

## Lab 5 – Linguagem Assembly para MIPS

***Neste trabalho de laboratório pretende-se iniciar a programação em assembly para MIPS.***

Os exercícios que se seguem devem ser executados usando o MARS, que é um simulador do MIPS disponível em <http://courses.missouristate.edu/KenVollmar/MARS/>. Leia atentamente a documentação e explore a sua utilização. Na aula deve tentar explicar todos os detalhes dos exercícios realizados.

### 1. Um pequeno programa para MIPS

Escreva no MARS um programa **lab5\_1.asm** que, dados os valores em \$16 (\$s0) e \$17 (\$s1), coloque nos registos temporários \$8 (\$t0) a \$15 (\$t7) os seguintes valores:

```
$t0 = $s0  
$t1 = $s1  
$t2 = $t0 + $t1  
$t3 = $t1 + $t2  
...  
$t7 = $t5 + $t6
```

Ou seja, para cada registo de \$8 (\$t0) a \$15 (\$t7), armazena a soma dos dois registos temporários anteriores.

Para testar o programa altere os valores iniciais de \$16 (\$s0) e \$17 (\$s1).

### 2. Análise de um programa para MIPS

Carregue no MARS o ficheiro **mars1.asm** e execute-o. Repare que o programa envia para o "stdio" strings, caracteres e inteiros de funções que fazem uso de chamadas ao sistema. Note que o MARS além de simular o processador, simula também um micro-sistema operativo cujas funcionalidades são acedidas colocando no registo \$v0 o número do serviço que se pretende e em \$a0 o argumento a passar a esse serviço.

Este programa contém 2 erros: o primeiro é a falta de um espaço antes da palavra MIPS e o segundo deverá ser capaz de descobrir. (Nota: verifique a correspondência entre os valores da tabela ASCII e os caracteres impressos no ecrã.)

- a) Corrija esses 2 erros.
- b) Modifique o programa para que apresente a frase numa só linha sem números nem separadores.

### 3. Criação do primeiro programa para MIPS

Crie e teste no MARS um programa que peça ao utilizador 1 inteiro, usando a syscall correspondente (ver ajuda do MARS) e apresente no ecrã todos os números naturais superiores a esse número e inferiores ao seu dobro.

Exemplo: Número dado é 5. Deve apresentar no ecrã os números 6, 7, 8, 9. A ordem como são apresentados é irrelevante.

### 4. Programa MIPS para duplicar array de valores

Use o esqueleto do ficheiro **doubleArray.asm** para implementar um programa que duplique todos os valores de um `array` de valores, com `n` elementos.

O esqueleto do programa já carrega o endereço do vetor `array` para o registo `$s0` (1a `$s0, array`) e o endereço da variável `n` para o registo `$s1` (1a `$s1, n`). De seguida o programa carrega da memória o valor da variável `n` para o registo `$s2`.

Implemente o ciclo para ler, duplicar e armazenar os valores do array conforme as instruções em comentário no ficheiro.