



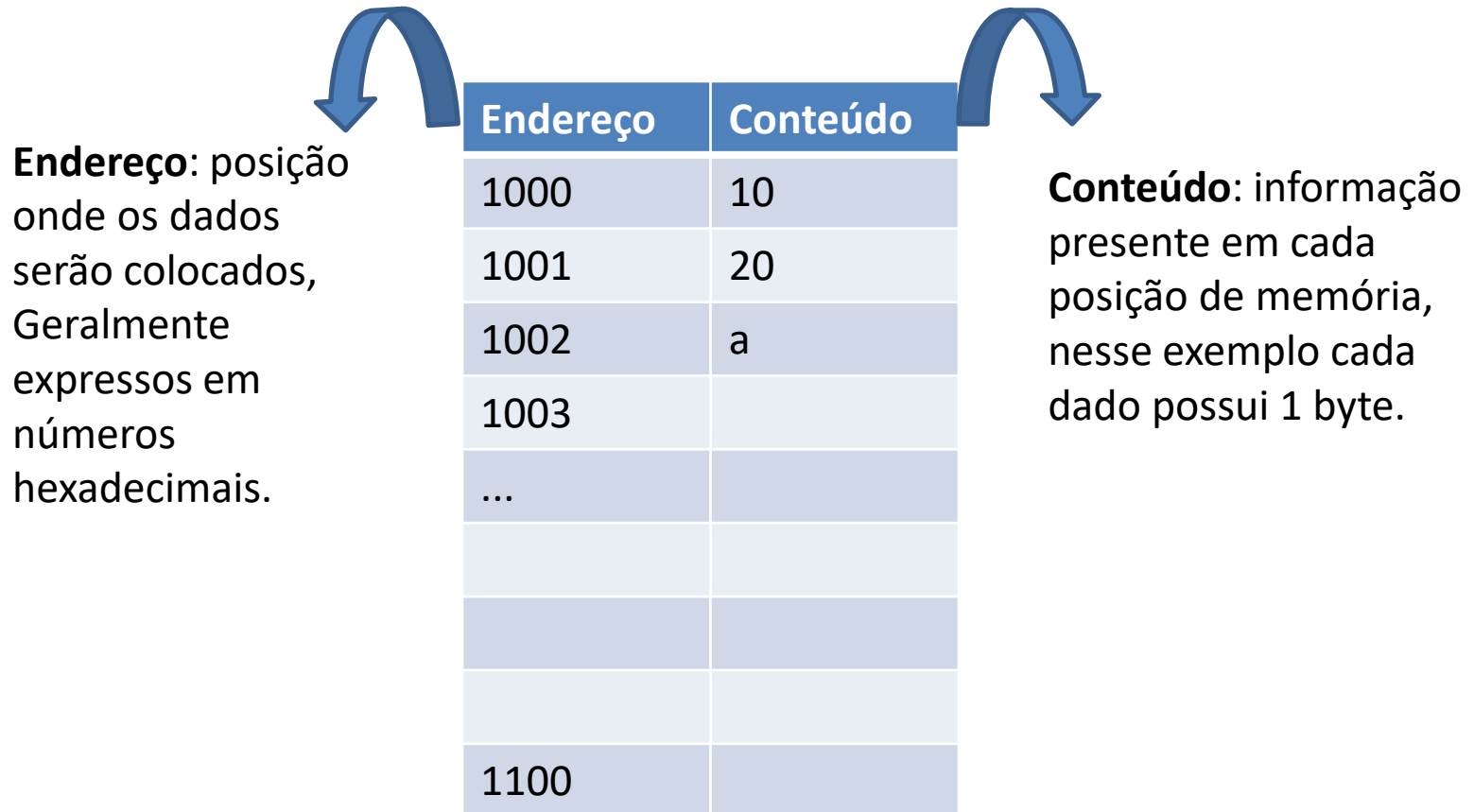
Variável, leitura e escrita de dados em C

Prof. Lilian Berton

São José dos Campos, 2019

Variáveis

Variáveis em um programa C estão associadas a **posições de memória que armazenam informações.**



Variáveis

Para declarar uma variável em C preciso definir um **tipo** e um nome para a mesma.

Endereço	Conteúdo
1000	10
1001	20
1002	a
1003	1.5
...	
1100	

```
int valor1 = 10;
```

```
int valor2 = 20;
```

```
char letra1 = 'a';
```

```
float valor3 = 1.5
```

Tipos de dados

- O tipo de uma variável define os valores que ela pode assumir e as operações que podem ser realizadas com ela.
- `char`: um byte que armazena o código de um caractere do conjunto de caracteres local.
- `int`: um inteiro cujo tamanho depende do processador, tipicamente 16 ou 32 bits.
- `float`: um número real com precisão simples.
- `double`: um número real com precisão dupla.

