

Comandos de Entrada e Saída

- Comunicação com Exterior
 - Não apenas teclado e mouse
- Equipamentos de Entrada
 - Teclado, mouse, discos, joystick, microfone, leitora ótica(scanner), célula foto-sensora, cd-player, câmera, etc...
- Equipamentos de Saída
 - Vídeo, vídeo (texto ou gráfico)
 - impressoras diversas
 - auto-falantes
 - discos
 - plotters
 - fita

Algumas Funções de Saída

- Função `Putchar()`
 - escrevem 'B' no vídeo
 - `c = 'b';`
 - `putchar (c + 'A' - 'a');`
- Função *printf ()*
 - *printf (cadeia de controle , outros argumentos);*

Especificação de formatos com Printf

Formato	Tipo da expressão
<code>%c</code>	Caractere
<code>%d , %i</code>	Inteiro decimal com 2 bytes
<code>%ld</code>	Inteiro decimal com 4 bytes
<code>%u</code>	Inteiro decimal, sem sinal
<code>%o</code>	Inteiro octal, sem sinal
<code>%x , %X</code>	Inteiro hexadecimal, sem sinal
<code>%f</code>	Real com 6 casas decimais
<code>%g</code>	Real com no máximo 6 dígitos significativos
<code>%e , %E</code>	Real em notação exponencial
<code>%s</code>	Cadeia de caracteres

Especificações do campo, exemplo:

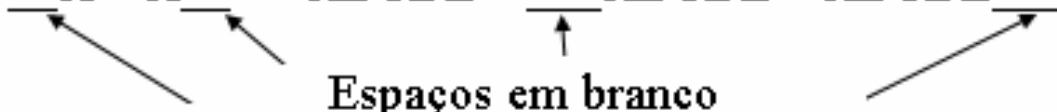
`%3c, %8d, %5.2f, ....`

Exemplos

```
printf ("***%c***%3c***%-3c***%s***%8s***%-8s***",  
        'A', 'A', 'A', "ABCDE", "ABCDE", "ABCDE");
```

o resultado exibido será:

***A** A**A ***ABCDE** ABCDE**ABCDE ***



Espaços em branco

```
printf ("***%c***%d***%c***%d***%4d***%-4d***",  
        'B', 'B', 69, 69, 35, 35);
```

o resultado exibido será:

B66***E***69***_35***35_***

```
printf ("***%c***%3c***%-3c***%s***%3s***%8s*%5.2f",  
        'A', 'A', 'A', "ABCDE", "ABCDE", "ABCDE", 1.3452);
```

Exemplo com Ponto Flutuante

```
printf("x =%20.10f", 1234.12344347263849);
```

resultado: $x = \frac{1234.1234434726}{20 \text{ casas}}$

```
printf("x = %10.7d", 1234);
```

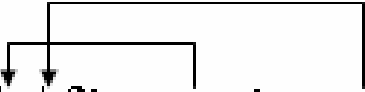
resultado: $x = \frac{0001234}{10 \text{ casas}}$

```
printf("cad =%10.6s", "ABCDEFGH")
```

resultado: $\text{cad}_{-} = \frac{\boxed{\text{ABCDEF}}}{10 \text{ casas}}$ 6 casas

Variação da Precisão

- Seja o comando


`printf("valor =%*.*f", expr1, expr2, expr3);`

O valor de *expr3* será impresso num espaço de *expr1* casas, com *expr2* dígitos depois do ponto decimal.

- Qual o resultado do programa abaixo?

```
void main () {  
    double x; int m, n;  
    x = 337.3777337337;  
    for (n = 1; n <= 4; n++) {  
        m = 8 + n;  
        printf ("### %*.*f ###\n", m, n, x);  
    }  
}
```

Algumas Funções de Entrada

- Funções: getch, getche e getchar
Leitura de um caracter.

- Função Scanf:

- Forma:

- scanf (*cadeia de controle, outros argumentos*);

```
char a, b; int n; double x;  
scanf ("%c%c%d%lf", &a, &b, &n, &x);
```

cadeia de controle: "%c%c%d%lf"
outros argumentos: &a, &b, &n, &x

Os outros
argumentos
devem ser
endereços

Alguns Problemas de Leitura

- Qual a impressão feita para a entrada: abcde e abcde. E se houvesse getchar no lugar de getch ?

```
main() {  
    char c;  
    printf ("Escreva uma sentenca encerrada por '\nenter':");  
    scanf("%c",&c);  
    printf("Escreva outra senteca terminada por enter:");  
    while ((c=getch()) != '\n' && c!= '\r') {  
        putchar (c);  
    }  
}
```

Buffer de Teclado

- As funções ***getchar()*** e ***scanf()*** não fazem leitura diretamente do teclado; elas lêem do “**buffer do teclado**”. Todos os caracteres digitados, inclusive o “enter”, ficam guardados no buffer
- ***getchar()*** e ***scanf()*** fazem a leitura, ao ser digitado “Enter”
- Uma leitura pode não consumir todo o buffer; se o buffer não estiver vazio no início de uma leitura, tais funções fazem a leitura sem aguardar digitação por parte do operador.

Mais sobre scanf...

- A leitura de um caractere incompatível com o formato de leitura de um argumento:
 - interrompe a leitura do argumento;
 - mantém o caractere no buffer de teclado (ou buffer de leitura);
 - dá ao argumento o valor formado até então.
- O que será impresso pelos comandos abaixo, dada a entrada: 123wex<enter>

```
char c; long a;  
scanf ("%ld%c", &a, &c);  
printf ("a: %ld; c: %c***\n", a, c);  
scanf ("%c", &c);  
printf ("c: %c***\n", c);  
scanf ("%ld", &a); scanf ("%c", &c);  
printf ("a: %ld; c: %c***\n", a, c);
```

Entrada e saída em disco

- Todo arquivo precisa ser associado a uma variável do tipo **FILE** do programa, para ser acessado por um programa.
- Comandos para essa associação:

```
FILE *fl;
```

```
fl = fopen ( nome_do_arquivo, modo);
```

Modo	Significado
"r"	Abrir arquivo para leitura
"w"	Abrir arquivo para escrita
"a"	Abrir arquivo para aumentá-lo
"r+"	Abrir arquivo para leitura e escrita
"w+"	Abrir arquivo para escrita e leitura

Comandos para Leitura e Escrita em Arquivos

- Comando para leitura de arquivos:
 - **fscanf** (*Variável_File*, *Cadeia_de_Controle*, *Outros_Argumentos*);
- Comando para escrever em arquivo:
 - **fprintf** (*Variável_File*, *Cadeia_de_Controle*, *Outros_Argumentos*);

Mais Comandos

- Comando para fechar arquivos:
 - **fgetc**(*Variável_File*);
 - retorna um caracter do arquivo
 - **fclose** (*Variável_File*);
 - desfaz a associação entre a *Variável_File* do programa e o arquivo do diretório.

Exemplo de Manipulação de Arquivos

```
int a, soma; FILE *f1, *f2;  
soma = 0;  
f1 = fopen ("infile1.cpp", "r");  
f2 = fopen ("outfile1 ", "w");  
while (fscanf(f1, "%d", &a) == 1) soma += a;  
fprintf (f2, "A soma eh %d", soma);  
fclose (f1); fclose (f2);
```

Conteúdo de infile1.cpp:
23 56 24 1239 234

Conteúdo de outfile1:
A soma eh 1576