Université de Bordeaux — U.F.R. Mathématiques et Informatique
Licence Informatique 2ème année — Année 2013 – 2014

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
1010	cafe	4242	babe	9009

(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) (2 points)
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. Donnez le prototype de la fonction `g`.

(2.4) (1 point)

Pourquoi la fonction `g` ne contient-elle pas en son début la séquence d'instructions « `pushl %ebp` / `rrmovl %esp,%ebp` » que l'on trouve par exemple au début de la fonction `f` ?

Intéressons-nous ensuite à la fonction `f`.

(2.5) (1 point)

Quel est le rôle des instructions commençant aux adresses `0x028` et `0x02a` ? Pourquoi ces instructions sont-elles nécessaires ?

(2.6) (1 point)

Combien de fois effectue-t-on la boucle de corps commençant en `f1` ? À quelle condition cette boucle s'arrête-t-elle ?

(2.7) (1 point)

Quel(s) paramètre(s) la fonction `f` passe-t-elle successivement à la fonction `g` à chaque tour de boucle ?

(2.8) (2 points)

Que fait la fonction `f` ? Quelles modifications apporte-t-elle aux variables globales situées à partir de l'adresse `0x100` ?

(2.9) (1 point)

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

(2.10) (1 point)

Par quelle instruction plus courte aurait-on pu remplacer l'instruction de l'adresse `0x03e` ?

Et pour finir, retour à la fonction `g`.

(2.11) (2 points)

Expliquez ce que fait la fonction `g`. Pour vous aider, vous pouvez prendre comme exemples d'étude les valeurs qui lui sont passées par la fonction `f`.

```
0x000:              |          .pos    0
0x000: 308400020000 | main:    irmovl  pile,%esp
0x006: 2045         |          rrmovl  %esp,%ebp
0x008: 308018010000 |          irmovl  t2,%eax
0x00e: a008         |          pushl   %eax
0x010: 308000010000 |          irmovl  t1,%eax
0x016: a008         |          pushl   %eax
0x018: 8024000000   |          call    f
0x01d: 608408000000 |          iaddl   8,%esp
0x023: 10           |          halt
                    |
0x024: a058         | f:       pushl   %ebp
0x026: 2045         |          rrmovl  %esp,%ebp
0x028: a068         |          pushl   %esi
0x02a: a078         |          pushl   %edi
0x02c: 506508000000 |          mrmovl  8(%ebp),%esi
0x032: 50750c000000 |          mrmovl  12(%ebp),%edi
0x038: 501600000000 | f1:      mrmovl  (%esi),%ecx
0x03e: 618100000000 |          isubl   0,%ecx
0x044: 7375000000   |          je      f2
0x049: 502604000000 |          mrmovl  4(%esi),%edx
0x04f: a028         |          pushl   %edx
0x051: a018         |          pushl   %ecx
0x053: 807e000000   |          call    g
0x058: 608408000000 |          iaddl   8,%esp
0x05e: 400700000000 |          rmmovl  %eax,(%edi)
0x064: 608608000000 |          iaddl   8,%esi
0x06a: 608704000000 |          iaddl   4,%edi
0x070: 7038000000   |          jmp     f1
0x075: b078         | f2:      popl    %edi
0x077: b068         |          popl    %esi
0x079: 2054         |          rrmovl  %ebp,%esp
0x07b: b058         |          popl    %ebp
0x07d: 90           |          ret
```

```
0x07e: 500404000000 | g:       mrmovl  4(%esp),%eax
0x084: 502408000000 |          mrmovl  8(%esp),%edx
0x08a: 2021         | g1:      rrmovl  %edx,%ecx
0x08c: 6101         |          subl    %eax,%ecx
0x08e: 73a6000000   |          je      g3
0x093: 729f000000   |          jl      g2
0x098: 2012         |          rrmovl  %ecx,%edx
0x09a: 708a000000   |          jmp     g1
0x09f: 6120         | g2:      subl    %edx,%eax
0x0a1: 708a000000   |          jmp     g1
0x0a6: 90           | g3:      ret
                    |
0x100:              |          .pos    0x100
0x100: 0f000000     | t1:      .long   15
0x104: 28000000     |          .long   40
0x108: 0c000000     |          .long   12
0x10c: 12000000     |          .long   18
0x110: 00000000     |          .long   0
0x114: 00000000     |          .long   0
0x118: 00000000     | t2:      .long   0
0x11c: 00000000     |          .long   0
                    |
0x200:              |          .pos    0x200
0x200: 00000000     | pile:    .long   0
```