

Arquitetura de Computadores I
2ª série de problemas
9.11.2015

1. Converta o seguinte código assembly do MIPS em código máquina

```
lw      $t7, 20($t0)
sub     $t1, $t7, $s0
addi    $t0, $t0, 4
```

2. Traduza os blocos de código C seguintes para assembly do MIPS utilizando as instruções **beq**, **bne** e **slt**.
Assuma que g e h estão, respetivamente, nos registos \$s0 e \$s1

```
a.  if (g > h)
    g = g+h;
    else
    g = g-h;
```

```
b.  if (g >= h)
    g = g+1;
    else
    g = g-1;
```

```
c.  if (g <= h)
    g = 0;
    else
    h = 0;
```

3. A instrução **li** (Load Immediate) é uma instrução virtual. Qual a tradução em instruções nativas de
li \$t0, 0x1002002C

4. No MIPS as únicas instruções nativas de **branch** são branch on equal e branch on not equal. Indique como são traduzidas para instruções nativas as seguintes instruções:

- a) bge \$t0, \$t1, Label
- b) blt \$t0, \$t1, Label

5. unsigned int sum(unsigned int n)

```
{
  if (n == 0) return 0;
  else return n + sum(n-1);
}
```

- a) Converta a função C para assembly do MIPS. Siga as convenções de invocação de funções do MIPS.
- b) Suponha que o código da função está carregado em memória a partir do endereço 0x00400028 e que é invocada sum(4). Represente os estados do stack ao longo da execução da função.

6. Nas aulas foi traduzida para assembly a seguinte versão recursiva do cálculo de factorial(n):

```
int fact (int n)
{
  if (n < 1) return (1);
  else return n * fact(n - 1);
}
```

Faça a tradução para uma versão iterativa de factorial(n). Indique o conteúdo do stack durante a execução da versão recursiva e da versão iterativa quando é invocado fact(3).