

Aula 04

MC 102 - Algoritmos e Programação de Computadores

Entrada e Saída de Dados. Escrevendo um programa

Endereço de variáveis

- Variáveis armazenam dados na memória. Logo, elas possuem um endereço (posição) de memória associado.
- Esse endereço é o **Local** onde o conteúdo da variável é armazenado no sistema (uma pessoa é “armazenada” no endereço de sua residência)
- Só são conhecidos quando o programa está sendo executado
 - Quando escrevemos o programa, não sabemos qual será o endereço de uma variável: ela ainda não existe na memória!

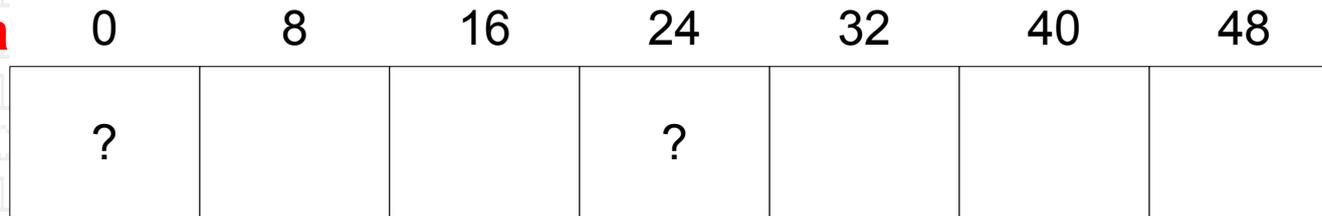
Endereço de variáveis

- O sistema operacional é quem define onde a variável será armazenada
 - Ele é responsável pelo gerenciamento da memória
- O endereço de uma variável pode mudar entre diferentes execuções de um mesmo programa, usando a mesma máquina

Armazenamento de Dados

```
int a, b;
```

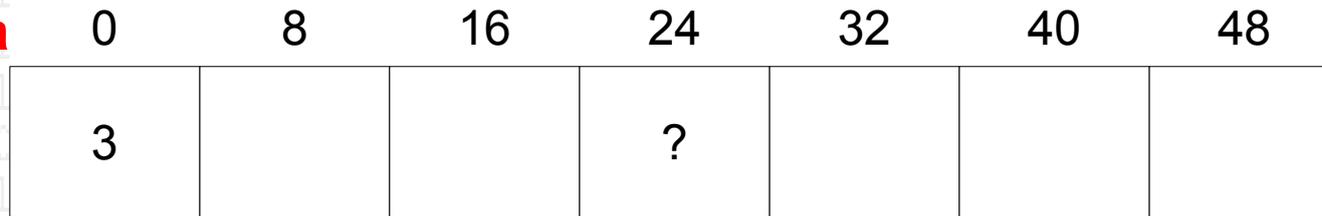
**Endereço na
memória**



Armazenamento de Dados

```
int a, b;  
a = 3;
```

**Endereço na
memória**



a

b

Armazenamento de Dados

```
int a, b;
```

```
a = 3;
```

```
b = a + 5;
```

**Endereço na
memória**

0	8	16	24	32	40	48
3			8			

a

b

Armazenamento de Dados

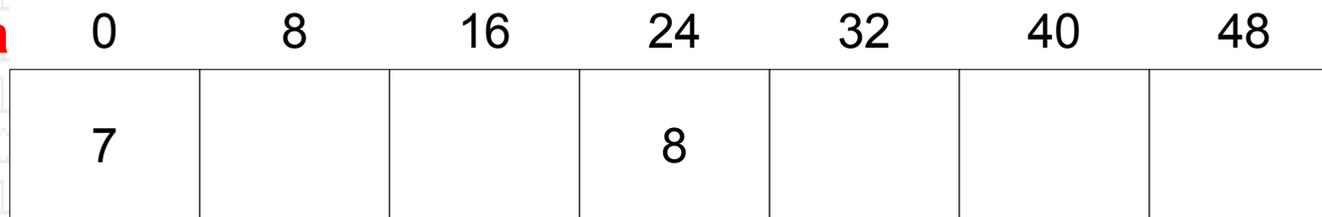
```
int a, b;
```

```
a = 3;
```

```
b = a + 5;
```

```
a = 7;
```

**Endereço na
memória**



Entrada de Dados

Como fazer para entrar
com dados no programa?

O teclado é o dispositivo de entrada
padrão, mas como perguntar ao usuário
e ler o conteúdo digitado?

