



DIRO
IFT 1214

EXAMEN FINAL

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle, local 2377

Http : [//www.iro.umontreal.ca/~mignotte/ift1214/](http://www.iro.umontreal.ca/~mignotte/ift1214/)

E-mail : mignotte@iro.umontreal.ca

Date : 27/04/2004

I	Assembleur/LMC (30 pts).
II	Programmation Shell Script (22 pts).
III	Codes correcteurs (22 pts).
IV	Misc. (28 pts).

Directives

- TOUTE DOCUMENTATION ET CALCULATRICE PERSONNELLE EST PERMISE
- Les réponses devront être clairement présentées et justifiées (elles peuvent être concises mais devront néanmoins contenir les résultats intermédiaires nécessaires permettant de montrer sans ambiguïté que vous êtes arrivés au résultat demandé).
- Si vous ne comprenez pas une question, faites en une interprétation, et proposez une réponse.

I. Assembleur/LMC (30 pts)

1. Programmation d'un Calcul d'un factoriel

Implémenter en code d'instructions mnémoniques, avec le jeux d'instruction donné plus bas correspondant au modèle LMC vu en cours, un programme permettant de calculer le factoriel d'un nombre n . On supposera que la valeur entière $n > 0$ dont on désire calculer le factoriel est entrée au début du programme par l'instruction IN et que le résultat est communiqué en sortie par l'instruction OUT.

<15 pts> (Rappel : n factoriel, noté $n!$ = $n(n-1)(n-2) \times \dots \times 1$. Exemple ; $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$.)

On utilisera le jeux d'instruction présenté dans le chapitre LITTLE MAN COMPUTER (cf. Fig.) et auquel on ajoutera l' instruction suivante :

MUL xx Multiplie le contenu de la case mémoire xx au nombre stocké dans l'accumulateur (registre du calculateur). Le résultat de la multiplication est stocké dans l'accumulateur.

LDA	5xx	Load
STO	3xx	Store
ADD	1xx	Add
SUB	2xx	Subtract
IN	901	Input
OUT	902	Output
COB or HLT	000	Coffee break (or Halt)
BRZ	7xx	Branch if zero
BRP	8xx	Branch if positive or zero
BR	6xx	Branch unconditional
DAT		Data storage location

2. Langage LMC et registres

Voici un programme en code d'instructions mnémoniques,

```
          LDA  C1
L1        OUT
          SUB  C2
          STO  TEMP
          BRZ  END
          BR   L1
END       HLT
C1        DAT  100
C2        DAT  2
TEMP      DAT  0
```

- Indiquez ce que ce programme va faire. <5 pts>
- Convertir ce programme en Code machine (codes d'opérations) en logeant ce programme à partir de l'adresse mémoire 00. <5 pts>
- Donner la valeur des différents registres ACCU, IR, PC, MAR, MDR à la fin des instructions **STO TEMP** et **BRZ END** et à la première itération de ce programme (i.e., à la première exécution de ces instruction). <5 pts>

Réponse

- Un exemple possible, valable $\forall n > 0$,

```

                IN
                STO RES
L1             SUB UN
                BRZ END
                STO CPT
                MUL RES
                STO RES
                LDA CPT
                BR L1
END           LDA RES
                OUT
                HLT

UN            DAT 1
RES          DAT 1
CPT          DAT 0

```

<15 pts>

BONUS <+1 pt>

si le cas 0! = 1 est traité ; pour cela ajouter avant la deuxième ligne BRZ END

2a

Le programme va afficher en sortie 100, 98, 96, ..., 4, 2 puis s'arrêter. <5 pts>

2b

La conversion en code machine est immédiate,

```

00 507
01 902
02 208
03 309
04 706
05 601
06 000
07 100
08 002
09 000

```

<5 pts>

2c

	ACCU	IR	PC	MAR	MDR
Fin de STO TMP	098	309	004	009	098
Fin de BRZ END	098	706	005	05	706
	<1 pt>				

NOTA : IL EST AUSSI JUSTE DE DIRE QUE MAR=04 POUR L'INSTRUCTION **BRZ END** SI ON PRÉCISE QUE C'EST À LA FIN DE SON CYCLE FETCH.

