



**FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA**

Departamento de Engenharia Electrotécnica

PROGRAMAÇÃO DE MICROPROCESSADORES

2021 / 2022

Mestrado Integrado em Engenharia Electrotécnica
e Computadores

1º ano

1º semestre

Trabalho nº 7
Apontadores

1 Introdução

No capítulo 8 do livro “Linguagem C” de Luís Damas, recomendado para a disciplina de Programação de Microprocessadores, é apresentado os apontadores. Esta aula visa consolidar esta matéria através de um conjunto de exercícios. Faça todos os exercícios pedidos em ficheiros separados e **GUARDE O CÓDIGO desenvolvido na memória USB**. Durante a aula o docente pode pedir-lhe para mostrar o código desenvolvido.

2 Objetivo do trabalho

O objetivo do trabalho é mostrar as três áreas principais onde os apontadores são usados:

- 1) na passagem de parâmetros nas funções;
- 2) em percorrer listas/vetores;
- 3) em trabalhar (executar operações) em endereços de memória onde estão variáveis de um certo tipo, sem saber exatamente o nome das variáveis.

3 Revisitar o “jogo do mais ou menos”

Um dos usos mais comuns dos apontadores é na passagem de parâmetros para as funções. Recupere o código do “jogo do mais ou menos” mas agora o `main` vai ser modificado (e com ele o programa quase todo).

O `main` deve ser qualquer coisa como o que está mostrado em baixo (de propósito não são mostrados os argumentos das funções):

```
main ()
{
    do {
        if (automático())
            sorteia_valores ();
        else
            pergunta_valores ();
        jogo ();
    } while (joga_outra_vez());
}
```

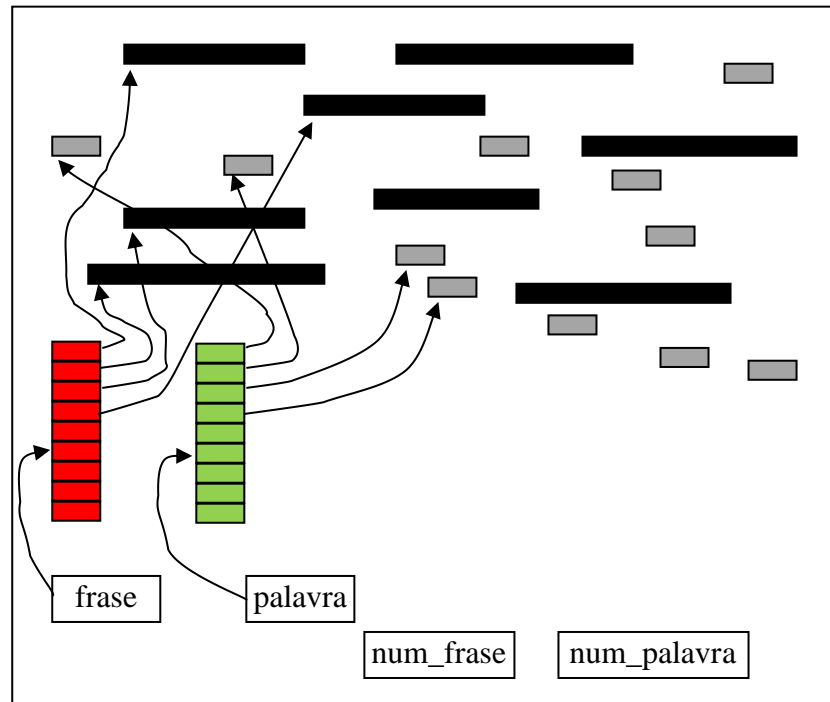
4 Apontadores e vetores

Deve ter reparado no livro que os apontadores e os vetores são parentes muito próximos. Se não reparou volte a estudar o livro agora. Existe mesmo uma secção chamada de “Apontadores e Vetores”.

Este exercício serve para aprender a usar estes dois conceitos. A figura da página seguinte pretende mostrar a memória de um computador de um ponto vista imaginário.

Espalhados pela memória existem *strings* com pedaços de texto (grandes e escuros), e existem *strings* só com uma palavra (pequenos e cinzentos). Para tornar tudo mais flexível, não se sabe quantos existem de cada espécie, nem onde eles realmente estão em memória.

Para saber onde estão as *strings* na memória existem dois vetores de apontadores para `char` – representados a vermelho (para os textos) e a verde (para as palavras). Cada posição deles contém o endereço da respectiva *string*. Em ambos os casos estão representadas as 4 primeiras setas, para se perceber.



Existem ainda duas variáveis, chamadas “frase” e “palavra” que são apontadores para apontadores para `char`. O propósito é percorrer os vetores vermelho e verde, para se poder testar todas as frases e todas as palavras.

Finalmente existem ainda mais duas variáveis do tipo inteiro que servem para guardar as escolhas do utilizador. São chamadas de “num_frase” e num_palavra”.

O objetivo do programa é perguntar ao utilizador que indique o número de uma frase e de uma palavra e depois verifique se essa palavra está nessa frase.

Para isso tem de programar umas funções que sejam muito genéricas e que devolvam um resultado verdadeiro ou falso consoante uma certa palavra se encontra numa certa frase.

O objetivo de se usar apontadores é que o programa serviria para fazer isto exaustivamente para uma quantidade muito grande de frases e uma quantidade muito grande de palavras.