Função scanf () - Definição

Função para obter (ler) números e caracteres digitados no teclado

`scanf(<formato>, <end. variáveis>);`

Parâmetros da função scanf

- Uma cadeia de caracteres (string) indicando o tipo das variáveis que serão lidas e o formato dessa leitura
- Uma lista do endereço das variáveis

Formatos de Leitura de Variável

Código

Função

%d

Lê um inteiro decimal

%u

Lê um inteiro decimal sem sinal

%l

Lê um inteiro longo

%f

Lê um ponto flutuante

%lf

Lê um double

%c

Lê um caracter

%s

Lê uma seqüência de caracteres

&: Operador *address-of* em C

- Este operador fornece o endereço da variável durante a execução do programa

```
int variavel;
```

```
scanf("%d", &variavel);
```

- A função `scanf` exige o endereço para alterar o valor da variável

– **Esquecer de colocar & é comum**

- Pode gerar erros de execução
- A variável não terá o valor alterado

Exercício

Endereço na memória

0	8	16	24	32	40	48
34			12.67			

a

b

a ?

b ?

&a ?

&b ?

Exercício

Endereço na memória

0	8	16	24	32	40	48
34			12.67			

a

b

a 34

b 12.67

&a 0

&b 24

Função scanf - Definição

Leitura de dados pelo teclado:

Leitura de 1 variável:

```
scanf(<tipo_de_variável>, <end_variável>);
```

```
Ex: scanf("%d", &a); scanf("%f", &raio);
```

Leitura múltipla:

```
scanf(<tipos_de_variáveis>, <end_variáveis>);
```

```
Ex: scanf("%d %f", &a, &raio);
```

Obs: Sempre usar <espaço> entre os formatos

Função scanf – Leitura simples

Lê um inteiro:

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a;

    scanf("%d", &a);
}
```

Função scanf – Leitura Múltipla

Lê um inteiro e um ponto flutuante:

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a;
    float b;

    scanf("%d %f", &a, &b);
}
```

Saída de Dados

Como fazer para imprimir os dados do programa?

O monitor é o dispositivo de saída padrão, mas como enviar o conteúdo para lá?

Função printf() - Definição

- Função para imprimir números e caracteres no monitor
- Utiliza os mesmos formatos de scanf, com algumas opções extras

`printf(<formato>, <variáveis>, <constantes>);`

Caracteres especiais para impressão

<code>\t</code>	tabulação horizontal	<code><tab></code>
<code>\n</code>	nova linha	<code><enter></code>
<code>\”</code>	áspas	“
<code>\'</code>	apóstrofo	'
<code>\?</code>	interrogação	?
<code>\\</code>	barra invertida	\
<code>%%</code>	porcentagem	%

Formatos Inteiros

```
printf(<codigo>, 9);
```

<u>Código</u>	<u>Resultado</u>
---------------	------------------

<code>%d</code>	9
-----------------	---

forma geral: `%<número>d`

<code>%4d</code>	<code><espaço><espaço><espaço>9</code>
------------------	--

forma geral: `%0<número>d`

<code>%04d</code>	0009
-------------------	------

forma geral: `%<numero1>.0<numero2>`

<code>%6.04d</code>	<code><espaço><espaço>0009</code>
---------------------	---

<code>%+d</code>	+9
------------------	----

Obs: O sinal - sempre aparece para números negativos

Formatos Ponto Flutuante

```
printf (<codigo>, 42.1);
```

<u>Código</u>	<u>Resultado</u>
---------------	------------------

%f	42.100000
----	-----------

%e	4.210000e+01
----	--------------

forma geral: %<tamanho>f

%6f	<espaço><espaço>42.1
-----	----------------------

forma geral: %<tamanho>.<decimais>f

%6.2f	<espaço>42.10
-------	---------------

forma geral: %0<tamanho>.<decimais>f

%06.2f	042.10
--------	--------

forma geral: %.<decimais>f

%.2f	42.10
------	-------

Formatos Caracter e String

- **Formato caracter %c**

```
printf("%c", 'A'); imprime A
```

```
printf("%c", 65); imprime A
```

- **Formato string %s**

```
printf("%s", "A"); imprime A
```

```
printf("%s", "Primeira\nSegunda");
```

```
imprime: Primeira  
Segunda
```

Leitura e Impressão simples

Lê um inteiro e imprime valor:

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a;

    printf("Entre com o valor de a: ");
    scanf("%d", &a);
    printf("\nObrigado! Valor fornecido: %d\n", a);
}
```

Leitura e Impressão Múltipla

Lê um inteiro e um ponto flutuante e imprime seus valores:

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a;
    float b;

    printf("Entre com um inteiro e um valor real: ");
    scanf("%d %f", &a, &b);
    printf("\nObrigado! a: %d e b:%f\n", a, b);
}
```