II. Programmation Shell Script (22 pts)

1. Manipulation/Modification de Fichiers

Voici un fichier de note (appelé LISTE avec des noms d'étudiants fictifs) dans le format utilisé par l'administration pour la gestion des notes des étudiants.

```
: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 :WRI21058103 : 99.0: 100.0:100.0:100.0: 100.0: 97.0: 99.700 :
cragxxxx:Crag :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 :
```

On vous demande de créer un script MMOY.SH permettant d'estimer la moyenne des moyennes des étudiants et le nombre d'étudiants. Ce script devra passer en paramètre le fichier LISTE (pour une exécution du style MMOY.SH LISTE et un affichage du style > MOYENNE DE LISTE = 77.4131818 POUR 11 ETUDIANTS) et devra bien sur fonctionner pour tout autre fichier utilisant le même format ¹ mais avec un nombre d'étudiants différents. <15 pts>

2. Recherche de Fichiers

On vous demande de créer un script FINDSTRGS.SH permettant de 1- rechercher dans le répertoire où vous le lancer ainsi que tous ses sous répertoire et sous-sous répertoire etc. tous les fichiers ayant une extension .tex, 2- de sauver le résultat de cette recherche dans un fichier tmp.file 3- finalement extraire dans cette liste les fichiers possédant à l'intérieur de ce fichier aussi une chaîne de caractère passée en argument à ce script (pour une exécution du style FINDSTRGS.SH STRING). ² <7 pts>

Réponse

1.

```
#!/bin/sh
# M Moy.sh script

# Resuperation Nom du fichier passé en paramètre
InputFilename=$1

# Calcul du Nombre d'étudiants
NbEtudiants='cat $InputFilename | wc -l'
let NbEtudiants=NbEtudiants-2

nb=1
cpt=2
Result=0

while [ $nb -le $NbEtudiants ]
do
# Extraction de chaque moy dans var. moy
line='head -n $cpt $InputFilename | tail -n 1'
moy='echo $line | cut -d: -f10'

# Somme partielle
Result='echo $Result + $moy | bc'

let cpt=cpt+1
let nb=nb+1
done

# Calcul Moyenne
Moyenne='echo $Result/$NbEtudiants | bc -l'

# Affichage
echo Moyenne de ${InputFilename}=$Moyenne pour $NbEtudiants etudiants
```

<15 pts>

¹Un format avec lequel la moyenne des étudiants sera toujours en 10ième champ. On supposera qu'il y a toujours le bon nombre d'arguments à ce script Appeler aussi le calculateur bc avec l'option -l pour pouvoir faire des divisions non entière, exemple :

```
VAR1=5.5 VAR2=3.14159
```

```
RESULT='echo $VAR1/$VAR2 | bc -l'
```

```
echo $RESULT
```

qui permettrait d'afficher à l'écran le résultat espéré, c'est à dire > 1.750705853.

²On supposera qu'il y a toujours le bon nombre d'arguments à ce script, et que le script affiche à l'écran le nom du fichier et la ligne de ce fichier contenant la chaîne de caractère spécifiée. (Nota : Suivant la stratégie utilisée, et les commandes utilisées, ce script peut s'écrire entre 1 et 12 lignes).

BARÈME DES 15 PTS :
 Récup. de l'argument : 2 Pts
 Récup. du Nb. Etudiants/Nb lignes : 2 Pts
 Extraction de chaque Moy : 3 Pts
 Somme partielle : 2 Pts
 Affichage : 2 Pts
 Recup chaque ligne & Reste : 4 Pts

2. Version longue et version courte

```
#!/bin/sh
# FindStrgs Script

# Recuperation parametres
String1=$1

# Creation liste de fichiers
# & recuperation nb de fichiers
find -name \*.tex > tmp.file
Nbfiles=`cat tmp.file | wc -l`

# Pour chaque fichier recherche
cpt=1
while [ $cpt -lt $Nbfiles ]
do
  file=`head -n $cpt tmp.file | tail -n 1`
  if grep $String1 $file
  then
    echo $file
  fi
  let cpt=cpt+1
done
```

```
#!/bin/sh
# Find2Strgs Script

# Recherche
find -name \*.tex > tmp.file | xargs grep $1
```

<7 pts>

BARÈME DES 7 PTS :
 Récup. de l'argument : 2 Pts
 Creation Liste de fichier : 2 Pts
 Recherche pour chaque fichier : 2 Pts
 Reste : 1 Pts

III. Codes correcteurs (22 pts)

1. Codage de Hamming

On reçoit le message suivant 23556_8 qui est encodé avec un code de Hamming utilisant 4 bits de contrôle et une parité paire. Retrouver la séquence originale en base 2 et octal en corrigeant les erreurs si nécessaire et indiquer ensuite en binaire quel est le message corrigé. <7 pts>

2. Codage CRC

- On veut transmettre le message 111_8 en utilisant la méthode CRC avec le polynôme générateur $G(x) = x^5 + x^3 + 1$. Trouver le message à transmettre et le mettre en octal. <10 pts>
- On reçoit le message 4461_8 utilisant un code CRC avec le polynôme générateur $G(x) = x^5 + x^3 + 1$. Le message a-t-il été corrompu? <5 pts>

Réponse

1.

