

## SISTEMAS COMPUTACIONAIS

LICENCIATURA EM ENGENHARIA INFORMÁTICA - 2º ANO

### Cotações

- 1) \_ 1.25
- 2) \_ 1.25
- 3) \_ 1.25
- 4) \_ 1.25
- 5) \_ 1.25
- 6) \_ 1.25
- 7) \_ 1.25
- 8) \_ 1.25
- 9a) \_ 1.00
- 9b) \_ 1.00
- 10) \_ 1.00
- 11a) \_ 1.00
- 11b) \_ 1.00
- 12a) \_ 1.00
- 12b) \_ 1.00
- 13a) \_ 1.00
- 13b) \_ 1.00
- 13c) \_ 1.00

PROVA DE AVALIAÇÃO ESCRITA SEM CONSULTA – VERSÃO A TERÇA, 8 DE FEVEREIRO DE 2022  
SEM CONSULTA DURAÇÃO: 50 MINUTOS (COMPLEMENTAR) / 100 MINUTOS (EXAME) COM TOLERÂNCIA INCLuíDA

NOME: \_\_\_\_\_ NÚMERO: \_\_\_\_\_

**Atenção:** Justifique todas as suas respostas na medida do necessário, sem escrever testamentos. Pense, organize as ideias e escreva. Pode apresentar exemplos e/ou grafismos para melhor elucidar as suas respostas. Considere  $1\text{TiB}=2^{40}$   $1\text{GiB}=2^{30}$  Bytes,  $1\text{MiB}=2^{20}$  Bytes e  $1\text{kiB}=2^{10}$  Bytes. Nas questões de código, pode dividir a zona de resposta a meio para ter o dobro de linhas.

### TEÓRICA / PARTE 1

1) Descreva qual funcionalidade do registo **R14 (LR)** da arquitetura ARMv7, referida nas aulas.

---

---

---

---

---

---

---

---

---

---

2) Apresente duas limitações na passagem de parâmetros por registos.

---

---

---

---

---

---

---

---

---

---

3) Na instrução **B** (branch da arquitetura ARMv7) existem várias condições. Indique pelo menos duas variantes com **objetivos distintos**, referindo o conteúdo do registo de *flags* (bit ou bits).

---

---

---

---

---

---

---

---

---

---

- 4) Apresente a sequência de passos que devem ser realizados e associados a uma interrupção.

---

---

---

---

---

---

---

---

---

---

***TEÓRICA / PARTE 2***

- 5) Indique o que significa “*cache miss*” memória cache? Qual o funcionamento da cache nessa situação?

---

---

---

---

---

---

---

---

---

---

- 6) Descreva a estrutura de um carater (sequência de bits) na comunicação série.

---

---

---

---

---

---

---

---

---

---

- 7) Num sistema operativo real é possível criar programas em *assembly* e utilizar os recursos desse sistema, como por exemplo escrever uma mensagem na consola / terminal?

[illegible]

- 8) Que registos podem/devem ser utilizados para invocar uma chamada ao System Dispatcher / OS API, a partir do *assembly*?

[illegible]

*PRÁTICA / ASSEMBLY*

- 9) Responda às questões abaixo, tendo em conta o código assembly ARMv7 apresentado:

Código:

```

;R1 with STRING address
START  mov      r0, #0
LOOP   ldrb     r2, [r1], #1
        cmp     r2, #0
        beq     ENDLOOP
        add     r0, r0, #1
        bal     LOOP
ENDLOOP

```

- a) De que tipo é estrutura de “decisão e controlo” utilizada no código?

[illegible]

b) Qual o significado do valor final do registos **R0**?

[illegible]

10) Tendo um vetor de valores de 32 bit, elabore um programa em assembly ARMv7 capaz de contar quantos elementos desse vetor são pares e quantos são zero (zero não é par nem ímpar).

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is a vertical margin line on the left side, creating a narrow left margin. The paper appears to be a standard notebook or ledger page.

11) Tendo por base o assembly ARMv7 e o emulador VisUAL responda às seguintes questões:

- a) Desenvolva uma sub-rotina que determine o número total de bits com o valor “1” de número de 8 bit, passado com parâmetro. Os registos de entrada e saída devem ser definidos por si.


- b) Desenvolva uma sub-rotina que determine o número total de bits com o valor “1” de um vetor de valores 8 bits utilizando (invocando) a sub-rotina anteriormente definida.


## PRÁTICA / ASSEMBLY / PLATAFORMA GNU

12) Considere o trecho de código seguinte (suponha que as equivalências / valores estão corretos):

Código:

```
error:      mov r7, #OS_WRITE
            mov r0, #OS_STDERR
            ldr r1, =errmsg
            mov r2, #31
            swi 0
```

a) Indique qual o objetivo / o que faz o código.

---

---

---

---

---

---

b) O que representam os valores / registos **R0, R1, R2 e R7**?


13) Novamente com base no assembly ARMv7 e um sistema com as ferramentas GNU:

a) Desenvolva uma sub-rotina que determine o somatório dos valores de um vetor de números de 8 bit, sendo o vetor de número terminado pelo valor 0. A interface com a sub-rotina deve ser a seguinte: **R0** – endereço do primeiro elemento do vetor; retorno **R0** (somatório).

[illegible]

- b) Indique que elementos / diretivas adicionais devem ser colocados no código, de forma a ser possível invocar a sub-rotina a partir de um programa em C.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

- c) Apresente um trecho de código / dados em C que utiliza a sub-rotina definida anteriormente.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**Área extra para respostas (identificar qual a pergunta)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

This image shows a single page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.