Abrangência dos dados (16 bits)

Tipo	Tamanho(bytes)	Abrangência		
char	1	-128	a	127
unsigned char	1	0	a	255
int	2	-32768	a	32767
unsigned int	2	0	a	65535
short int	2	-32768	a	32767
long int	4	-2.147.483.648	a	2.147.483.647
unsigned long	4	0	a	4.294.967.295
float	4	$-3,4 \cdot 10^{38}$	a	$3,4 \cdot 10^{38}$
double	8	$-1,7 \cdot 10^{308}$	a	$1,7 \cdot 10^{-308}$
long double	10	$-3,4 \cdot 10^{4932}$	a	$3,4 \cdot 10^{4932}$

Nomeando variáveis

1. O primeiro caractere tem que ser uma letra ou underscore “_” . O restante do nome pode conter letras, dígitos e sublinhados.
2. Palavras-chave de C não podem ser utilizadas como nome de variáveis: int, for, while, etc...
3. Não pode conter espaços em branco
4. Não pode conter letras com acentos
5. Não pode utilizar símbolos (exceto underscore)
6. C é case-sensitive: contador ≠ Contador ≠ CONTADOR

Corretos	Incorretos
Contador	1contador
Teste23	Oi!gente
Alto_Paraiso	Alto..paraiso
_sizeint	x*y

Atribuição

- Em C utilizamos o símbolo “=” para atribuir um valor para uma variável.

```
1
2  #include<stdio.h>
3
4  int main() {
5      int A = 10;
6      int B = 10;
7      printf("A soma desses numeros eh = %d",A+B);
8
9  }
10
```

Atribuição

```
1
2  #include<stdio.h>
3
4  int main() {
5      int A = 10;
6      int B = 10;
7      printf("A soma desses numeros eh = %d",A+B);
8
9  }
10
```

Endereço	Conteúdo	
1000	10	← A
1001	10	← B
1002		
1003		
...		
1100		

Atribuição

```
1
2  #include<stdio.h>
3
4  int main() {
5      int A = 10;
6      int B = 10;
7      int C = A + B;
8      printf("A soma desses numeros eh = %d",C);
9
10 }
11
```

Endereço	Conteúdo
1000	10
1001	10
1002	20
1003	
...	
1100	

← A

← B

← C

Escrevendo informações em C: printf()

- Instrução responsável por promover saída de dados. Sua forma geral é:
- `printf("tipo de saída", lista de variáveis)`
- Os “tipos de saída” a serem utilizados pelo `printf()` no momento serão os mesmos dos “tipos de entrada” do `scanf()`.
- Alguns “tipos de saída”:
 - `%c` – leitura de caractere (**char**)
 - `%d` – leitura de números inteiros (**int**)
 - `%f e %lf` – leitura de número reais (**float e double**)
 - `%s` – leitura de caracteres (**string**)

```
printf("%d",A);
```

```
Printf("Meu nome eh Lilian");
```

Lendo dados em C: scanf()

- Instrução responsável por promover leitura de dados. Sua forma geral é:
- `scanf("tipo de entrada", lista de variáveis)`
- Alguns "tipos de entrada":
 - `%c` – leitura de caractere (**char**)
 - `%d` – leitura de números inteiros (**int**)
 - `%f` e `%lf` – leitura de número reais (**float e double**)
 - `%s` – leitura de caracteres (**string**)

Cada variável a ser lida (scanf) deverá ser precedida pelo carácter & (endereço).

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

Exemplo 1: Programa soma

```
1
2  #include<stdio.h>
3
4  int main() {
5      int A;
6      int B;
7      printf("Insira dois numeros inteiros: \n");
8      scanf("%d %d",&A,&B);
9      printf("A soma desses numeros eh = %d",A+B);
10
11 }
12
```

Exemplo 1: Programa soma

```
Insira dois numeros inteiros:  
2  
2  
A soma desses numeros eh = 4  
-----  
Process exited after 3.78 seconds with return value 0  
Press any key to continue . . .
```

Exemplo 2: Programa idade

```
#include <stdio.h>
int main() {
    int idade; //variável que armazena idade
    printf ("Quantos anos vc tem?"); /*imprime msg na tela*/
    scanf ("%d", &idade);
    printf ("Então você nasceu em %d", 2019-idade);
    return 0;
}
```

- `//` e `/* */` representam comentários e não interfere no programa.
- `Idade` é uma variável: todo programa precisa de espaço na memória para poder trabalhar e as declarações reservam o espaço necessário para isso.
- A **função printf** imprime todos os caracteres que estão entre aspas. Para cada `"%d"`, printf imprime na tela um número inteiro, o qual aparece depois da vírgula.
- A **função scanf** lê um número inteiro pelo teclado e armazena em uma variável, para tal deve-se usar o símbolo `&nome-da-variável`.

Exemplo 3: Programa potência

- Dado um número, calcula seu quadrado

```
#include <stdio.h>
```

```
main(void) {
```

```
int numero;
```

```
printf("Digite um numero: ");
```

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

```
printf("%d elevado ao quadrado é igual a %d. \n", numero,  
numero*numero);
```

```
}
```

- Quais as variáveis? Qual o tipo dela?
- Qual a entrada do programa?
- Qual a saída do programa?

Escopo de uma variável

- Escopo define onde e quando uma variável pode ser usada.

```
#include <stdio.h>
int i = 0;      ➡ /* variável global */
                /* visível em todo arquivo */

main(void) {
    i++;
    int j = 0;  ➡ /* variável local */
                /* visível apenas na função */
    printf( "%d %d", i,j );
}
```


Escopo de uma variável

- Escopo define onde e quando uma variável pode ser usada.

```
#include <stdio.h>
int i = 0;      ➡ /* variável global */
                /* visível em todo arquivo */

main(void) {
    i++;
    int j = 0;   ➡ /* variável local */
    printf( "%d %d", i,j ); /* visível apenas na função*/
}
```



...e a este fenômeno, de um sinal contínuo no tempo representado por uma senóide de alta frequência adquirir a identidade de uma senóide de frequência mais baixa depois da amostragem, damos o nome de aliasing...

PQP!! Eu só queria aprender a fazer jogos!

TRANSFORMADA DE FOURIER

$$x_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j e^{\frac{2\pi i}{n} jk} \quad k = 0, \dots, n-1$$

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \quad j = 0, \dots, n-1$$