Para ajudar os alunos na introdução de frases e de palavras, use o programa da página seguinte que já contém 7 frases e 6 palavras, para além das variáveis descritas acima. Pode fazer “copy” e “paste” para o seu editor e depois mudar algumas coisas.

```

/*
 *                               Ficheiro frases.c
 */
#include <stdio.h>
#include <string.h>

#define bool int
#define TRUE 1
#define FALSE 0

#define MAX_WORD_LENGTH 21
#define MAX_TEXT_LENGTH 81
#define MAX_WORD 20

    código aqui

main() {
    char frase1 [MAX_TEXT_LENGTH] = "A string that contains a dog and a cat";
    char frase2 [MAX_TEXT_LENGTH] = "A string that contains camel and mouse";
    char frase3 [MAX_TEXT_LENGTH] = "The dog was eating with the horse";
    char frase4 [MAX_TEXT_LENGTH] = "The camel drunk all the water";
    char frase5 [MAX_TEXT_LENGTH] = "The mouse entered into the pipe";
    char frase6 [MAX_TEXT_LENGTH] = "Uma string que pode conter dog ou camel";
    char frase7 [MAX_TEXT_LENGTH] = "The horse is bigger than the dog";
    char palavra1 [MAX_WORD_LENGTH] = "dog";
    char palavra2 [MAX_WORD_LENGTH] = "horse";
    char palavra3 [MAX_WORD_LENGTH] = "cat";
    char palavra4 [MAX_WORD_LENGTH] = "mouse";
    char palavra5 [MAX_WORD_LENGTH] = "camel";
    char palavra6 [MAX_WORD_LENGTH] = "pig";

    char *frase_ptr [MAX_WORD];                // para apontar a frases
    char *pal_ptr [MAX_WORD];                  // para apontar a palavras
    char ** frase, ** palavra;
    int num_frase, num_palavra;                // para conter a escolha do util.

    frase_ptr [0] = frase1;
    frase_ptr [1] = frase2;
    frase_ptr [2] = frase3;
    frase_ptr [3] = frase4;
    frase_ptr [4] = frase5;
    frase_ptr [5] = frase6;
    frase_ptr [6] = frase7;
    pal_ptr [0] = palavra1;
    pal_ptr [1] = palavra2;
    pal_ptr [2] = palavra3;
    pal_ptr [3] = palavra4;
    pal_ptr [4] = palavra5;
    pal_ptr [5] = palavra6;

    while (1) {
        escolha (&num_frase, &num_palavra);

        código aqui

        ver_se_contem (frase_atual, palavra_atual);
    }
}

```

A função *ver_se_contem* deve escrever a frase e a palavra escolhidas e depois deve escrever sim ou não consoante a palavra se encontra na frase, ou não.

O programa deve perguntar um número de uma frase e um número de uma palavra ao utilizador. Depois deve ver se a frase contém a palavra.

Repare também que programar com apontadores é bastante versátil mas **muito perigoso**. Tomando como exemplo o nosso programa, se o utilizador escolher um número de sequência inválido, por exemplo, o número 26 pode acontecer que o valor lá colocado aponte para a uma zona de memória proibida e com isso o programa rebenta. **Tente fazer isso para ver como o computador reage a um erro desses**.

5 Tratamento sistemático de variáveis de certos tipos de dados

O último propósito de uso de apontadores que vamos estudar é programar certo tipo de comportamento a certo tipo de dados sem estar relacionado com uma variável específica. Um exemplo usando o que acontece numa oficina de viaturas é muito elucidativo. Imagine então uma oficina e uma mudança de uma peça num modelo de carro sem interessar o carro específico (a matrícula). Todos os Audi A3, por exemplo, e não este ou aquele carro em particular. Assim, uma função recebe um apontador para inteiro, por exemplo, e executa uma certa operação com ele (ou a ele).

Estes exemplos fazem mais sentido com o uso de estruturas, mas como as estruturas vão ser estudadas a seguir (Trabalho 8), os programas aqui explicados acabam por ser um pouco tontos. De um modo geral, o verdadeiro uso de apontadores começa a fazer-se sentir em problemas com um grau de complexidade um pouco maior do que o que se exige em Programação de Microprocessadores. O propósito dos problemas deste trabalho é simplesmente fazer uma primeira familiarização dos alunos com os apontadores.

Imagine então uma porção de variáveis inteiras que representam a conta bancária de pessoas. De propósito não se vão usar vetores para não cairmos nos casos das secções anteriores. Imagine agora uma função que adiciona uma certa quantia (eventualmente negativa) a uma conta.

Assim, pode-se imaginar que se pretende somar uma certa quantia a todos os homens. Ou retirar uma certa quantia a todas as mulheres.

Use o programa abaixo.

Escreva uma função que faça mesmo isso – que adicione um valor inteiro a uma variável que recebe de entrada. A função não sabe o que significa de fato a variável.

Esta função vai ser chamada a partir o `main`, e o `main` é que vai saber o que está a fazer. Por exemplo, atendendo ao programa em baixo e às variáveis que ele tem, se chamar sequencialmente com os nomes `joao`, `pedro`, `jose` e `jorge` cada vez com um certo valor, está-se, da fato, a fazer um programa que vai adicionar um certo valor às contas dos homens.

O exemplo é muito simples. Com o uso de estruturas e de vetores conseguem-se fazer programas mais compactos que executam este tipo de comportamentos...

```

/*
 *                               Ficheiro frases.c
 */
#include <stdio.h>
#include <string.h>

#define bool int
#define TRUE 1
#define FALSE 0

#define MAX_WORD_LENGTH 21
#define MAX_TEXT_LENGTH 81
#define MAX_WORD 20

    código aqui

main() {
    int joao, ana, mariana, catarina, pedro, filipa, margarida, jose, jorge,
    susana, cristina;

    joao = 1000;
    ana = 2500;
    mariana = 3000;
    catarina = 1200;
    pedro = 2500;
    filipa = 340;
    margarida = 3400;
    jose = 120;
    jorge = 340;
    susana = 120;
    cristina = 1290;

    código aqui

    }
}

```