Université de Bordeaux — U.F.R. Mathématiques et Informatique
Licence Informatique 2ème année — Année 2015 – 2016

ARCHITECTURE DES ORDINATEURS

Devoir Surveillé 1

1 heure 20 minutes
sans documents

N.B. : - Travaillez calmement. Ne bâclez pas vos réponses : il vaut mieux traiter correctement moins de questions que tout faire de travers.
- Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Question 1 (6 points)

On considère des entiers codés sur 16 bits, interprétés comme des entiers relatifs codés en complément à deux. Chacun de ces entiers peut (et doit) s'écrire avec quatre chiffres hexadécimaux (on omettra le préfixe « 0x »). Soient les cinq entiers suivants :

M	N	P	Q	R
cad0	6789	dede	9006	3210

(1.1) (1 point)
Parmi ces 5 entiers, quels sont ceux qui sont négatifs ? Expliquez pourquoi.

(1.2) (1 point)
Classez ces entiers du plus petit au plus grand (par exemple : « M < N < P < Q < R », mais cette réponse choisie arbitrairement n'est probablement pas la bonne...). Justifiez la réponse sur votre copie.

(1.3) (2 points)
Calculez $N + P$. Pour cela, effectuez l'addition en binaire directement sur la copie — une réponse brute ne rapportera aucun point. Y a-t-il débordement (*overflow*) ?

(1.4) (2 points)
Calculez $-M$ et $N - M$. Comme pour la question précédente, le détail des calculs doit figurer sur la copie.

Question 2 (14 points)

Vous trouverez plus bas un programme écrit en langage Y86. Tous les commentaires ayant malencontreusement été effacés, il vous faut partir de zéro pour comprendre ce que fait ce programme, en répondant aux questions ci-dessous.

Attention :
— *pas de panique, inutile de commencer par lire en détail ce programme ;*
— *beaucoup de questions sont indépendantes ;*
— *les réponses qui, au lieu d'expliquer, paraphrasent simplement le code (telles que « on soustrait %edx à %ebx ») ne rapporteront aucun point.*

On va tout d'abord s'intéresser à la fonction principale.

(2.1) (1 point)
Combien de paramètres la fonction `f` semble-t-elle prendre ? Renvoie-t-elle une valeur ? Expliquer quelle(s) partie(s) du code permet(tent) de répondre à ces questions.

(2.2) (1 point)
Quel est le sens des instructions `irmovl` situées aux adresses `0x008` et `0x010` ? En langage C, quel serait le type des paramètres passés à la fonction `f` ?

Intéressons-nous maintenant à la fonction `g`.

(2.3) (1 point)

Combien de paramètres la fonction `g` semble-t-elle prendre ? Renvoie-t-elle une valeur ? Expliquer quelle(s) partie(s) du code permet(tent) de répondre à ces questions.

Intéressons-nous ensuite à la fonction `f`.

(2.4) (2 points)

Dessinez l'état de la pile après la fin de l'instruction de l'adresse `0x02c`. Indiquez la position du pointeur de pile, écrivez les valeurs numériques ayant déjà été empilées dans chacune des cellules de la pile, et indiquez à côté de chaque cellule le décalage par rapport à `%esp` permettant d'y accéder.

(2.5) (2 points)

Quel est le rôle de l'instruction de l'adresse `0x053` ? Pourquoi a-t-elle été placée ici et non dans la fonction `g` ? Quelle instruction lui correspond-elle après le `call g` ? Pourquoi ?

(2.6) (1 point)

Quel est le rôle de l'instruction à l'adresse `0x055` ? Quelle instruction lui correspond-elle après le `call g` ? Pourquoi ?

Intéressons-nous de nouveau à la fonction `g`.

(2.7) (1 point)

Quel est le rôle de l'instruction située à l'adresse `0x08e` ? Pourquoi a-t-elle été placée ici et non dans la fonction `f` ?

(2.8) (1 point)

Expliquez ce que fait la fonction `g`. Donnez son prototype.

Et terminons par la fonction `f`.

(2.9) (2 points)

Que fait la fonction `f` ? Quelles modifications apporte-t-elle aux variables globales situées à partir de l'adresse `0x104` ? Combien d'appels à la fonction `g` ont-ils lieu ?

(2.10) (1 point)

En vous basant sur le codage de certaines instructions (dites lesquelles), quels sont les numéros sur 4 bits correspondant aux registres `%ecx` et `%esi` ?

(2.11) (1 point)

Par quelle instruction plus courte aurait-on pu remplacer l'instruction de l'adresse `0x03a` ?

```
0x000:              |          .pos    0
0x000: 308400020000 | main:    irmovl  pile,%esp
0x006: 2045         |          rrmovl  %esp,%ebp
0x008: 308014010000 |          irmovl  t2,%eax
0x00e: a008         |          pushl   %eax
0x010: 308004010000 |          irmovl  t1,%eax
0x016: a008         |          pushl   %eax
0x018: 802a000000   |          call    f
0x01d: c08408000000 |          iaddl   8,%esp
0x023: 400800010000 |          rmmovl  %eax,t0
0x029: 10           |          halt
                    |
0x02a: a068         | f:       pushl   %esi
0x02c: a078         |          pushl   %edi
0x02e: 50640c000000 |          mrmovl  12(%esp),%esi
0x034: 507410000000 |          mrmovl  16(%esp),%edi
0x03a: 308000000000 |          irmovl  0,%eax
0x040: 501600000000 | f1:      mrmovl  (%esi),%ecx
0x046: 502700000000 |          mrmovl  (%edi),%edx
0x04c: 6112         |          subl    %ecx,%edx
0x04e: 737d000000   |          je      f2
0x053: a008         |          pushl   %eax
0x055: a068         |          pushl   %esi
0x057: a078         |          pushl   %edi
0x059: 8082000000   |          call    g
0x05e: c08408000000 |          iaddl   8,%esp
0x064: b008         |          popl    %eax
0x066: c08604000000 |          iaddl   4,%esi
0x06c: c08704000000 |          iaddl   4,%edi
0x072: c08001000000 |          iaddl   1,%eax
0x078: 7040000000   |          jmp     f1
0x07d: b078         | f2:      popl    %edi
0x07f: b068         |          popl    %esi
0x081: 90           |          ret
                    |
0x082: 501404000000 | g:       mrmovl  4(%esp),%ecx
0x088: 502408000000 |          mrmovl  8(%esp),%edx
0x08e: a038         |          pushl   %ebx
0x090: 500100000000 |          mrmovl  (%ecx),%eax
0x096: 503200000000 |          mrmovl  (%edx),%ebx
0x09c: 403100000000 |          rmmovl  %ebx,(%ecx)
0x0a2: 400200000000 |          rmmovl  %eax,(%edx)
0x0a8: b038         |          popl    %ebx
0x0aa: 90           |          ret
                    |
0x100:              |          .pos    0x100
0x100: 00000000     | t0:      .long   0
0x104: 13000000     | t1:      .long   19
0x108: 1b000000     |          .long   27
0x10c: 0c000000     |          .long   12
0x110: 2a000000     |          .long   42
0x114: 07000000     | t2:      .long   7
0x118: 33000000     |          .long   51
0x11c: 22000000     |          .long   34
0x120: 2a000000     |          .long   42
                    |
0x200:              |          .pos    0x200
0x200: 00000000     | pile:    .long   0
```