

Exame de Recurso de Arquitectura de Sistemas e Computadores I

Licenciatura em Engenharia Informática

5 de Julho de 2013

1. Considere a função `xpto` seguinte a executar num processador MIPS32 - little endian:

```
.data
S:      .asciiz "o rato roeu a rolha da garrafa"

        .text

xpto:   move $s0, $a0
        li $s1, 0
A:      lb $s2, 0($s0)
        bgt $s2, 'z', B
        nop
        blt $s2, 'a', B
        nop
        move $a0, $s2
        jal upper          # converte caracter para maiuscula
        nop
        sb $v0, 0($s0)
        addi $s1, $s1, 1
B:      addi $s0, $s0, 1
        bne $s2, $zero, A
        nop
        move $v0, $s1
        jr $ra
        nop
```

ASCII: 'a' = 97 = 0x61; 'A' = 65 = 0x41; 'z' = 122 = 0x7a; 'Z' = 90 = 0x5a.

- (a) Identifique todas as pseudo-instruções e substitua-as por instruções reais. Determine o espaço de memória ocupado pelo código máquina desta função.
 - (b) Escreva o código máquina correspondente às instruções marcadas pelas etiquetas A e B, e descubra quais as instruções cujo código máquina é 0x03e00008 e 0x34110000 (localize-as no programa).
 - (c) Explique qual o objectivo da função `xpto`. Indique quais são os argumentos e valor de retorno. Explique resumidamente como funciona a função (por exemplo, se é recursiva, se modifica arrays memória, etc. Não explique o que fazem as instruções individualmente, mas sim como um todo)
 - (d) A função `xpto` segue correctamente as convenções?
(Em caso negativo, aponte todos os erros e faça as correcções necessárias)
 - (e) Supondo que a string S é carregada em memória no endereço 0x10010004 e que a última instrução está no endereço 0x00401008, indique o valor (em hexadecimal) dos bytes nos seguintes endereços de memória: i) 0x10010007; ii) 0x1001000d; iii) 0x0040100b; iv) 0x00401004.
 - (f) Considerando uma frequência de relógio de 1.2 GHz, e que cada instrução executa num ciclo de relógio, determine o tempo de execução da função `xpto` quando é passada a string S como argumento. Represente o resultado em μs e ns .
 - (g) Optimize o código e calcule o *speedup* obtido.
2. Represente os números -13 , 1.1 , $-\text{Inf}$ e 0.0 em vírgula flutuante IEEE754, precisão simples. (escreva o resultado em binário e hexadecimal)