Universidade Fernando Pessoa

Arquitectura de Computadores
Ficha de Exercícios nº4

Objectivos:

- Funções, representação de instruções.

1. Considere o seguinte programa em linguagem C:

```
main() {
        int i, s=0;
        for(i=0;i<10;i++){
                s+=i;
        }
        printf("The final value is %d\n",s);
}
```

   a. Escreva o seu equivalente em assembly do MIPS (sem usar pseudo instruções).
   b. De seguida proceda à sua assemblagem manual. Assuma que o endereço de memória onde ficará guardada a primeira instrução é 0x00000000[1].
   c. Compare o código hexadecimal que obteve na alínea anterior com o gerado pelo simulador MIPS para o mesmo procedimento.

2. Considere o seguinte código máquina do MIPS:

Endereço	Instrução
0x00400024	0x00008020
0x00400028	0x20080001
0x0040002C	0x20090064
0x00400030	0x11090003
0x00400034	0x02088020
0x00400038	0x21080001
0x0040003C	0x0810000c
0x00400040	

   a. Descodifique cada uma das instruções apresentadas.
   b. O que faz este programa?

3. Considere o seguinte código em C:

```
main() {
        int i,j=0;
        for (i=1; i<1000; i=i*2) j+=i;
}
```

   a. Escreva o correspondente código em assembly do MIPS
   b. Escreva o correspondente código máquina em hexadecimal

Bibliografia:

[1] Patterson & Hennessy – *Computer Organization and Design: The hardware/software interface 4rd Ed* – MKP 2009.

[1] Deve tratar a chamada ao procedimento `printf` como uma chamada a uma função usando a instrução `jal` e não invocar as chamadas ao sistema do SPIM como habitualmente faz.

University Fernando Pessoa

Computer Architecture
Exercise sheet nº4

Goals:

- Understand functions in MIPS assembly
- Understand the representation in machine language (binary or hexadecimal) of MIPS assembly instructions

1. Consider the following C program:

```
main() {
	int i, s=0;
	for(i=0;i<10;i++){
		s+=i;
	}
	printf("The final value is%d\n",s);
}
```

   a. Convert it to MIPS assembly (without using pseudo instructions).
   b. Proceed with the manual assembly of the MIPS assembly program. Assume that the memory address where the first instruction resides is 0x00000000[2].
   c. Compare the manual machine language code obtained with the one produced by the MIPS Simulator for the same source code.

2. Consider the following MIPS machine language program:

Address	Instruction
0x00400024	0x00008020
0x00400028	0x20080001
0x0040002C	0x20090064
0x00400030	0x11090003
0x00400034	0x02088020
0x00400038	0x21080001
0x0040003C	0x0810000c
0x00400040	

   a. Decode to MIPS assembly each of the previous machine language instructions.
   b. Explain what is the purpose of this program?

3. Consider the following C program:

```
main() {
	int i,j=0;
	for (i=1; i<1000; i=i*2) j+=i;
}
```

   a. Convert it to MIPS assembly.
   b. Proceed with the manual assembly of the MIPS assembly program. Present the machine code in hexadecimal.

Bibliography:

[1] Patterson & Hennessy – *Computer Organization and Design: The hardware/software interface 4rd Ed* – MKP 2009.

[2] You should consider the call to `printf` procedure as a normal function call using the `jal` assembly instruction instead of invoking the simulator system calls as you usually do.