Exercício

Retornando ao exercício de cálculo da média, escreva um programa em C que obtém 3 variáveis inteiras (a, b, c). Calcule a média destes números e imprima o resultado, com precisão de 2 casas decimais.

```
#include <stdio.h>

int main(int argc, char **argv) {
```

Exercício

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a, b, c;
    float soma, media;

    printf("Entre com o valor de a, b, c: ");
    scanf("%d %d %d", &a, &b, &c);

    soma = a + b + c;
    media = soma/3;

    printf("\nA media desses números eh: %.2f\n", media);
}
```

Programa – Exemplo 1

Comprimento de uma circunferência

```
#include <stdio.h>
#define PI 3.1415
int main(int argc, char **argv) {
    float comp, r;

    printf("Digite o raio da circunferência: ");
    scanf("%f", &r);
    comp = 2 * PI * r;
    printf("\nO comprimento para uma circunferência
de raio %f é %f\n", r, comp);
}
```

Programa – Cabeçalho

Comprimento de uma circunferência

```
#include <stdio.h>
#define PI 3.1415
int main(int argc, char **argv) {
    float comp, r;
    printf("Digite o raio da circunferência: ");
    scanf("%f", &r);
    comp = 2 * PI * r;
    printf("\nO comprimento para uma circunferência
de raio %f é %f\n", r, comp);
}
```

Cabeçalho:

- inclusão de bibliotecas
- definição de constantes e variáveis

Programa – Cabeçalho

Comprimento de uma circunferência

```
#include <stdio.h>
#define PI 3.1415
```

Cabeçalho:

- inclusão de bibliotecas
- definição de constantes e variáveis

biblioteca **stdio.h** (*standard input/output*)

- contém variáveis e funções usadas para E/S

Programa – Corpo Principal

Comprimento de uma circunferência

```
#include <stdio.h>
#define PI 3.1415

int main(int argc, char **argv) {
    float comp, r;

    printf("Digite o raio da circunferência: ");
    scanf("%f", &r);
    comp = 2 * PI * r;
    printf("\nO comprimento para uma circunferência
    de raio %f é %f\n", r, comp);
}
```

Corpo principal:

- definição de variáveis
- seqüência de comandos para execução o programa em si

Programa – Corpo Principal

Comprimento de uma circunferência

```
#include <stdio.h>
#define PI 3.1415
int main(int argc, char **argv) {
```

main()

- função principal do programa
- é a única obrigatória no programa C

Corpo principal:

- definição de variáveis
- seqüência de comandos para execução o programa em si

Programa – Corpo Principal

Comprimento de uma circunferência

```
#include <stdio.h>
#define PI 3.1415

int main(int argc, char **argv) {
    float comp, r;

    printf("Digite o raio da circunferência: ");
    scanf("%f", &r);
    comp = 2 * PI * r;
    printf("\nO comprimento para uma circunferência
    de raio %f é %f\n", r, comp);
}
```

Corpo principal:

- definição de variáveis
- seqüência de comandos para execução o programa em si

Exercício

Faça um programa que leia a altura e peso de uma pessoa e imprima seu índice de massa corporal ($IMC = \text{peso} / \text{altura}^2$)



Exercício - Algoritmo

índice de massa corporal

Entrada de dados:

altura

peso

Saída

índice de massa corporal

Algoritmo

Informe a altura

Informe o peso

calcula $imc = peso / (altura * altura)$

imprime imc

Exercício - Solução

Índice de massa corporal (IMC = peso / altura²)

```
#include <stdio.h>

int main(int argc, char **argv) {
    float altura, peso, imc;

    printf("Digite sua altura e seu peso: ");
    scanf("%f %f", &altura, &peso);

    imc = peso / (altura * altura);

    printf("\nSeu IMC = %f\n", imc);
}
```

Exercício

Escreva um programa que troca o conteúdo de duas variáveis inteiras (a e b) e usando uma terceira variável (c). O programa deve ler o valor de a e b do teclado e imprimir o valor original de cada uma e o valor após a troca.

Exercício – 3 variáveis

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a, b, c;

    printf("Entre com o valor de a e b: ");
    scanf("%d %d", &a, &b);

    printf("Valor de a:%d. Valor de b:%d.", a,b);

    c = a;
    a = b;
    b = c;

    printf("Valor de a:%d. Valor de b:%d.", a,b);
}
```

Exercício

É possível fazer a troca

usando apenas 2 variáveis?

Exercício – 2 variáveis

```
#include <stdio.h>

int main(int argc, char **argv) {
    int a, b;

    printf("Entre com o valor de a e b: ");
    scanf("%d %d", &a, &b);

    printf("Valor de a:%d. Valor de b:%d.", a,b);

    a = a + b;
    b = a - b;
    a = a - b;

    printf("Valor de a:%d. Valor de b:%d.", a,b);
}
```