INF1600

Travail pratique 2

Architecture IA-32 et ASM

Département de Génie Informatique et Génie Logiciel
Polytechnique Montréal

1 Introduction et sommaire

Ce travail pratique a pour but de vous familiariser avec les instructions d'assembleur IA-32 selon la syntaxe AT&T. Vous aurez à manipuler du calcul arithmétique basé sur les propriétés de la suite de Fibonacci. Les problèmes considérés traitent l'adressage de la mémoire, la gestion de la pile et les instructions arithmétiques de division et multiplication.

1.1 Remise

Voici les détails concernant la remise de ce travail pratique :

- Méthode : sur Moodle, une seule remise par équipe, incluant un **rapport PDF** et les **fichiers sources** que vous modifiez en un seul fichier compressé.

- Format du rapport PDF : Incluez une page titre où doivent figurer les noms et matricules des deux membres de l'équipe, votre groupe de laboratoire, le nom et le sigle du cours, la date de remise et le nom de l'École. Dans une seconde page, incluez le barème de la section 1.2. Finalement, pensez à incluez les réponses aux questions et des captures d'écran si requis.

- Format des fichiers sources : Modifiez les fichiers assembleurs demandés dans les dossiers désignés à chaque question, puis compresser le tout en un seul fichier source <.Zip> que vous devez nommer comme suit :
  `<matricule1>-<matricule2>-<tp2>.<Zip>`

Attention :

L'équipe de deux que vous avez formé pour le TP1 est la même pour ce TP et la suite des TPs de cette session.

1.2 Barème

Les travaux pratiques 1 à 5 sont notés sur 4 points chacun, pour un total de 20/20. Le TP2 est noté selon le barème suivant. Reproduisez ce tableau dans le document PDF que vous remettrez.

Le TP2 comprend une question bonus de 0.5. Il est donc possible d'obtenir 4.5 pour ce travail pratique. Notez cependant que la note totale des travaux pratiques ne peut pas dépasser 20/20.

TP 2			**/4,00**
Partie 1			/2,00
	Q1	/0,25	
	Q2	/0,5	
	Q3	/0,25	
	Q4	/0,25	
	Q5	/0,5	
	Q6	/0,25	
Partie 2			/2,00
	Q1	/0,5	
	Q2	/0,5	
	Q3	/0,5	
	Q4	/0,5	
Bonus			/0,5

2 Mystères arithmétiques de la suite de Fibonacci

Dans ce travail pratique, nous allons voir de remarquables propriétés de la suite de Fibonacci. En guise de rappel, la suite de Fibonacci est une suite de nombres entiers dont chaque terme successif représente la somme des deux termes précédents, et qui commence par 0 puis 1. La suite commence ainsi :

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, \dots$$

F_n est le nième terme de cette suite qui débute par $n = 0$. Ainsi, la suite de Fibonacci $F_0, F_1, F_2, \dots$ est définie par la formule de récurrence :

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-2} + F_{n-1}\ ;\ \ n \in \mathbb{N}, n \geq 2$$

2.1 Partie 1 : La période de Pisano

La période de Pisano, nommée d'après *Leonardo Pisano*, mieux connu sous le nom de *Fibonacci*, a été découverte pour la première fois en 1774 par le mathématicien Lagrange. Ce dernier a remarqué l'existence de fonctions périodiques dans la suite de Fibonacci, obtenues en prenant le modulo de chaque terme pris pour une valeur entière donnée. Par exemple, la séquence modulo 4 de la suite de Fibonacci présente une période de 6 :

$$F_n = \ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, \dots$$

$$F_n[4] = \ 0, 1, 1, 2, 3, 1, 0, 1, 1, 2, 3, 1, 0, 1, 1, 2, 3, 1, 0, 1, ..$$

La fonction Pisano(*m*) donne la période de la séquence des nombres de Fibonacci, calculée modulo *m*. Ainsi, $Pisano(4) = 6$.

Q1/ À l'aide du jeu d'instructions IA-32, complétez le programme assembleur `fibo.s` qui doit calculer et écrire successivement en mémoire les $n - 1$ termes de la suite de Fibonacci : $F_2, F_3 \dots F_n$. L'écriture doit se faire à partir de l'adresse `adr_fibo` réservée dans la partie .data du même programme.

Notez bien que F_0, F_1 vous sont déjà disponibles à l'adresse mémoire désignée, et que la valeur de *n* est disponible dans le registre `%edi`.

Q2/ Soit $m \in \mathbb{N}\ ;\ m \geq 2$. Complétez les programmes assembleurs `fibo_mod_m_div.s` et `fibo_mod_m.s` qui doivent calculer les restes de la division des *n* nombres de Fibonacci par *m* (calculs modulo *m*), et les écrire successivement en mémoire à partir de l'adresse réservée `adr_fibo_mod_m`.

Notez bien que le programme `fibo_mod_m_div.s` doit impérativement utiliser l'instruction « `div` », tandis que le programme `fibo_mod_m.s` ne doit pas l'utiliser. Les valeurs de *n* et *m* sont disponibles dans les registres `%edi` et `%esi` respectivement.

Q3/ Pensez-vous que l'approche utilisée à la Q2 est efficace pour calculer le modulo d'un très grand nombre de Fibonacci? Justifiez votre réponse.

Q4/ Soit $m \in \mathbb{N}\ ;\ m \geq 2$. En se basant sur l'un de vos programmes de la Q2, quelle est la période de Pisano que vous remarquez pour $m = 2$, $m = 3$ et $m = 4$?

