

TP1 : Représentation des données en mémoire et opérations arithmétiques

Introduction

Ce TP utilise un simulateur du jeu d'instructions MIPS pour exécuter des programmes écrits en langage assembleur. Ils sont accessibles sur la page web du cours, à l'adresse <http://www.lri.fr/~de/CLM1314.html> . Ces programmes permettront d'observer

- L'implantation des données en mémoire
- L'exécution d'instructions arithmétiques MIPS et l'observation des cas de débordement

Mode d'emploi de QtSpim

On utilise le simulateur QtSpim, fonctionnant sous Windows. Il est téléchargeable à l'adresse <http://spimsimulator.sourceforge.net> pour utilisation sur PC personnel.

Visualisation des informations

- Visualisation des registres entiers et flottants en binaire, décimal ou hexadécimal. (Privilégier l'utilisation de l'hexadécimal)
- Segment Data : permet de visualiser (binaire, décimal ou hexa) les données du système d'exploitation (kernel) et de l'utilisateur (user)
- Segment Text : permet de visualiser le code système (kernel) et le code utilisateur (user).

Chargement des fichiers

- Onglet « file » : « Load file » ou « Reinitialize and Load file »

Exécution de la simulation

- Onglet « simulator » : exécution complète (Run/Continue) ou pas à pas (single step)

Représentation des données en mémoire

Exécuter successivement les programmes Memoire1M.s, Memoire2M.s et Memoire2M.s

1. Memoire1M.s : voir l'implantation mémoire.
2. Memoire2M.s : observer le « padding » utilisé pour respecter l'alignement mémoire.
3. Memoire3M.s : en mode « disassembly », exécuter pas à pas et voir le résultat d'exécution des différentes instructions Load. Quel ordre utilise le simulateur MIPS QtSpim (big ou little endian) ?

Instructions arithmétiques entières

1. Exécuter le programme addM.s pas à pas et voir le résultat d'exécution des instructions arithmétiques.
2. Remplacer successivement les instructions addu par des instructions add et voir le résultat obtenu.

Instructions arithmétiques flottantes

1. Exécuter le programme FopM.s pas à pas et voir le résultat d'exécution des instructions arithmétiques flottantes (vue décimale et vue hexadécimale)
2. Faire plusieurs exécutions de FopM.s en modifiant le contenu des variables n1 et n2.
3. Modifier le programme FopM.s pour calculer la surface d'une sphère selon la formule $S = 4 * \pi * R^2$. (utiliser votre numéro d'étudiant comme valeur de R).

Compte rendu de TP

Envoyer à de@lri.fr un compte rendu individuel ou par binôme pour le calcul de la surface d'une sphère avec

- une copie du programme de surface_sphere.s
- une copie d'écran montrant le contenu des registres flottants (décimal) et User data segment (hexadécimal).

Calculer la surface avec Rayon sphère = Numéro étudiant (pour compte rendu individuel) et avec deux exécutions pour compte rendu par binôme (les deux numéros étudiants).

Date limite : 13 Février 2014.

Annexe 1 : Big endian et little endian

Exemple avec 0x90AB12CD

	Big ENDIAN	Little ENDIAN
Adresse	Valeur	Valeur
1000	90	CD
1001	AB	12
1002	12	AB
1003	CD	90

Annexe 2 : Codage ASCII

I:SA	MSB	0		1		2		3		4		5		6		7	
		nn	nm	no	ni	nn	nm	no	ni	nn	nm	no	ni	nn	nm	no	ni
0	0000	NUL	DL	SP	0	@	P	~	p								
1	0001	SOH	DC1		1	A	Q	a	q								
2	0010	STX	DC2	"	2	O	R	b	r								
3	0011	ETX	DC3	#	3	C	S	c	s								
4	0100	EOT	DC4	\$	4	D	T	d	t								
5	0101	ENQ	NAK	%	5	E	U	e	u								
6	0110	ACK	SYN	&	6	F	V	f	v								
7	0111	BEL	ETB	'	7	G	W	g	w								
8	1000	BS	CAN	:	8	H	X	h	x								
9	1001	HT	EM	;	9	I	Y	i	y								
A	1010	L	SUB	x	:	J	Z	j	z								
B	1011	VT	ESC	+	:	K	[k	}								
C	1100	FF	FS	.	<	L	!	l									
D	1101	CR	GS	-	=	M	:	m	{								
E	1110	SO	RS	.	>	N	^	n	~								
F	1111	SI	US	!	?	O	_	o	DEL								

Exemples : le chiffre '1' est codé 0x31, le caractère a est codé 0x61, le caractère A est codé 0x41, le point d'interrogation est codé 0x3F, etc.