DIRO
IFT 1215

INTRODUCTION AUX SYSTÈMES INFORMATIQUES
DEVOIR N° 4

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle
Http : //www.iro.umontreal.ca/∼mignotte/ift1215/
E-mail : mignotte@iro.umontreal.ca

Programmation Shell Script

Dans ce devoir, on se propose d'implémenter trois routines en langage SCRIPT, utilisant le Bash Shell (/bin/bash) vu en cours.

I. Partie entière de $\sqrt{x}$

Implémenter l'algorithme de Dijkstra permettant de rechercher la partie entière de la racine du nombre entier en langage Shell Script (précédemment implémenter en langage assembleur lors du TP numéro 3). Le script devra s'appeler ROOT.SH. On rappelle le pseudo-code de cette méthode,

Algorithme de Dijkstra

| x Calcul de la partie entière de $\sqrt{x}$

1. Initialisation
$y \leftarrow 0 \qquad \delta \leftarrow 2^{15}$

2. Itération
while $(\delta \neq 0)$ **do**
    if $(y+\delta)^2 \leq x$ ► $y \leftarrow y + \delta$
    else $\delta \leftarrow \delta/2$

Le shell script est bien adapté à l'implémentation de ce type d'algorithme car les calculs (somme, multiplication et division) se font sur des entiers et le résultats de ceux-ci sont encore des entiers (la division est une division entière par exemple). On verra en exercice **III** comment des nombres flottants peuvent être néanmoins manipulés facilement avec le script shell.

II. Gestion/Renommage de Fichiers

On voudrait maintenant un script qui s'appellerait Duplq.sh et qui permettrait de renommer l'ensemble des images de format .ppm existant dans le répertoire courant (où l'on exécute le script) et qui, si la commande Duplq.sh Pic 0 3 était exécuté (ou /bin/sh Duplq.sh Pic 0 3 si la commande précedente ne marche pas), ferait la transformation suivante :

Pic0.ppm	renommé en	Pic3.ppm
Pic1.ppm	renommé en	Pic4.ppm
Pic2.ppm	renommé en	Pic5.ppm
Pic3.ppm	renommé en	Pic6.ppm
Pic4.ppm	renommé en	Pic7.ppm
etc ...		

Attention !, lors du premier renommage (Pic0.ppm renommé en Pic3.ppm), le script devra faire en sorte de ne pas effacer le fichier Pic3.ppm existant dans le répertoire courant. Le deuxième et troisième argument sont supposé être positifs mais le deuxième arguments pas forcement supérieur au premier (Duplq.sh Pic 3 0 devrait marcher aussi).

III. Manipulation/Modification de Fichiers

Une façon simple de gérer les nombres et les opération flottantes avec le script schell est de considérer le calculateur bc. Un exemple d'opération permettant d'additionner deux nombres non entiers (VAR1 et VAR2) serait de considérer la syntaxe suivante [1],

```
VAR1=5.5
VAR2=3.14159
RESULT=`echo $VAR1+$VAR2 | bc`
echo $RESULT
```

qui permettrait d'afficher à l'écran le résultat espéré, c'est à dire 8.64159. On va utiliser ce calculateur bc pour l'élaboration du script suivant ;

Ci-dessous est un fichier de note [2](appelé LISTE avec des noms d'étudiants fictifs) dans le format utilisé par l'administration. Le format de ce fichier permet d'utiliser celui-ci pour l'extraction sécuritaire et confidentielle des notes de l'étudiant par le programme getnotes que vous connaissez.