Examinez attentivement les séquences périodiques pour les valeurs précédentes de *m*. Que remarquez-vous? Quel est le trait commun entre ces séquences?

Q5/ Soit $m, n \in \mathbb{N} . m \geq 2$. En généralisant le constat précédent à tout *m*, complétez le programme assembleur `fibo_pisano_m.s` qui calcule la période de Pisano pour un entier *m*.

Notez bien que le programme devra mettre le résultat final dans le registre `%eax`. Les valeurs de *n* et *m* vous sont disponibles dans les registres `%edi` et `%esi` respectivement. Vous pouvez utiliser l'adresse mémoire réservé `adr_pisano` pour vos calculs.

Q6/ Soit $m, n \in \mathbb{N}$; $m \geq 2$. Complétez le programme assembleur `fibo_grand.s` qui calcule $F_n \; mod \; [m]$ pour un *n* très grand.

Notez bien que vous devez utiliser la périodicité de Pisano pour faire revenir *n* vers une valeur beaucoup plus basse. Les valeurs de *n*, *m* et *p* sont disponibles dans les registres `%eax`, `%edi`, `%esi` respectivement, où *p* = Pisano(*m*). Vous pouvez utiliser l'adresse mémoire réservée `adr` pour vos calculs.

2.2 Partie 2 : Propriétés de la période de Pisano

La période de Pisano présente de nombreux résultats mathématiques : Comme vous l'avez probablement remarqué, la séquence périodique commence toujours par 0 puis 1. De plus, la séquence ne peut avoir qu'un nombre fini de zéro : 1 zéro, 2 zéros ou 4 zéros. On note également que pour $p, q \in \mathbb{N}$ premiers entre eux ($\text{pgcd}(p, q) = 1$), $\text{Pisano}(p \times q)$ n'est autre que le plus petit commun multiple de $\text{Pisano}(p)$ et $\text{Pisano}(q)$.

Q1/ Complétez le programme assembleur `fibo_mod_zero.s` qui calcule le nombre de zéros dans la séquence périodique de Pisano, telle que définie dans la section `.data` du même programme. Vérifiez votre programme en testant votre code sur la deuxième séquence présente dans la mémoire. Pour ce faire, modifiez les commentaires dans le code.

Notez bien que la valeur de la période de Pisano ($\text{Pisano}(m)$) vous est donnée au début du programme par le registre `%edi`.

Q2/ Examinez attentivement la période de Pisano suivante pour $m = 4$:

$$0, 1, 1, 2, 3, 1, 0, 1, 1, 2, 3, 1, 0, 1, 1, 2, 3, 1, 0, 1$$

En vous basant sur le principe de Fibonacci et le calcul du modulo pour la période de Pisano, Pourriez-vous déduire la relation entre les différentes valeurs? Expliquez et justifiez votre réponse en s'appuyant sur un autre exemple (*m* différent).

Q3/ Soit $m \in \mathbb{N}$; $m \geq 2$. Complétez le programme assembleur `fibo_mod_m_v3.s` qui doit calculer les termes de la séquence de Pisano (c'est-à-dire le modulo de la suite de Fibonacci) et les écrire en mémoire à partir de l'adresse réservée `adr_v3`, en utilisant la relation déduite de la question précédente.

Notez bien que votre programme doit être différent de ceux de la Q2 dans la Partie 1 et utiliser l'observation faite à la Q2 de la partie 2. Les valeurs de *n* et *m* sont disponibles dans les registres `%edi` et `%esi` respectivement.

Q4/ Soit $p, q \in \mathbb{N}$ premiers entre eux. Complétez les programme assembleurs `fibo_mod_p_q_mul.s` et `fibo_mod_p_q.s` qui doivent calculer Pisano$(p \times q)$.

Notez bien que le programme `fibo_mod_p_q_mul.s` doit impérativement utiliser l'instruction « `mul` », tandis que le programme `fibo_mod_p_q_mul.s` ne doit pas l'utiliser. Les deux programmes devront mettre le résultat final dans le registre `%eax`. Les valeurs p et q sont disponibles dans les registres `%edi` et `%esi` respectivement.

2.3 Bonus :

On définit la *suite de Lucas* par la relation de récurrence suivante :

$$L_0 = 2$$
$$L_1 = 1$$
$$L_n = L_{n-2} + L_{n-1} \,;\ \ n \in N, n \geq 2$$

Elle est définie par la même relation de récurrence que la suite de Fibonacci, mais avec les valeurs 1 et 3 comme point de départ.

La période de Pisano des nombres de Lucas a la propriété suivante :

Si $m = L_{2k} \geq 3, k \geq 1$, alors Pisano$(m)$=$8k$

Si $m = L_{2k+1} \geq 3, k \geq 1$, alors Pisano$(m)$=$4k$+$2$

C'est-à-dire que si m est un nombre de Lucas $L_n \geq 3$ pour *n* pair, alors $\text{Pisano}_\text{L}(m) = 4$n ; et si *m* est un nombre de Lucas $L_n \geq 3$ pour *n* impair, alors $\text{Pisano}_\text{L}(m)$ =2n.

Complétez le programme assembleur `lucas.s` qui calcule $\text{Pisano}_\text{L}(m)$ pour *m*, un nombre de Lucas $L_n \geq 3$.

Notez bien que le programme devra mettre le résultat final dans le registre `%eax`. Les valeurs de *n* et *m* sont disponibles dans les registres `%edi` et `%esi` respectivement. Vous pouvez utiliser l'adresse mémoire réservée `adr_lucas` pour vos calculs.