



UNIPAC

Universidade Presidente Antônio Carlos

Bacharelado em Ciência da Computação

Introdução a Programação

Material de Apoio

Parte XII – Alocação Dinâmica

Prof. Nairon Neri Silva

naironsilva@unipac.br

1º sem / 2020

Material cedido pela professora Livia

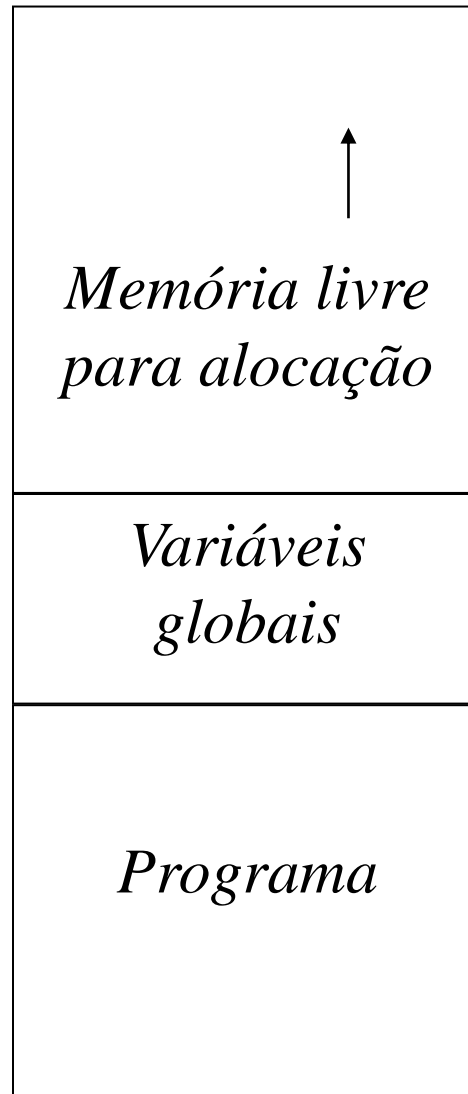
Alocação dinâmica

- Criação e destruição de informações na memória em tempo de execução. É de responsabilidade do programador desalocar toda memória que vier a alocar durante a execução do seu aplicativo.
- O armazenamento para a informação é alocado da memória livre conforme suas necessidades.

Variáveis estáticas X Variáveis dinâmicas

- Variáveis estáticas: armazenamento é fixo durante todo o tempo de validade da variável. O programador sabe, de antemão, a quantidade de armazenamento necessária para todas as situações.
- Variáveis dinâmicas (Ponteiros): tipos de variáveis que apontam para uma área da memória. A principal função dos ponteiros é permitir a criação e a manipulação das variáveis dinâmicas, as quais irão compor as estruturas de dados dinâmicas.

Alocação dinâmica



Ponteiros

- Declaração de ponteiros:

*tipo_do_ponteiro *nome_da_variável;*

- É o asterisco (*) que faz o compilador saber que aquela variável não vai guardar um valor mas sim um endereço para aquele tipo especificado.
- Exemplos:

```
int *pt;  
char *temp, *pt2;
```

Ponteiros

- Quando declaramos ponteiros, e ainda não foram inicializados, significa que eles apontam para um lugar indefinido. Este lugar pode estar, por exemplo, na porção da memória reservada ao sistema operacional do computador.
- Logo, *o ponteiro deve ser inicializado (apontado para algum lugar conhecido) antes de ser usado!*
- Para saber o endereço de uma variável basta usar o operador &.

Ponteiros

- Exemplo:

```
int count=10;  
int *pt;  
pt=&count;  
*pt=12
```

- As duas variáveis apontam para o mesmo endereço de memória. Alterando o valor de *pt* estará alterando o valor de *count*.

Ponteiros

- Exemplo:

```
#include <stdio.h>
main () {
    int num, valor;
    int *p;
    num=55;
    p=&num;      /* Pega o endereco de num */
    valor=*p;    /* Valor e igualado a num de maneira indireta */
    printf ("\n\n%d\n", valor);
    printf ("Endereco para onde o ponteiro aponta: %p\n", p);
    printf ("Valor da variavel apontada: %d\n", *p);
}
```