

Examen court de système d'exploitation

Département d'informatique – IUT Villetaneuse

Jeudi 22 mars 2012 – 1H30

Prénom et Nom :

Groupe :

Remarques : tous les documents de cours, td/tp sont autorisés ainsi que les copies de emails envoyés. Le barème est indicatif. N'oubliez pas d'expliquer ce que vous faites et de fournir des scripts bien documentés.

1 Langage de commande bash - 10 pts

On vous donne un fichier texte organisé en colonnes comme suit :

```
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
```

On vous demande d'écrire un script qui va faire la somme des entiers présents sur la première colonne, puis qui va afficher le nombre d'étudiants dans chacun des groupes A0, A1, A2, B0, B1.

L'algorithme à mettre en place consiste, en un seul passage sur le fichier d'entrée, à repérer les entiers présents sur la première colonne et le groupe présent sur la troisième colonne de chaque ligne. Selon la valeur du groupe on augmente une variable rendant compte du nombre de personne dans le groupe. Nous obtenons :

```
#!/bin/bash
somme=0
nb_a0=0
nb_a1=0
nb_b0=0
nb_b1=0
nb_b2=0
while read line
do
    # on traite la première colonne
    tmp='echo $line | cut -d ',' -f 1'
    somme=$((somme+tmp))
    # on traite la troisième colonne
    tmp='echo $line | cut -d ',' -f 3'
    case "$tmp" in
        "A0") nb_a0=$((nb_a0+1)); break;;
        "A1") nb_a1=$((nb_a1+1)); break;;
        "B0") nb_b0=$((nb_b0+1)); break;;
        "B1") nb_b1=$((nb_b1+1)); break;;
        "B2") nb_b2=$((nb_b2+1)); break;;
    esac
done < file.txt
echo "somme des PID = $somme"
echo "nombre de personne(s) en A0 : $nb_a0"
echo "nombre de personne(s) en A1 : $nb_a1"
echo "nombre de personne(s) en B0 : $nb_b0"
echo "nombre de personne(s) en B1 : $nb_b1"
echo "nombre de personne(s) en B2 : $nb_b2"
```

```
new-host-2:SE christophecerin$ /bin/bash test1.sh
somme des PID = 84716304
nombre de personne(s) en A0 : 2
nombre de personne(s) en A1 : 2
nombre de personne(s) en A2 : 1
nombre de personne(s) en B0 : 2
nombre de personne(s) en B1 : 1
```

2 Langage de commande Bash - 5pts

En vous inspirant du code suivant vu et commenté en cours, écrivez une fonction Bash qui va rechercher si un fichier passé en paramètre à la fonction est présent dans l'arborescence. N'oubliez pas d'expliquer ce que vous faites, par exemple, il faudra répondre à la question : est-ce que vous arrêtez les recherches à la première occurrence trouvée.

```
#!/bin/bash

lsrep ()
{
  for i in `ls` ; do
    if [ "$i" != "." -a "$i" != ".." ] ; then
      for ((j=1;j<$2;j++)); do
        echo -n " ";
      done
      echo "$i"
      if [ -d "$i" ] && ! [ -h "$i" ] ; then
        nb_blancs=`expr $2 + 5`
        cd "$i"
        lsrep $i $nb_blancs
        cd ..
      fi
    fi
  done
}
```

```

        fi
    fi
done
}

lsrep . 0

```

La seule modification majeure à apporter au code est d'insérer un test pour vérifier si \$i est égal au nom du fichier recherché et qui est passé en paramètre (\$2) à la fonction search().

Concernant les modifications mineures, il s'agit de supprimer la gestion des blancs. Au final, nous obtenons le code suivant qui affiche toutes les occurrences du fichier recherché :

```

#!/bin/bash

search ()
{
    for i in `ls` ; do
        if [ "$i" != "." -a "$i" != ".." ] ; then
            if [ "$i" = "$2" ]
            then
                echo "on a trouvé $2"
            fi
            if [ -d "$i" ] && ! [ -h "$i" ] ; then
                cd "$i"
                search $i $2
                cd ..
            fi
        fi
    fi
done
}

```

```
# on recherche toutes les occurrences du fichier 'titi'
search . 'titi'
```

3 Expressions régulières - 5 pts

La commande top a retourné les informations suivantes dans un fichier texte :

| PID | COMMAND | %CPU | TIME | #TH | #WQ | #PORT | #MREG | RPRVT | RSHRD | RSIZE |
|-------|--------------|------|----------|-----|-----|-------|-------|-------|-------|-------|
| 28182 | top | 11.1 | 00:02.35 | 1/1 | 0 | 24 | 33 | 1004K | 264K | 1584K |
| 28168 | quicklookd | 0.0 | 00:00.06 | 6 | 2 | 79 | 73 | 2212K | 4124K | 5436K |
| 28165 | Emacs | 0.2 | 00:08.04 | 2 | 1 | 109 | 180 | 26M | 16M | 37M |
| 28138 | mdworker | 0.0 | 00:00.64 | 3 | 1 | 69 | 117 | 8248K | 4272K | 21M |
| 28043 | ssh | 0.0 | 00:00.02 | 1 | 0 | 22 | 24 | 596K | 244K | 1876K |
| 27925 | VDCAssistant | 0.0 | 00:00.08 | 4 | 1 | 90 | 73 | 960K | 4172K | 4284K |
| 27924 | PluginProces | 22.6 | 14:30.65 | 20 | 2 | 293 | 2709 | 76M | 80M+ | 144M+ |
| 27870 | Preview | 0.0 | 00:08.49 | 2 | 1 | 107 | 178 | 15M | 23M | 38M |
| 24555 | activitymoni | 0.0 | 10:08.34 | 1 | 0 | 27 | 38 | 796K | 244K | 1220K |
| 24552 | Activity Mon | 0.0 | 08:53.53 | 2 | 1 | 107 | 281 | 4116K | 75M | 16M |
| 16020 | iTunesHelper | 0.0 | 00:01.25 | 3 | 1 | 52 | 77 | 488K | 3824K | 632K |

Donner les expressions régulières à passer à la commande egrep pour repérer que :

- la ligne contient un PID ≥ 12 ;
- la commande (champs COMMAND) Emacs a été lancée ;
- le pourcentage d'utilisation du CPU est compris entre 10 et 20% (compris).

A partir du début de ligne on doit trouver des digits, l'avant dernier appartient à la classe [1-9], le dernier à la classe [2-9]

A partir du début de ligne on trouve des digits, puis des blancs, puis

le mot clé 'Emacs'

A partir du début de ligne on doit trouver des digits suivis de blancs suivis de caractères alphabétiques suivis de blancs suivis soit d'un 1 puis d'un digit puis d'un point puis d'un digit soit d'un 2 suivi d'un 0 suivi d'un point suivi d'un 0.