier temps vous ne vous souciez pas des cas un peu tordus comme celui ci où vous êtes conduits à ne dessiner qu'un seul coté du chapeau :

```
|-----|
|X          <= ligne 1
| X         <= ligne 2
|  X        ''
|   X       ''
|    X      ''
|     X     <= ligne 6
|port-cerin:$
|
```

Vous exposerez, dans un deuxième temps, les cas particuliers qui peuvent arriver sans coder les solutions.

- Un algorithme possible est le suivant en supposant $j \geq i$. Comme on a $p=j-i+1$ lignes à afficher, nous pouvons mettre en place une boucle de $i \leq p < j$ itérations dans laquelle on affiche un premier X en position $\text{ligne} \leq p+1$ et un deuxième X en position $\text{ligne} \leq 2*\text{ligne}+1$. 'ligne' est initialisé à 1 et le sommet du chapeau est traité dans un cas à par. C'est cet algorithme que nous codons ci-dessous :

```
#!/bin/bash

if [ $1 -gt $2 ]
then
    echo "le deuxieme parametre doit etre superieur ou egal au premier"
    exit
fi

# on efface l'ecran
tput clear

# on affiche le sommet du chapeau
tput cup 0 $1
echo -n 'X'
# on affiche les j-i lignes qui restent
ligne=1
for ((i=$(( $1 )) ; $i < $2; i=$((i+1))))
do
    tput cup $ligne $((i+1))
    echo -n 'X'
    tput cup $ligne $((i-2*ligne+1))
    echo -n 'X'
```

```
    ligne=$((ligne+1))
done

echo
```

Concernant les conditions entre i et j pour ne pas écrire dans la marge gauche avec le code ci-dessus, il faut que $i \leq j \wedge j - i + 1 \geq i$.

2 Langage de commande bash - 10pts

1- Faites une explication du programme principal dans le script ci-dessous puis...

2- Compléter le corps de la fonction `verif` pour que le script ci-dessous affiche uniquement les noms des étudiants écrits tout en majuscule, ou bien tout en minuscule. On commencera par décrire l'algorithme suivi.

Pour le fichier `ListeGroupes.csv` le programme affichera :

```
$ more ListeGroupes.csv
10501311,Mr,B0,DELNEUF,MAXIME
10602789,Mr,B1,DELPIERRE,JEREMY
10601320,Mr,A0,DESSAINT,FLORENT
10601814,Mr,A1,DUARTE,MATHIEU
10603349,Mr,A1,DUFAG,ALEXANDRE
10601894,Mr,A0,DUMONTIER,STEPHANE
10601882,Mr,A2,DURAND,PIERRE
10601945,Melle,B0,BELALIMAT,LEILA
$ /bin/bash majuscule.sh ListeGroupes.csv
-----Fichier en entree: ListeGroupes.csv-----
-----Resultat :-----
BELALIMAT DURAND DUMONTIER DUFAG DUARTE DESSAINT DELPIERRE DELNEUF

#!/bin/bash

# L'algorithme qui est suivi est le suivant :
#
#
#
#
#
#
#
#
#
#
```

```
#  
#  
#  
#  
#  
#  
#
```

```
verif()  
{
```

```
}  
# Le format des fichiers (ici ListeGroupes.csv) est le suivant :  
#10501311,Mr,B0,DELNEUF,MAXIME  
#10602789,Mr,B1,DELPierre,JEREMY  
#10601320,Mr,A0,DESSAINT,FLORENT
```

```
#10601814,Mr,A1,DUARTE,MATHIEU
#10603349,Mr,A1,DUFAG,ALEXANDRE
#10601894,Mr,A0,DUMONTIER,STEPHANE
#10601882,Mr,A2,DURAND,PIERRE
#10601945,Melle,B0,BELALIMAT,LEILA
#10603349,Mr,A1,DUFAG,ALEXANDRE
```

```
# Commentaires sur le programme principal :
```

```
#
#
#
#
#
#
#
#
#
#
#
#
#
```

```
if [ -e /tmp/solution ]; then
    rm /tmp/solution
fi
echo "-----Fichier en entree: $1-----"
cat < $1 | while true    #pour toutes les lignes du fichier
do
    read ligne
    if [ "$ligne" = "" ]; then break; fi
    # traitement de $ligne
    nom='echo $ligne | cut -d "," -f 4'
    verif "$nom"
    if [ $? ]
    then
        echo -n "$nom " >> /tmp/solution
    fi
done

echo
echo "-----Resultat :-----"
cat /tmp/solution
rm /tmp/solution
```

```

#!/bin/bash

# L'algorithme qui est suivi pour la fonction test() est le suivant :
# on parcourt une première fois la chaîne passée en paramètre et
# si on ne détecte que des minuscules alors on peut retourner 1 ; sinon
# on doit vérifier si la chaîne est une chaîne de majuscule
# et retourner 1 si c'est le cas et 0 sinon.
#

verif()
{
    bool=1
    for ((i=0 ; i<${#1}; i++ ))
    do
        c=${1:$i:1}
        if [ "$c" \> "z" -a "$c" \< "a" ]
        then
            bool=0;break;
        fi
    done

    # si bool=1 c'est que la chaîne est en minuscule
    if [ $bool -eq 1 ]
    then
        return 1;
    fi
    # si bool=0 c'est que la chaîne n'est pas en minuscule
    # et on vérifie si elle comporte que des majuscules
    bool=1
    for ((i=0 ; i<${#1}; i++ ))
    do
        c=${1:$i:1}
        if [ "$c" \> "Z" -a "$c" \< "A" ]
        then
            bool=0;break;
        fi
    done

    # si bool=1 c'est que la chaîne est en minuscule
    if [ $bool -eq 1 ]
    then
        return 1;
    fi
}

```

```

else
    return 0
fi
}

# Le format des fichiers (ici ListeGroupes.csv) est le suivant :
#10501311,Mr,B0,DELNEUF,MAXIME
#10602789,Mr,B1,DELPierre, JEREMY
#10601320,Mr,A0,DESSAINT,FLORENT
#10601814,Mr,A1,DUARTE,MATHIEU
#10603349,Mr,A1,DUFAG,ALEXANDRE
#10601894,Mr,A0,DUMONTIER,STEPHANE
#10601882,Mr,A2,DURAND,PIERRE
#10601945,Melle,B0,BELALIMAT,LEILA
#10603349,Mr,A1,DUFAG,ALEXANDRE

if [ -e /tmp/solution ]; then
    rm /tmp/solution
fi
echo "-----Fichier en entree: $1-----"
cat < $1 | while true    #pour toutes les lignes du fichier
do
    read ligne
    if [ "$ligne" = "" ]; then break; fi
    # traitement de $ligne
    nom='echo $ligne | cut -d "," -f 4'
    verif "$nom"
    if [ $? ]
    then
        echo -n "$nom ">> /tmp/solution
    fi
done

echo
echo "-----Resultat :-----"
cat /tmp/solution
rm /tmp/solution

```

Remarque : pour détecter si une chaîne est composée que de majuscules, on peut aussi s'inspirer du code suivant qui utilise une expression régulière :

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

```
echo "La chaine en entree : $1"
j='echo "$1" | egrep "^[:upper:]+$"'
echo $j

if [ "$j" != "" ]
then
    echo "la chaine contient que des majuscules";
else
    echo "faux : la chaine ne contient pas que des majuscules";
fi
```