

Curso Profissional: Programador/a de Informática
PSD – 10.º ano: UFCD 0810 - Programação em C/C++ - avançada
Ficha de Trabalho 9

Ano letivo 21/22

Estruturas de dados com ponteiros

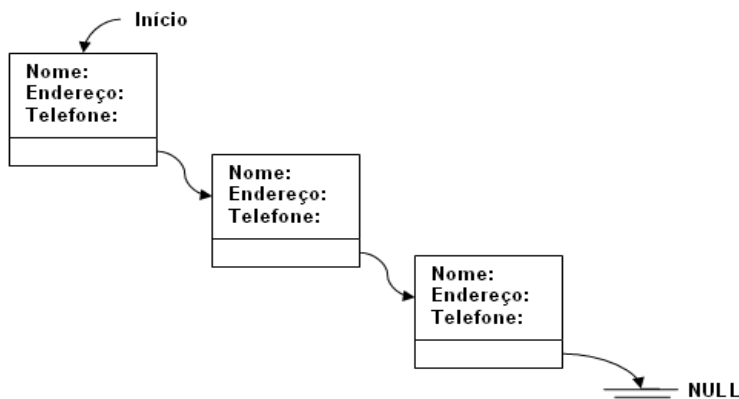
Exemplo1 (para apenas um registo e sem utilização de memória dinâmica):

typedef struct

```
{  
    char nome[100];  
    int idade;  
} pessoa;  
main()  
{  
    pessoa joao;  
    pessoa *p = &joao;  
    strcpy(joao.nome, "joao da silva");  
    joao.idade = 20;  
    printf("%s, %d\n", p->nome, p->idade); // o mesmo que (*p).idade  
    p->idade = 18;  
    printf("%s, %d\n", joao.nome, joao.idade);  
}
```

A principal função dos ponteiros é permitir a criação e a manipulação das variáveis dinâmicas, as quais irão compor as **estruturas de dados dinâmicas**.

Se tivermos que fazer um programa que terá que ler uma certa quantidade indeterminada de registos do teclado. Não sabemos se serão 10, 100 ou até 1000 registos. A princípio, poderemos superdimensionar um array, desde que o computador tenha memória suficiente, mas mesmo assim, corre-se o risco de, no futuro, termos que redimensionar o array. Para um caso como este, podemos utilizar o conceito de variáveis dinâmicas. Para tal, devemos declarar um ponteiro para uma variável, cuja estrutura seja constituída pelos campos onde armazenaremos o(s) valor(es) e por um que aponte para a próxima variável dinâmica.



Exemplo 2 (com utilização de memória dinâmica):

Este programa lê registos com a estrutura abaixo, até que se digite 'fim' quando é pedido o nome da pessoa. Este programa tem a capacidade de ler um número ilimitado de registos sem a preocupação de se definir um array e sua respetiva dimensão.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
struct pessoa
{
    int idade;
    char nome[21], genero;
    float altura;
    struct pessoa *prox;
} d,*aux,*p;

void Linha ()
{
    int i;
    for (i=1;i<80;i++)
        printf("-");
}

main()
{
    int i;
    p=NULL;
    do
    {
        Linha();
        aux=(struct pessoa*) malloc(sizeof(struct pessoa));
        printf ("Nome da pessoa -----> "); scanf("%s", &(*aux).nome);fflush(stdin);
        if (strcmp((*aux).nome,"fim")!=0)
        {
            printf("Género ----->"); scanf(" %c",&(*aux).genero);fflush(stdin);
            printf("Idade ----->"); scanf("%d",&(*aux).idade);fflush(stdin);
            printf("Altura ----->"); scanf("%f", &(*aux).altura);fflush(stdin);
            (*aux).prox=p;
            p=aux;
        }
    }while (strcmp((*aux).nome,"fim")!=0);
```

```

system ("cls");
Linha();
aux=p;
while (aux!=NULL)
{
    printf("%s%5c%5d%5.2f\n",(*aux).nome,(*aux).genero,(*aux).idade,(*aux).altura);
    aux=(*aux).prox;
}
printf("\n");
}

```

Nota: ([*p](#)).idade = 31; // mesmo efeito que d.idade = 31

Exercício1 : Escreve as instruções necessárias para criar um programa que:

- i. Declare um ponteiro para uma estrutura chamada Lista, com os campos :codigo, nome e proximo, em que código é do tipo inteiro, nome do tipo string e proximo um ponteiro para a estrutura;
- ii. Insira no registo 10 elementos;
- iii. Liste os elementos inseridos.