



2005/2006

**Curso:** LMCC

**Exame 2ª Chamada** 10/Jul/06

**Disciplina:** Arquitectura de Computadores

**Duração:** 2h30m

**Nota:** Apresente sempre o raciocínio ou os cálculos que efectuar; o não cumprimento desta regra equivale à não resolução do exercício. Use o verso das folhas do enunciado do exame como papel de rascunho.

1. Considere o ficheiro com o enunciado deste exame. **Indique, justificando:**

a) <sup>(A)</sup> Se o ficheiro pode ser do tipo `.txt`, com os caracteres codificados em ASCII de 7 bits.

b) <sup>(A)</sup> Se o formato GIF é mais adequado que o JPEG para representar os logótipos da UM.

2. <sup>(A/R)</sup> Os principais blocos que constituem um computador incluem um CPU e a memória principal. **Identifique e caracterize** (dimensão, direcção e funcionalidade) os principais sinais ou canais de comunicação que deverão existir entre o CPU e a memória principal.

3. Considere um PC de 1ª geração, com registos de 16-bits, *little endian*, com capacidade de endereçamento de apenas 16-bits (versão simplificada), e em que os inteiros são representados normalmente por apenas 16 bits em complemento para 2.

Considere ainda que as células de memória desse PC contêm os seguintes valores (representados em vários sistemas de numeração), a partir do endereço `0x8ca0`:

$240_8$ ,  $0x8c$ ,  $0x42$ ,  $1000_2$ ,  $96$ ,  $0_5$ ,  $31_4$ ,  $12$ ,  $17_{11}$ ,  $14_6$ ,  $80$ ,  $100_2$ ,  $86_9$ ,  $0_3$ ,  $140_8$ ,  $0$

Nº

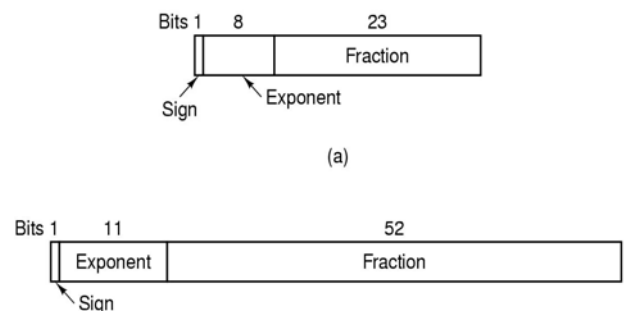
Nome

- a) (A) Supondo que as duas primeiras células contêm um apontador para um valor inteiro, **represente em decimal** esse valor inteiro (basta apresentar a expressão aritmética na base decimal).
- b) (A) Supondo que as duas primeiras células contêm um apontador para um valor real (precisão simples), **represente em decimal** esse valor real (apresente só a expressão aritmética na base decimal).

**Notas de apoio** (norma IEEE 754)

Normalized	±	0 < Exp < Max	Any bit pattern
Denormalized	±	0	Any nonzero bit pattern
Zero	±	0	0
Infinity	±	1 1 1 ... 1	0
Not a number	±	1 1 1 ... 1	Any nonzero bit pattern

↙ Sign bit



Valor decimal de um fp em binário:

precisão simples, normalizado:  $V = (-1)^S * (1.F) * 2^{E-127}$   
 precisão simples, desnormalizado:  $V = (-1)^S * (0.F) * 2^{-126}$

- c) (B) Pretende-se representar com precisão o nº de Avogadro (=6.022\*10<sup>23</sup>) usando uma notação de vírgula flutuante com 20 bits. **Indique, justificando**, se é possível e com quantos bits para o expoente/mantissa.

Nº	Nome
----	------

4. Considere o seguinte programa em C (listagem incompleta) que foi editado num ficheiro e posteriormente compilado para executar num PC (com CPU IA32):

```
void ex2ch (int *xp, int *yp)
{
    int x,y;
    corpo_da_função
}

void main ()
{
    int x = 5;
    int y = 2;
    ex2ch (&x, &y);
    printf ("x=%d , y=%d /n", x, y);
}
```

