Curso: LCC **1ª Teste** 30/Abr/08
Disciplina: Sistemas de Computação **Duração (máx):** 1h30m

Avisos

1. Este Teste vale 10 valores e é constituído por 2 partes:
   - **Parte 1**: – **obrigatória**
     7 questões de representação de Informação, estrutura de um computador e execução de instruções.
     Falhando mais de 2 questões não satisfaz os resultados de aprendizagem requeridos para aprovação na UC; falhando 1 a 2 haverá lugar a reapreciação (analisado caso a caso).
   - **Parte 2** Classificativa – **opcional**,
     Para definição da classificação final (até 3 valores neste Teste, na escala 10 a 20), apenas para quem satisfizer os critérios mínimos.
2. 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.

Parte 1

1. **(A)** A representação de uma imagem num computador pode ser feita através do valor da intensidade de cor de cada um dos seus *pixels* em cada um dos canais de RGB, usando 8 bits para cada cor. **Indique**, **justificando**, se esta é a forma de representar imagens fotográficas proposta pelo ***J**oint **P**hotographic **E**xperts **G**roup*.

2. **(A)** Considere que o registo `%bx` (IA-32) contém o conjunto de bits correspondente ao valor `0x84`. Sabendo que esse valor é um `short int` (codificado em complemento para 2), **represente** em decimal o valor que está lá guardado.

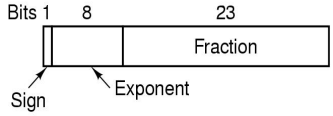
Nº	**Nome**

3. **(A)** Considere a representação de valores em vírgula flutuante, precisão simples, de acordo com a norma IEEE 754. **Calcule** no eixo positivo, a menor potência de 10 que é possível representar com a notação normalizada (será do tipo 10^{-x}; calcule o valor de x).

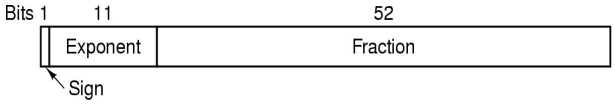
Notas de apoio (norma IEEE 754)

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



(a)



Valor decimal de um fp em binário:
precisão simples, normalizado: $V = (-1)^S * (1.F) * 2^{E-127}$
precisão simples, desnormalizado: $V = (-1)^S * (0.F) * 2^{-126}$

4. **(A)** A partir de um programa fonte escrito numa HLL, é possível obter-se um ficheiro executável armazenado num disco. **Identifique** e **caracterize** todos os passos necessários para transformar o ficheiro fonte em ficheiro executável.

Nº	**Nome**

5. Considere a execução da instrução (em *assembly*) `pushl %eax`, (cujo início ocorreu quando o processador terminou a execução da instrução anterior), num processador IA-32 (*little endian*), e assuma que o conteúdo de alguns registos é o seguinte (em hexadecimal):

```
%eax  ff fe 80 04
%eip  08 c2 04 48
%esp  08 c2 8f 00
```

a) **(A)** **Apresente**, por ordem cronológica e em hexadecimal, toda a informação que circula apenas no barramento de endereços.

b) **(A)** **Mostre** o conteúdo da célula de memória apontado por `(%esp)+3` após a execução desta instrução. Apresente os cálculos que efectuar.

c) **(A)** Considere que a instrução que se segue a esta é `leal 8(%esp),%eax` . **Mostre** o conteúdo do registo `%ah` após execução desta instrução. Apresente os cálculos/raciocínio que efectuar.

Nº	**Nome**

Parte 2

6. **(R/B)** Considere a utilização da constante física Fermi Coupling, com o valor **3.670 336[31] e+48 kg^{-2}**. Este valor é usado em simulações de um processo físico, em que foi inicialmente armazenado num programa escrito em C como valor do tipo `double`. Na utilização de uma dada biblioteca de cálculo no IA-32, ele teve de ser convertido para o formato `float`. **Apresente** a sua representação em hexadecimal neste formato.

7. **(R)** Suponha que o registo `%edx` contém o valor `x` , que `%ebx` contém o valor `list e` que `%esi` contém o valor `index`. Utilizando o menor nº de instruções em *assembly* do IA-32, construa o pedaço de código que coloca num elemento do vector `array` o resultado da seguinte operação aritmética:

```
array[x] = ( 32 + list + índex*4 ) + 65
```

Nota: `array` é um vector de apontadores (uma variável local da função), que se encontra na *stack*, a começar no endereço a 40 posições acima do `%ebp`, i.e., no endereço calculado por <`%ebp`>-40

8. **(R)** Considere o seguinte conjunto de instruções em *assembly* do IA-32. Se a condição de teste da instrução de salto condicional `jg` for verdadeira, **indique** a instrução que irá executar após o salto?

```
1 80483c8: 7e 11                jle   80483db
2 80483ca: 8d b6 00 00 00 00    lea   0x0(%esi),%esi
3 80483d0: 89 d0                mov   %edx,%eax
4 80483d2: c1 f8 01             sar   $0x1,%eax
5 80483d5: 29 c2                sub   %eax,%edx
6 80483d7: 85 d2                test  %edx,%edx
7 80483d9: 7f f5                jg    xxxxxxx
8 80483db: 89 d0                mov   %edx,%eax
```

Nº	**Nome**