1. Dans cette syntaxe, les parenthèses et expression du style RESULT=`echo ($VAR1+$VAR2)*2 | bc` n'est pas permise. Par contre, une expression du style RESULT=`echo $tp1*0.075 + $tp2*0.075 + $tp3*0.075 + $tp4*0.075 + $Intra*0.3 + $Final*0.4 | bc` est tout à fait permise.
2. Ce fichier LISTE est disponible sur le lien à droite de celui que vous avez utilisez pour transférer ce fichier Devoir n^o 4.

```
:NOM Prenom:CODE               :.Tp1 :.Intra:.Tp2 :.Tp3 :.Final :.TP4 :.Moy:.Lettre:

actarusx:Actarus  :ACTA28658105 :  45.5:  75.00:100.0:  95.0:   48.50:  90.0:
alcorxxx:Alcor    :ALCO18083408 :  83.0:  64.75:  95.0:  87.5:   74.00:  97.0:
venusiax:Venusia  :VENU13097708 :  83.0:  55.00:  95.0:  87.5:   41.50:  97.0:
phenicia:Phenicia :PHEN14100790 :  94.0:  50.50:  95.0:  95.0:   59.50:  99.0:
mizarxxx:Mizar    :MIZA22128106 :  98.0:  95.50:  95.0: 100.0:   80.50:  90.0:
procyonx:Procyon  :PROC01033701 :  99.0:  99.50:100.0:  95.0:   98.00:100.0:
flamxxxx:Flam     :FLAM26077904 :  89.0:  37.25:  97.5:  87.5:   63.50:  97.0:
johannxx:johann   :JOHA23058000 :  94.0:  99.25:100.0:  95.0:   36.00:100.0:
wrightxx:Wright   :WRIG21058103 :  99.0:  100.0:100.0: 100.0:   100.0:  97.0:
cragxxxx:Grag     :CRAG05018002 :  99.0:  90.00:  90.0: 100.0:   97.00:  99.0:
malaxxxx:Mala     :MALA09557914 :  96.0:  69.50:100.0: 100.0:   34.00:  47.0:
```

On vous demande de créer un script permettant de générer automatiquement le fichier LISTEMOY dans lequel les différentes moyennes des étudiants sont calculées (avec la pondération suivante ; 30% pour l'intra, 40% pour le final et 30% pour l'ensemble des 4 Tps), c'est à dire le fichier

```
:NOM Prenom:CODE               :.Tp1 :.Intra:.Tp2 :.Tp3 :.Final :.TP4 :.Moy:.Lettre:

actarusx:Actarus :ACTA28658105 : 45.5: 75.00:100.0: 95.0: 48.50: 90.0: 66.687 :
alcorxxx:Alcor :ALCO18083408 : 83.0: 64.75: 95.0: 87.5: 74.00: 97.0: 76.207 :
venusiax:Venusia :VENU13097708 : 83.0: 55.00: 95.0: 87.5: 41.50: 97.0: 60.287 :
phenicia:Phenicia :PHEN14100790 : 94.0: 50.50: 95.0: 95.0: 59.50: 99.0: 67.675 :
mizarxxx:Mizar :MIZA22128106 : 98.0: 95.50: 95.0:100.0: 80.50: 90.0: 89.575 :
procyonx:Procyon :PROC01033701 : 99.0: 99.50:100.0: 95.0: 98.00:100.0: 98.600 :
flamxxxx:Flam :FLAM26077904 : 89.0: 37.25: 97.5: 87.5: 63.50: 97.0: 64.394 :
johannxx:johann :JOHA23058000 : 94.0: 99.25:100.0: 95.0: 36.00:100.0: 73.345 :
wrightxx:Wright :WRIG21058103 : 99.0: 100.0:100.0:100.0: 100.0: 97.0: 99.700 :
cragxxxx:Grag :CRAG05018002 : 99.0: 90.00: 90.0:100.0: 97.00: 99.0: 94.900 :
malaxxxx:Mala :MALA09557914 : 96.0: 69.50:100.0:100.0: 34.00: 47.0: 60.175 :
```

Le script devra s'appeler Moy.sh et devra passer en paramètre le fichier LISTE (pour une exécution du style Moy.sh LISTE)[3].

Remise & Rapport

Ce rapport devra être fait par groupe de deux étudiants. Vous devez rendre ce travail au démonstrateur le jour de la démonstration, i.e., le vendredi, à la date de remise spécifiée dans le fichier *planning* situé sur la page web du cours dans la rubrique *Introduction & Programme*. Ce rapport sera en fait constitué des trois programmes SCRIPT et seront remis grâce à la commande Remise. Il y aura une pénalité de 20% par jour de retard.

3. Les commandes script et UNIX intéressantes pour l'élaboration de ce script sont entre autre ; echo, head, >, let, wc -l, while, tail, » ... Bien sur d'autres commandes ou un sous ensemble de celle proposé sont possibles. Je rappelle que si line=actarusx :Actarus : ACTA28658105 : 45.5 : 75.00 : , la commande tp1='echo $line | cut -d : -f4' permettra de mettre 45.5 dans la variable tp1.