On convertit d'abord le message en binaire.

$$23556_8 \blacktriangleright 10\ 011\ 101\ 101\ 110_2 \quad \langle 1 \text{ pt} \rangle$$

Avec la méthode de Hamming simplifiée qui consiste à faire l'addition modulo 2 des positions de ce message, on obtient ;

$$\begin{array}{r}
 14 \rightarrow 1110 \\
 11 \rightarrow 1011 \\
 10 \rightarrow 1010 \\
 09 \rightarrow 1001 \\
 07 \rightarrow 0111 \\
 06 \rightarrow 0110 \\
 04 \rightarrow 0100 \\
 03 \rightarrow 0011 \\
 02 \rightarrow 0010 \\
 \hline
 0010 \quad \langle 4 \text{ pts} \rangle
 \end{array}$$

Cette addition nous indique que le bit à la position $0010_2 = 2_{10}$ est erroné. Le message corrigé est $10\ 011\ 101\ 101\ 100_2 = 23554_8 \langle 1 \text{ pt} \rangle$. En enlevant les bits de contrôle (position 1, 2, 4, 8), on obtient la message binaire informatif suivant ; $0100111101_2 = 1175_8. \langle 1 \text{ pt} \rangle$

2a.

- On convertit d'abord le message en binaire.

$$111_8 \blacktriangleright 1001001_2 \quad \langle 1 \text{ pt} \rangle$$

- On réalise ensuite la division modulo 2 suivante,

$$\begin{array}{r}
 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \quad | \quad 1\ 0\ 1\ 0\ 0\ 1 \\
 \hline
 1\ 0\ 1\ 0\ 0\ 1 \quad | \quad 1\ 0\ 1\ 1\ 1\ 0\ 0 \\
 \hline
 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0 \quad | \\
 \quad 1\ 0\ 1\ 0\ 0\ 1 \quad | \\
 \quad \hline
 \quad 1\ 1\ 1\ 1\ 1\ 0 \quad | \\
 \quad 1\ 0\ 1\ 0\ 0\ 1 \quad | \\
 \quad \hline
 \quad 1\ 0\ 1\ 1\ 1\ 0 \quad | \\
 \quad 1\ 0\ 1\ 0\ 0\ 1 \quad | \\
 \quad \hline
 \quad \quad 1\ 1\ 1\ 0\ 0 \quad |
 \end{array}
 \quad \langle 7 \text{ pts} \rangle$$

- On transmet le message $\underbrace{1001001}_{M(x)} \underbrace{11100}_{R(x)} = 4474_8. \quad \langle 2 \text{ pts} \rangle$

2b.

- On réalise ensuite la division modulo 2 suivante,

$$\begin{array}{cccccccc|cccc}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & | & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & & & & & & & | & 1 & 0 & 1 & 1 & 1 & 0 \\
- & - & - & - & - & - & & & & & & & | & & & & & & & \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & & & & & | & & & & & & & \\
& & & 1 & 0 & 1 & 0 & 0 & 1 & & & & | & & & & & & & \\
& & & & - & - & - & - & - & - & & & | & & & & & & & \\
& & & & & 1 & 1 & 1 & 1 & 0 & 0 & & | & & & & & & & \\
& & & & & 1 & 0 & 1 & 0 & 0 & 1 & & | & & & & & & & \\
& & & & & & - & - & - & - & - & - & | & & & & & & & \\
& & & & & & & 1 & 0 & 1 & 0 & 1 & 0 & | & & & & & & & \\
& & & & & & & 1 & 0 & 1 & 0 & 0 & 1 & | & & & & & & & \\
& & & & & & & & - & - & - & - & - & | & & & & & & & \\
& & & & & & & & & & & 1 & 1 & 0 & 1 & | & & & & & &
\end{array}$$

Le reste $R(x) = 1101$ est non nul donc le message est corrompu. <5 pts>

IV. Misc. (28 pts)

Répondez aux question suivantes ;

1. CPU & Mémoire

- (a) Quelles sont les avantages/inconvénients d'un séquenceur micro-programmés sur un séquenceur câblé. <3 pts>
- (b) Un système informatique à un *hit ratio* de 95 et un temps d'accès de cette mémoire cache de 100 ns et un temps d'accès de 800 ns lors d'un *cache miss* (défaut de bloc). Quel est le temps d'accès moyen de ce système. <3 pts>
- (c) Que veut dire techniquement l'anglicisme *swapper* pour un système informatique? <2 pts>

