
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 (5 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
9876	F00D	3945	C0DE	1418

- (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) (1 point)
Calculez $N + Q$. 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 $-N$ et $M - N$. Y a-t-il débordement (*overflow*) ? Comme pour la question précédente, le détail des calculs doit figurer sur la copie.

Question 2 (2 points)

Exprimer par une proposition logique les propositions suivantes, opérant sur les quatres variables logiques A, B, C et D. Chaque proposition logique doit donc renvoyer 1 si l'assertion est vraie, et 0 sinon :

- (2.1) Les variables A, B, C, D sont toutes égales à 1.
(2.2) Toutes les variables A, B, C, D sont nulles.
(2.3) Au moins l'une des variables A, B, C ou D est égale à 1.
(2.4) Au moins l'une des variables A, B, C, D est égale à 0.

Question 3 (9 points)

Voici le texte d'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 : les réponses qui, au lieu d'expliquer, paraphrasent simplement le code (telles que « on soustrait %edx à %ebx ») ne rapporteront aucun point.

0x000:		.pos	0	(3.1)	(1 point)
0x000: 308400020000	main:	irmovl	pile,%esp		Quels sont les paramètres de la fonction <code>mystere</code> ? Expliquez quelle partie du code permet de répondre à cette question.
0x006: 308004000000		irmovl	4,%eax		
0x00c: a008		pushl	%eax		
0x00e: 308000010000		irmovl	t,%eax		
0x014: a008		pushl	%eax		
0x016: 8022000000		call	mystere	(3.2)	(6 points)
0x01b: 608408000000		iaddl	8,%esp		Que calcule la fonction <code>mystere</code> , et que renvoie-t-elle? Pour cela :
0x021: 10		halt			— Expliquez le rôle de chacun des registres utilisés au sein de cette fonction et leur évolution au cours du déroulement de la fonction;
0x022: 502404000000	mystere:	rrmovl	4(%esp),%edx		— Indiquez à quoi sert, dans le cas de cette fonction, le test de l'adresse 0x059;
0x028: 501408000000		rrmovl	8(%esp),%ecx		— Indiquez à quoi sert, dans le cas de cette fonction, l'instruction d'adresse 0x05e;
0x02e: 618101000000		isubl	1,%ecx		— Indiquez à quelles modifications en mémoire centrale ce programme conduit, dans la zone comprise entre 0x100 et 0x120.
0x034: 7166000000		jle	et2	(3.3)	(1 point)
0x039: 500200000000		rrmovl	(%edx),%eax		À quoi le test de l'adresse 0x034 sert-il?
0x03f: a008		pushl	%eax		
0x041: 500204000000	et1:	rrmovl	4(%edx),%eax		
0x047: 400200000000		rrmovl	%eax,(%edx)		
0x04d: 608204000000		iaddl	4,%edx		
0x053: 618101000000		isubl	1,%ecx		
0x059: 7641000000		jg	et1	(3.4)	(1 point)
0x05e: b008		popl	%eax		Si ce code avait été produit par un compilateur C, quel aurait été le prototype de la fonction <code>mystere</code> ?
0x060: 400200000000		rrmovl	%eax,(%edx)		
0x066: 90	et2:	ret			
0x100:		.pos	0x100		
0x100: 02000000	t:	.long	2		
0x104: 07000000		.long	7		
0x108: 01000000		.long	1		
0x10c: 03000000		.long	3		
0x200:		.pos	0x200		
0x200: 00000000	pile:	.long	0		

Question 4 (4 points)

Écrivez le code x86 d'un programme qui implémente la fonction C de prototype suivant :

```
void swap (int * p, int * q);
```

qui effectue l'échange des deux valeurs pointées par `p` et `q`. Ainsi, si à l'entrée de la fonction on avait `*p = x` et `*q = y`, alors au retour de la fonction on doit avoir `*p = y` et `*q = x`.

N.B. : Veillez au bon usage des registres que vous employez !