

Teste Tipo (Sistemas de Computação – LTSI)Componente teórica

Nota: Nas questões de respostas múltiplas, a escolha de mais que uma resposta ou de uma resposta absurda, implica o desconto de 50% da cotação da pergunta.

1. Numa conversa entre dois profissionais foram utilizados os seguintes termos: notação hexadecimal; endereço de memória; registos; código de operação; chamada da função de leitura do disco. Na sua opinião, a que nível de abstracção estariam os dois profissionais a conversar?
 - ☐ Nível do software de aplicação.
 - ☐ Nível do *Assembler*.
 - ☐ Nível do Sistema Operativo.
 - ☐ Nível da linguagem máquina.
2. Um dos componentes fundamentais de um computador actual é a memória dinâmica (ou DRAM). Esse componente é utilizado para:
 - ☐ Construir o subsistema de memória de massa.
 - ☐ Construir a memória de cache.
 - ☐ Construir a memória central.
 - ☐ Construir a memória de programas.
3. No MIPS, como na grande maioria dos processadores, os operandos são armazenados em registos internos para serem sujeitos às operações aritméticas e lógicas. Nesta perspectiva seria bom dispor de um número muito elevado de registos, atendendo à elevada complexidade dos problemas a resolver (tipicamente). No entanto esse número é reduzido, porque:
 - ☐ Durante a “guerra” CISC/RISC demonstrou-se que um número elevado de registos dificultava a tarefa de programação.
 - ☐ Quanto maior for o número de registos maior será a dimensão da CPU e, por conseguinte, maiores serão os tempos de atraso, que por sua vez condicionam o desempenho da própria CPU.
 - ☐ Quanto menor for o número de registos menor será a complexidade da CPU, pelo que o seu projecto e desenvolvimento será facilitado.
 - ☐ Existe um compromisso entre o número de registos e a sua dimensão (“largura”). No MIPS todos os registos são de 32 bits, pelo que têm que ser em número reduzido.
4. Ao construir um programa define-se, habitualmente, um espaço de *stack* – ao nível do *assembler* esse espaço é explicitamente reservado no programa, mas ao nível das HLL o compilador faz essa reserva automaticamente. Descreva, de uma forma sucinta, uma limitação resultante da geração automática da *stack*.

5. A construção de uma hierarquia de memórias assenta em questões tecnológicas e em princípios resultantes da prática de programação estruturada. As questões de programação estão ligadas ao conceito de "localidade", o qual pode revelar-se de duas formas, "localidade espacial" e "localidade temporal". A que se refere a primeira destas propriedades?
 - ☐ Localização dos programas em memória.
 - ☐ Execução por ciclos, que leva a que a mesma instrução seja executada diversas vezes.
 - ☐ Estrutura dos programas, nomeadamente à forma como os compiladores organizam o código.
 - ☐ Execução sequencial das instruções, que permite prever, com elevada probabilidade, qual a instrução seguinte a ser executada.
6. Suponha que está a trabalhar com um computador que implementa memória virtual. Durante a execução de um programa recebe a informação (do Sistema Operativo) que o programa vai abortar porque violou o espaço de endereçamento. Qual das seguintes situações poderá estar na base dessa mensagem?
 - ☐ O programa gerou um endereço virtual para uma página, excedendo o número de entradas da tabela de páginas do programa.
 - ☐ O programa gerou um endereço virtual que não se encontra na cache.
 - ☐ O programa gerou um endereço virtual associado a uma página que não se encontra na memória central.
 - ☐ O programa gerou um endereço cujo deslocamento está fora da gama de endereços da página.

Teste Tipo (Sistemas de Computação – LTSI)Componente prática

1. Considere o seguinte programa em *assembly* do MIPS, no qual se declara um vector de 10 elementos e que aparentemente executa uma determina operação, de uma forma iteractiva (*loop*). Nesse programa foram omitidas partes de algumas instruções e comentários.
- a) Complete os espaços em branco, preenchendo-os com o que falta das respectivas instruções e com os comentários que entender que são necessários, para deixar clara a sua intenção.

```
.data
save: .word 0,0,0,0,0,10,20,30,40,50
eol: .asciiz "\n"

.text
main:
     $s3,0          # 
    la $s6,save
Loop:
    slt $t3,$s3,      # 
    beq $t3,$zero,Exit
    add $t1,$s3,$s3
    add $t1,$t1,$t1
    add $t1,$t1,$s6
     $a0,0($t1)      # 
    li ,1          # 
    syscall
    li ,4          # 
    la $a0,eol
    syscall
     $s3,,1      #incrementa o contador
     Loop          # 
Exit:
    li $v0, 10
    syscall
```

- b) De uma forma sintética, descreva a função deste programa. e em particular do conjunto das três instruções que aparecem salientadas.

2. Escreva, em *assembly* do MIPS um procedimento denominado “**maior**” que, dado um número inteiro até 127 passado como parâmetro, devolva “0” (**zero**) se esse número for menor do que “50” (**cinquenta**) e devolva “100” (**cem**) caso contrário. Note que, na prática, isto corresponde a um arredondamento às centenas. Indique claramente as opções assumidas na passagem de parâmetros.
3. Escreva um programa que receba de um utilizador vinte números inteiros e, usando o procedimento “**maior**”, guarde num vector os valores correspondentes arredondados às centenas. Deverá, sempre que necessário, imprimir mensagens que orientem o utilizador no interface com o programa.