
ARCHITECTURE DES ORDINATEURS

Devoir Surveillé 1

1 heure et 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.

Exercice 1 (50 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
4B29	71F2	8023	3FD6	CDBA

- (1.1) (10 points)
Parmi ces 5 entiers, quels sont ceux qui sont négatifs ? Expliquez pourquoi.
- (1.2) (10 points)
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) (10 points)
Calculez $M + P$ et donnez le résultat en hexadécimal. Pour cela, effectuez l'addition exclusivement en binaire et directement sur la copie — une réponse brute ne rapportera aucun point. Y a-t-il débordement (*overflow*) ?
- (1.4) (20 points)
Calculez $-Q$ et $P - Q$, et donnez les résultats en hexadécimal. Y a-t-il débordement (*overflow*) ? Comme pour la question précédente, le détail des calculs en binaire doit figurer sur la copie.

Exercice 2 (150 points)

La page suivante contient 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.

(2.1) (20 points)

Quels sont les paramètres de la fonction `mystere`? Expliquez quelle partie du code permet de répondre à cette question.

Si ce code avait été produit par un compilateur C, quel aurait été le prototype de la fonction `mystere`?

(2.2) (20 points)

Quel est le rôle de l'instruction d'adresse `0x026`? Pourquoi cette instruction prend-elle en paramètre l'entier 16?

(2.3) (10 points)

Quel est le rôle de l'instruction d'adresse `0x067`?

(2.4) (10 points)

Quelle est la valeur numérique qui sera stockée à l'adresse d'étiquette `x` à la fin de l'exécution du programme? À quoi correspond-elle? Justifiez votre réponse.

(2.5) (10 points)

Quelle serait la valeur retournée par la fonction `mystere` si les valeurs entières stockées à partir de l'adresse `t` étaient : [4, 3, 7, 0, 7, 5, 5]?

(2.6) (30 points)

Dans la fonction `mystere`, quels sont les rôles respectifs des différents registres utilisés? Justifiez vos réponses.

(2.7) (10 points)

Dans la fonction `mystere`, quels sont les rôles respectifs des étiquettes `myst1` et `myst2`?

(2.8) (10 points)

Dites, en une phrase, ce que fait la fonction `mystere`, en prenant en compte les connaissances obtenues des questions précédentes.

(2.9) (10 points)

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

(2.10) (20 points)

Expliquez le codage binaire des six octets de l'instruction d'adresse `0x049`, en tirant éventuellement parti des informations obtenues en considérant d'autres instructions.

0x000:		.pos 0
0x000:	308400020000	init: irmovl pile,%esp
0x006:	308074000000	irmovl t,%eax
0x00c:	a008	pushl %eax
0x00e:	8020000000	call mystere
0x013:	c08404000000	iaddl 4,%esp
0x019:	400870000000	rrmovl %eax,x
0x01f:	10	halt
0x020:	a038	mystere: pushl %ebx
0x022:	a068	pushl %esi
0x024:	a078	pushl %edi
0x026:	503410000000	mrmovl 16(%esp),%ebx
0x02c:	3080ffffff	irmovl -1,%eax
0x032:	501300000000	mrmovl (%ebx),%ecx
0x038:	6211	andl %ecx,%ecx
0x03a:	7367000000	je myst2
0x03f:	2002	rrmovl %eax,%edx
0x041:	2017	myst1: rrmovl %ecx,%edi
0x043:	c08201000000	iaddl 1,%edx
0x049:	c08304000000	iaddl 4,%ebx
0x04f:	501300000000	mrmovl (%ebx),%ecx
0x055:	6211	andl %ecx,%ecx
0x057:	7367000000	je myst2
0x05c:	2016	rrmovl %ecx,%esi
0x05e:	6176	subl %edi,%esi
0x060:	7441000000	jne myst1
0x065:	2020	rrmovl %edx,%eax
0x067:	b078	myst2: popl %edi
0x069:	b068	popl %esi
0x06b:	b038	popl %ebx
0x06d:	90	ret
0x070:		.align 4
0x070:	00000000	x: .long 0
0x074:	02000000	t: .long 2
0x078:	03000000	.long 3
0x07c:	01000000	.long 1
0x080:	07000000	.long 7
0x084:	07000000	.long 7
0x088:	03000000	.long 3
0x08c:	05000000	.long 5
0x090:	00000000	.long 0
0x094:	08000000	.long 8
0x098:	02000000	.long 2
0x200:		.pos 0x200
0x200:		pile: