



Universidade do Minho
Escola de Engenharia

MIETI :: Métodos de Programação II
2015/16

Práticas Laboratoriais

Módulo 4

António Esteves
esteves@di.uminho.pt
Abril 2016

Exercício 1 - Leitura dum ficheiro em modo binário

- ◇ Escrever um programa em C para ler um ficheiro previamente escrito em modo binário.
- ◇ Assuma que o conteúdo do ficheiro é aquele que resultou do exercício 1 da aula teórica 6. Ou seja, o conteúdo do ficheiro deve ser lido para um *array*, de estruturas do tipo **Aluno**, com 5 posições. A estrutura **Aluno** possui 3 campos:
 - **nome** (*string* com 100 carateres)
 - **numero** (*int*)
 - **nota** (*float*)
- ◇ Assuma que os valores foram escritos começando no elemento zero do *array* até ao último, e para cada posição do *array* os campos da estrutura foram escritos pela ordem usada na declaração da estrutura.
- ◇ Depois de ler o ficheiro, mostre os valores lidos no ecrã.

Exercício 2 - Operações ao nível do bit

- Suponha que possui um variável `instruction` de 16 bits (`unsigned short int`), composta pelos seguintes conjuntos de bits (campos):

15..11	10 .. 6	5 .. 3	2 .. 0	← bit
OPCODE	FUNCTION	REG_DEST	REG_ORIG	

- Assumindo que a variável `instruction` contém o valor hexadecimal `0x76A5`, utilize operações ao nível do bit e escreva código para obter o valor dos 4 campos dessa variável.
- Considere ainda que quer definir o valor de `instruction` à custa dos seguintes campos:
`OPCODE = 0x1b`
`FUNCTION = 0x14`
`REG_DEST = 0x5`
`REG_ORIG = 0x3`
- Utilizando operações ao nível do bit, escreva o código que define o valor em causa.

Exercício 3 - Situações de overflow com inteiros

- ◇ Considere que as variáveis **i**, **i1**, **i2** e **i3** são do tipo **int**, enquanto as variáveis **u**, **u1**, **u2** e **u3** são do tipo **unsigned int**.
- ◇ Inicialize estas variáveis com os seguintes valores:
i1 = u1 = 1700000000
i2 = u2 = 500000000
i3 = u3 = 400000000
- ◇ Calcule as seguintes expressões e indique, para cada uma, se ocorre *overflow* (com/sem sinal):
i = i1+i2
i = i1+i3
u = u1+u2
i = i3-i2
u = u3-u2
i = 6*i3
u = 6*u3

Exercício 4 - Erros no sinal ou de truncatura

- ◇ Considere que as variáveis **i**, **i1**, **i2** e **i3** são do tipo **int**, as variáveis **u** e **u1** são do tipo **unsigned int** e as variáveis **si2** e **si3** são do tipo **short int**.

- ◇ Inicialize estas variáveis com os seguintes valores:

i1 = -100

u1 = 2500000000

i2 = 33000

i3 = 32000

- ◇ Calcule as seguintes expressões e indique, para cada uma, se ocorre um erro (no sinal ou de truncatura):

i = u1 <=> i = (int)u1

u = i1 <=> u = (unsigned int)i1

si2 = i2 <=> si2 = (short int)i2

si3 = i3 <=> si3 = (short int)i3

Exercício 5 - Conversão entre inteiros e reais

- ◇ Considere que as variáveis **i** e **i1** são do tipo **int** e as variáveis **f** e **f1** são do tipo **float**.

- ◇ Inicialize estas variáveis com os seguintes valores:

f1 = 3.75

i1 = 479856701

- ◇ Efetue as seguintes conversões entre **int** ↔ **float**:

i = (int)f1

f = (float)i1

- ◇ Explique os resultados obtidos nas duas conversões.

Exercício 6 - Qual a memória física e de swap do nosso sistema

- ◇ Utilizando comando(s) do Linux, indique quanta memória física possui o meu sistema (*cache*, ram e disco)?
- ◇ Utilizando comando(s) do Linux, indique quanta memória de *swap* possui o meu sistema (total, usada, livre, usada como *cache*)?

Exercício 7 - Tipo de ordenação dos bytes em memória

- ◇ Escrever um inteiro (`int`) em memória (por exemplo, `3F2E1D0C` em hexadecimal).
- ◇ Ler os 4 bytes, um a um, da memória onde se escreveu o inteiro anterior e tirar conclusões sobre o tipo de ordenação usada para armazenar os bytes dum inteiro em memória.