Considere ainda a seguinte listagem obtida durante o desenvolvimento desse programa:

```
00401060 <_ex2ch>:
401060: 55                push   %ebp
401061: 89 e5             mov    %esp,%ebp
401063: 53                push   %ebx
401064: 8b 4d 08          mov    0x8(%ebp),%ecx
401067: 8b 5d 0c          mov    0xc(%ebp),%ebx
40106a: 8b 11             mov    (%ecx),%edx
40106c: 8b 03             mov    (%ebx),%eax
40106e: 8d 04 80          lea   (%eax,%eax,4),%eax
401071: c1 e0 05          shl   $0x5,%eax
401074: 83 c0 40          add   $0x40,%eax
401077: 89 01             mov    %eax,(%ecx)
401079: 8d 54 d2 07       lea   0x7(%edx,%edx,8),%edx
40107d: 89 13             mov    %edx,(%ebx)
40107f: 5b                pop    %ebx
401080: 5d                pop    %ebp
401081: c3                ret
```

Sabe-se que, no main, no fim da execução da instrução que invoca a função `ex2ch` (e correspondente à instrução de `call` do IA32), os registos do CPU continham os seguintes valores:

<code>%eax</code> : 0x5	<code>%ebx</code> : 0x2c00	<code>%ebp</code> : 0x408c98
<code>%ecx</code> : 0x2c	<code>%esi</code> : 0x8008	<code>%esp</code> : 0x408c84
<code>%edx</code> : 0xffffffff80	<code>%edi</code> : 0x12	<code>%eip</code> : 0x401060

- a) **(A) Indique, justificando**, como se obteve a listagem com o código da função em *assembly* (i.e., que utilitários do Unix é que se teriam usado e como).

Nº

Nome

- b) (A/R) Quando o processador tiver concluído a execução da 1ª instrução do corpo da função (procure-a, pensando!), **indique, justificando**, a lista de registos e de todas as células de memória que foram modificadas (i.e., onde um novo valor foi escrito), desde que se analisou o conteúdo dos registos.
- c) O CPU acaba de executar a instrução em 0x401063 (`push %ebx`) e prepara-se para a seguinte (`mov ...`). **Indique, cronologicamente** e em binário ou hexadecimal:
- i) (A/R) O conjunto de comandos emitidos pela Unidade de Controlo do CPU (apenas estes, e a quem são destinados) para que essa instrução (`mov ...`) possa ser devidamente executada (incluindo a sua busca da memória, e considerando que ela vem toda de uma só vez da memória).  
**Sugestão:** imagine que está a fazer o papel de UC numa peça de teatro que mostre o funcionamento de um CPU na execução de instruções.
- ii) (A/R) Todo o tráfego que irá circular no barramento de dados desde essa altura até ao fim da execução da instrução seguinte.  
**Nota:** considere os barramentos de endereços e de dados de 32 bits.
- d) (A/R) A instrução em *assembly* em 0x40106e é semelhante a um `mov (%eax,%eax,4),%eax`.  
**Explique** as diferenças entre estas 2 instruções **e calcule** os valores que serão colocados em `%eax` para cada uma das instruções (se necessitar de conteúdos de memória, refira-os como sendo `M[endereço_calculado]`).

Nº

Nome

- e) (R/B) Considere que o corpo da função em `ex2ch` continha a estrutura de controlo `while ((x>0) && (x>y)) { }`, sendo `x` e `y` as variáveis alocadas respectivamente aos registos `%edx` e `%eax`. **Mostre** a parte do código que o comando `gcc -O2` aplicado a este programa iria gerar para implementar esta estrutura.
- f) (R) **Represente** a estrutura da *stack frame* desta função (`ex2ch`) quando estiver a executar a multiplicação, com indicação da **dimensão\_e\_descrição** de cada um dos valores nela presente.
- g) (B) **Modifique** a instrução em *assembly* do IA32 `add %eax, 8(%edx)` para que ela possa ser executada por um conjunto mínimo de instruções RISC (use/adapte a sintaxe do *assembler* do IA-32 da Gnu).
5. Considere o desenvolvimento de um programa em C com preocupações de desempenho, o qual necessita de efectuar operações complexas com elementos de matrizes.
- a) (B) **Apresente** uma lista de 3 recomendações fundamentadas que apresentaria a quem estivesse a desenvolver esse programa, de modo a contribuir para uma melhoria do seu desempenho.
- b) (B) Na medição de tempos de execução de programas para avaliação de desempenho, usam-se normalmente contadores de intervalos ou contadores de ciclos de *clock*. **Caracterize** sumariamente cada um destes, **mostrando** os prós e contras de cada (use o verso da folha para responder).

Nº

Nome