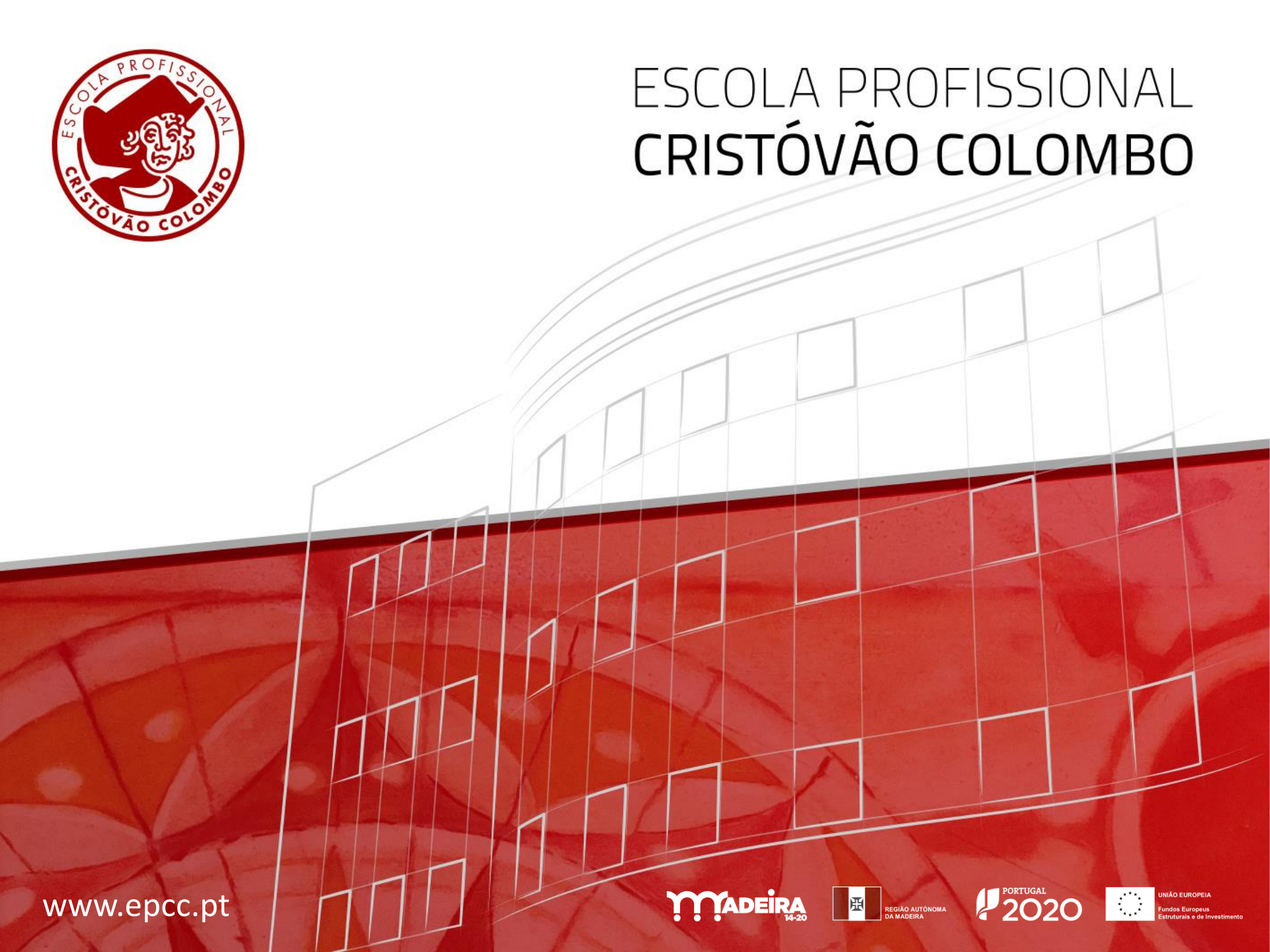
ESCOLA PROFISSIONAL
CRISTÓVÃO COLOMBO



www.epcc.pt

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA · Fundos Europeus Estruturais e de Investimento

Sistemas Operativos – Processamento Computacional

Instruções de Entrada/Saída

Autor da apresentação

09/09/2014

Saída de dados

Em C, para apresentar(imprimir) mensagens no monitor pode utilizar-se a função printf. Esta função está definida na biblioteca stdio.h e permite diversos argumentos.

Sintaxe:

printf(string_de_controlo [, arg1, arg2, ...])

Saída de dados

Os argumentos entre **[]** são opcionais. A *string* _de_controlo pode conter os caracteres a mostrar, sequências de escape e especificadores de conversão. Estes especificadores de conversão permitem definir como os argumentos vão ser mostrados no monitor, devendo existir um argumento por cada especificador de conversão.

