

Curso Profissional: Programador/a de Informática
PSD – 10.º ano: UFCD 0810 - Programação em C/C++ - avançada

Ficha de Trabalho 4

Ano letivo 21/22

A função `sizeof()`

Podemos descobrir quantos bytes certa variável ocupa através da função `sizeof()`.

Esta função recebe uma variável como argumento (exemplo 2), ou as palavras reservadas (exemplo 1) que representam as variáveis: `char`, `int`, `float`, etc.

Sempre que usamos a função `sizeof()`, ela devolve variáveis do tipo: *size_t* (números inteiros positivos). É possível também utilizar este operador aplicando-o a uma expressão.

Exemplo 1

```
#include <stdio.h>
main()
{
    printf("Char: %d byte\n", sizeof(char));
    printf("Int: %d bytes\n", sizeof(int));
    printf("Float: %d bytes\n", sizeof(float));
    printf("Double: %d bytes\n", sizeof(double));
}
```

Output:

Char: 1 byte
Int: 4 bytes
Float: 4 bytes
Double: 8 bytes

Exemplo 2

```
#include <stdio.h>
main()
{
    char carater;
    int inteiro;
    float Real;
    double Duplo;

    printf("Caráter: %d bytes\n", sizeof(carater));
    printf("Inteiro: %d bytes\n", sizeof(inteiro));
    printf("Float: %d bytes\n", sizeof(Real));
    printf("Double: %d bytes\n", sizeof(Duplo));
}
```

Campos de bits (Bit field):

Na definição de uma estrutura podem definir-se campos formados por um ou alguns bits, permitindo assim poupar espaço.

Se, por exemplo, ao armazenar o valor “SIM” ou “NÃO”, “MASCULINO” ou “FEMININO” irá despende-se 1 byte por cada um dos caracteres guardados em memória.

Mesmo que o utilizador coloque os valores 1 ou 0, irá gastar pelo menos 1 byte, pois o menor tipo que existe em C é o tipo `char`, que ocupa 1 byte.

Ao utilizar-se campos de bits bastará apenas um bit para armazenar esta informação, pois tem apenas 2 estados podendo ser utilizados por 1 único bit.

Suponhamos a definição de uma estrutura onde se colocam os seguintes campos:

- ⚡ Nome
- ⚡ Idade
- ⚡ Sexo (0: Mulher, 1:Homem)
- ⚡ Estado Civil (0:solteiro, 1:Casado, 2:Viúvo, 3:Divorciado)

```
#include <stdio.h>
#include <string.h>
struct pessoa
{
    char nome[30];
    int idade;
    unsigned sexo:1; /* 1 bit */
    unsigned est_civil: 2; /* 2 bits */
};
main()
{
    struct pessoa p;
    strcpy(p.nome, "Barnabéu oTerrível");
    p.idade=32;
    p.sexo=1; /*masculino */
    p.est_civil= 3; /*divorciado */
    /*Mostrar os valores */
    printf("Nome: %s\n", p.nome);
    printf("Idade: %d\n", p.idade);
    printf("Sexo: %s\n", p.sexo==0? "Feminino":"Masculino");
    printf("Estado civil: ");
    switch (p.est_civil)
    {
        case 0: puts("Solteiro");break;
        case 1: puts("Casado");break;
        case 2: puts("Viúvo");break;
        case 3: puts("Divorciado");
    }
}
```

Nota: Os campos de bits podem ser particularmente úteis se forem utilizados para representar os dados de inquéritos em que o conjunto de respostas é normalmente limitado e constituído por conjuntos pré-definidos de valores.

OUTPUT

Nome: Barnabéu o Terrível
Idade: 32
Sexo: Masculino
Estado civil: Divorciado