2. Cache

- (a) Supposons qu'une adresse en mémoire virtuelle paginée, nécessite 20 bits organisés de la façon suivante :

$$\underbrace{19, \dots, 12}_{\text{page}} \underbrace{11, \dots, 0}_{\text{offset}}$$

Quelle est la taille de cette mémoire virtuelle, exprimées en nombre de mots, nombre de pages et nombre de mots par page. <3 pts>

- (b) Quel est l'adresse binaire du 2 ième mot de la page 213? <3 pts>

3. Input & Output -Périphériques

- (a) Un lecteur de disquette tourne à 300 rpm. Ce disque est divisé en 12 secteurs et comprend 40 pistes (*tracks*) (1 seule face du disque est utilisable). Chaque piste peut contenir un bloque de 200 octets. Calculer la capacité de ce disque, le temps de latence minimum, le temps de latence maximum et le temps de transfert pour un bloc de 200 octets. <3 pts>

(b) Que veut dire dé-fragmenter pour un disque ? et l'intérêt d'une dé-fragmentation. <3 pts>

4. Structure d'un OS

(a) Donner deux exemples de services faisant partie du noyau (*kernel*) du système d'exploitation et n'en faisant pas partie. <2 pts>

5. TéléInformatique

(a) Expliquer quelle est l'avantage de la modulation pour la transmission. <3 pts>

(b) Quelle est l'adresse URL d'un document dont le nom de fichier est Exemple.html, situé dans le répertoire /home/users/dupond, sur la machine www.obelix.ch et accessible à travers le protocole ftp? <3 pts>

Réponse

1. (a) L'avantage d'un séquenceur micro-programmé réside 1- dans sa souplesse et sa simplicité (on peut rajouter des instructions au jeux d'instructions très facilement ou emuler un jeux d'instruction d'une autre machine très facilement). 2- Cela simplifie considérablement l'architecture du CPU (le CPU est allégé du cablage logique des instructions). L'inconvénient est 1- qu'il est plus lent qu'un séquenceur câblé. <3 pts>

(b) Tps d'accès moyen = $0.95(100)ns + 0.05(800)ns = 135ns$ <3 pts>

(c) Cela vient de l'anglais swap, qui veut dire échanger. Cela réfère à un système informatique qui est obligé d'échanger souvent de l'information entre la mémoire vive et la mémoire cache (i.e., qu'il y a souvent de défaut de blocs). Le système s'en trouve ralentit. <2 pts>

2. (a) 2^8 pages contenant chacune 2^{12} mots pour un total de 2^{20} mots. <3 pts>

(b)

$$\underbrace{11010101}_{2^{13}_{10}} \underbrace{000000000001}_{1_{10}}$$

<3 pts>

Attention, le nième élément se trouve à l'adresse $n - 1$ puisque l'on compte à partir de zéro.

3. (a)

• Capacité du disque = 40 (pistes) \times 200 octets = 8000 octets. <1 pt> ³

• Le temps de latence maximal est le temps que met le disque à faire une rotation entière ($T_{max} = 1/300$ minutes ou 1/5 secondes. <0.5 pt>

• Le temps de latence minimale est $T_{min} = 0$. <0.5 pt>

• Une rotation complète du disque prend 1/5 secondes pendant lequel il peut lire 12 secteurs ou une piste (un bloc). <1 pt>

(b) C'est le processus qui va re-arranger les données sauvegardées sur le disque pour les arrangées de façon le plus continue possible dans le but de sauver du temps d'accès à un programme.

<3 pts>

³Nota : Ici le nombre de secteurs ne servait pas à calculer la capacité du disque car on spécifiait bien dans l'énoncé que c'était chaque piste qui était constitué d'un bloque de 200 octets et non pas chaque secteur.

4. (a) cd, mkdir, ls, (toute commande liées à la gestion des fichiers), toute commande lié à la gestion du protocole TCP/IP, et de la gestion de la sécurité/droits d'accès, ... ∈ kernel <1 pt>

formatage d'un disque, dé-fragmentation du disque, anti-virus, compilateur, etc. ∉ kernel
<1 pt>

5. (a) La modulation permet de convertir le message à transmettre en un message dont les fréquences appartienne à la bande passante du canal de transmission. Celle-ci est surtout utilisé dans le cas d'un bande passante faible du canal de transmission. <3 pts>

(b) ftp ://www.obelix.ch/home/users/dupond/Exemple.html <3 pts>