Saída de dados

Cada especificador de conversão começa pelo carácter '%' e termina com um carácter de conversão.

```
char nome[80] = "Sergio";

printf("Olá: %s",nome);
```

Saída de dados - tabela

Código	Conversão/Formato do argumento
`%d`	Número decimal inteiro (int). Também pode ser usado `%i` como equivalente a `%d`.
`%u`	Número decimal natural (unsigned int), ou seja, sem sinal.
`%o`	Número inteiro representado na base octal. Exemplo: 41367 (corresponde ao decimal 17143).
`%x`	Número inteiro representado na base hexadecimal. Exemplo: 42f7 (corresponde ao decimal 17143). Se usarmos `%X`, as letras serão maiúsculas: 42F7.
`%X`	Hexadecimal com letras maiúsculas
`%f`	Número decimal de ponto flutuante. No caso da função printf, devido às conversões implícitas da linguagem C, serve tanto para **float** como para **double**. No caso da função scanf, `%f` serve para **float** e `%lf` serve para **double**.

Saída de dados - tabela

%e	Número em notação científica, por exemplo `5.97e-12`. Podemos usar %E para exibir o E maiúsculo (`5.97E-12`).
%E	Número em notação científica com o "e"maiúsculo
%g	Escolhe automaticamente o mais apropriado entre %f e %e. Novamente, podemos usar %G para escolher entre %f e %E.
%p	Ponteiro: exibe o endereço de memória do ponteiro em notação hexadecimal.
%c	Caractere: imprime o caractere que tem o código ASCII correspondente ao valor dado.
%s	Sequência de caracteres (string, em inglês).
%%	Imprime um %

Saída de dados

Carácter de escape são interpretados quando contidos numa string.

Exemplo:

```
printf("Olá \nMundo!");
```

Saída:

Olá
Mundo!

Entrada de dados

Para a entrada de dados existe uma função análoga a printf, a função scanf com a sintaxe:

scanf(string_de_controlo [, arg1, arg2, ...])

scanf lê caracteres da entrada padrão (o teclado) interpretando-os segundo o especificado na *string*_de_controlo e guardando os resultados nos argumentos.

Entrada de dados

Os argumentos têm que ser obrigatoriamente os endereços das variáveis onde se pretende armazenar os resultados. Para indicar o endereço de uma variável usa-se o operador & antes do nome da variável. Nas variáveis do tipo *array*, o nome da variável é já o endereço do seu primeiro elemento, pelo que não é necessário usar o operador &.

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundos Europeus Estruturais e de Investimento

Entrada de dados

A *string*_de_controlo pode conter:

- Espaços ou tabulações, que são ignorados.
- Caracteres normais (não '%'), que deverão corresponder aos próximos caracteres não brancos da sequência introduzida.
- Especificadores de conversão, que têm um formato semelhante ao já indicado para a função printf .

Entrada de dados - Exemplo

```
void main(){
     int x = 0;        /* variáveis declarada e inicializada */
     char nome[80];
     scanf("%d %s", &x,nome);

     printf("Olá: %d : %s", x, nome);
}
```

```
35 sergio
Olß: 35 : sergio
--------------------------------
Process exited after 10.5 seconds with return value 16
Press any key to continue . . .
```

Biblioteca stdlib.h e locale.h

O nome "*stdlib*" vem de **standard library**.

A biblioteca *stdlib* possui funções envolvendo alocação de memória, controle de processos, conversões e outras.

Método system - executa um comando externo.

Locale.h - biblioteca de idiomas.

Incluir as bibliotecas:

```
#include <locale.h> /* biblioteca de idiomas*/
#include <stdlib.h>
```

Biblioteca *stdlib*- Exemplo

```
//configurar idioma para pt-pt
setlocale(LC_ALL,"Portuguese");
printf(" Olá o programa espera um instrução. \n");
system("pause");
printf(" Alguma tecla foi pressionada.\n");
printf(" Volta a esperar novas instruções.\n");
system("pause");
printf(" Alguma tecla foi pressionada.\n");
```

```
 Olá o programa espera um instrução.
Press any key to continue . . .
 Alguma tecla foi pressionada.
 Volta a esperar novas instruções.
Press any key to continue . . .
 Alguma tecla foi pressionada.
```

Biblioteca *stdlib.h*

Alterar a cor do texto e do fundo do terminal.

```
//alteração da cor do texto e background
/*Código de cores (hexadecimal):
    0 - preto              1 - azul          2 - verde
    3 - verde claro        4 - vermelho      5 - roxo
    6 - amarelo            7 - branco        8 - cinza
    9 - azul claro
    A - verde claro        B - verde com transparencia
    C - vermelho claro     D- lilás
    E - Amarelo claro      F- branco brilhante
    Grupo de cores:
    color-> com uma instrução apenas altera letra
            com grupo alterar background e letra
    Exemplo EB : fundo : Amarelo claro, cor : verde com transparencia
    */
system("color E4");
printf(" Olá o programa espera um instrução. \n");
system("pause");
```