

# Organização e Funcionamento de um Computador

## Teste 2

Nº	Nome
----	------

**Nota:** Apresente sempre o raciocínio ou os cálculos que efectuar; o não cumprimento desta regra equivale à não resolução do exercício.

1. Considere o seguinte estado dum PC actual (com CPU IA32), com destaque para o banco de registos do CPU e para o conteúdo de algumas células de memória.

Banco de registos do CPU:

<i>eax</i>	00	00	<i>ah</i>	00	<i>al</i>	08
<i>ecx</i>	01	23	<i>ch</i>	45	<i>cl</i>	67
<i>edx</i>	a0	b0	<i>dh</i>	c0	<i>dl</i>	d0
<i>ebx</i>	ff	cf	<i>bh</i>	07	<i>bl</i>	7f
<i>esi</i>	00	00	<i>d5</i>		<i>e8</i>	
<i>edi</i>	08	81	<i>22</i>		<i>70</i>	
<i>esp</i>	08	c1	<i>20</i>		<i>42</i>	
<i>ebp</i>	08	c1	<i>20</i>		<i>40</i>	
<i>eip</i>	08	01	<i>42</i>		<i>36</i>	

Estado da memória a partir dos seguintes endereços (célula a célula):

0x8c12040:  
80 7f f8 f3 ff ff 4e 20 c1 08 00 02 00 00 08 0c

Estes dados foram obtidos com `objdump -d`:

0x8014230:  
55 89 e5 8b 45 0c 03 45 08 89 ec 5d c3 8d 76 00

**Nota1:** O IA32 é *little endian*.

**Nota2:** Todos os conteúdos indicados estão representados em hexadecimal.

- a) (A/R) A instrução que o PC está a acabar de processar na altura em que se analisou o seu estado (e apresentado no início do enunciado) corresponde à instrução `movl 0xc(%ebp), %eax` (a origem é uma localização em memória cujo endereço é a soma de `0xc` com o conteúdo do registo `%ebp`). O valor de `%eip` antes de iniciar a execução desta instrução era inferior em 3 unidades. Indique, cronologicamente, toda a informação que circula nos barramentos de endereços e de dados (em hexadecimal), desde o fim da instrução anterior, até ao fim do processamento desta instrução. Considere que (i) o decodificador de instruções não tem nenhum *byte* por decodificar, e que (ii) o barramento de dados é de 16 bits (o que não é verdade num PC actual!).

Nº	Nome
----	------

- b) <sup>(A)</sup> Mostre, em binário, o conteúdo do registo `%eax`, após a execução da instrução referida na alínea anterior.
- c) <sup>(A/R)</sup> A instrução referida nas alíneas anteriores (e executada pelo CPU) foi obtida a partir do código de uma atribuição de um valor a uma variável, numa função em C. Identifique e descreva resumidamente os mecanismos de conversão utilizados para, a partir do código C, se ter chegado ao código que foi directamente executado no CPU nas alíneas anteriores.
- d) <sup>(R/B)</sup> A próxima instrução que este PC vai executar é uma adição, conforme é possível verificar por um mapa de *opcodes*. Indique, justificando, (i) a localização dos operandos fonte e destino e (ii) o resultado da operação (em hexadecimal).