

# Aula 12

## Arquivos

**Responsável**

Prof. Seiji Isotani ([sisotani@icmc.usp.br](mailto:sisotani@icmc.usp.br))

## **Agenda:**

- **Uso de Arquivos:  
Conceitos, Tipos e Funções de manipulação de arquivos**
- **Arquivos texto e arquivos binários: Funções de  
Abertura, Leitura, Escrita e Fechamento de Arquivos**
- **Funções Adicionais. Exemplos Práticos**
- **Exercícios**

## Arquivos: Conceitos e Funções

### Tipos de Dispositivos de E/S:

Stdin, Stdout, Stderr, Arquivos

### Tipos de Dados em Arquivos:

Dados Simples (int, float, double, char, ...), Vetores e Dados Mistos

### Tipos de Arquivos:

Arquivo Texto, Arquivo Binário

### Funções de Manipulação de Arquivos:

Abertura de arquivos ..... fopen

Leitura e Escrita ..... fscanf / fprintf + fgets / fputs

Fechamento de arquivos ... fclose

Rotinas complementares ... feof , fflush, fchdir

## Arquivos em “C”:

### Tipos de Dispositivos de E/S

`scanf` => **`stdin`** (Arquivo: Standard Input ) ~ Teclado

`printf` => **`stdout`** (Arquivo: Standard Output ) ~ Monitor

mensagens de erro => **`stderr`** (Standard Error )

Todos Dispositivos de Entrada e Saída (I/O)

são vistos como ARQUIVOS.

STDIN, STDOUT, STDERR => Arquivos padrão, abertos e prontos para o uso

Exemplos:

```
fprintf(stdout, “Hello World!\n”); /* fprintf = file-printf */
```

## Arquivos em “C”:

### Tipos de Arquivos

#### ARQUIVO TEXTO (.txt)

**Armazenam dados na forma de texto (caracteres ASCII);**

- Podemos abrir um Arquivo TEXTO em um editor de textos, como o Notepad, o Wordpad ou o VI, e ler o seu conteúdo;
- Um número é escrito usando dígitos em ASCII, por exemplo: 123 é armazenado como os caracteres '1' '2' e '3' em seqüência formando uma seqüência de caracteres (string);
- O número 12345 ocupará portanto 5 bytes (5 caracteres no arquivo).

*Vantagens do arquivo texto:*

- Fácil de editar, inserir, remover conteúdo usando um editor de textos
- Fácil de visualizar e verificar o conteúdo que foi gravado

## Arquivos em “C”:

### Tipos de Arquivos

#### ARQUIVO BINÁRIO (.dat, .bin)

**Armazenam dados na forma binária (bytes, ints, doubles, etc);**

- **NÃO** podemos abrir um Arquivo BINÁRIO em um editor de texto, pois veremos apenas caracteres esquisitos;
- Um número é armazenado na sua forma binária, por exemplo:  
123 é armazenado como o **byte de valor 123 (ou seja 0111 1011)**, que corresponde ao **caracter '{' na tabela ASCII !**
- O número 12345 ocupará somente 2 bytes (se for um short int).

#### *Desvantagens do arquivo binário:*

- **Difícil de editar seu conteúdo pois não funciona no editor de textos**
- **Difícil de visualizar e verificar o conteúdo que foi gravado**

FORMATO	VANTAGENS	DESVANTAGENS
Texto	<ul style="list-style-type: none"> <li>• Legibilidade</li> <li>• Pode ser editado com editor de texto</li> <li>• Arquivo pode ser facilmente transferido para outra plataforma</li> </ul>	<ul style="list-style-type: none"> <li>• Representação numérica pode não ser precisa</li> <li>• Pode ocupar muito mais espaço</li> </ul>
Binário	<ul style="list-style-type: none"> <li>• Não existe erro de conversão de valores numéricos</li> <li>• Leitura ou escrita de arquivos é mais rápida</li> <li>• Pode ocupar menos espaço</li> </ul>	<ul style="list-style-type: none"> <li>• Transferência para outra plataforma pode ser problemático</li> </ul>

## Manipulação de Arquivos

1. Incluir o `<stdio.h>`: `#include <stdio.h>`

2. Declarar uma Variável do Tipo Arquivo (ponteiro para arquivo)

```
FILE *<nome_variável>; /* Arquivos já existentes: stdin, stdout, stderr */
```

3. Funções e Procedimentos pré-definidos na Linguagem C para a Manipulação de Arquivos: `<stdio.h>`

*fopen* - Abre um arquivo para leitura, escrita ou alteração

*fclose* - Fecha o arquivo

*fprintf* - Escreve dados em um arquivo

*fscanf* - Lê dados de um arquivo

*feof* - Testa para ver se encontramos o EoF (*End-of-File*)

Exemplo: `# include <stdio.h>`

```
FILE *Arquivo;
```

```
FILE *Arquivo; char Texto[100];
```

```
Arquivo = fopen ("dados.txt","w");
```

```
Arquivo = fopen ("dados.txt","r");
```

```
fprintf (Arquivo,"Hello World!");
```

```
fscanf (Arquivo, "%s", Texto);
```

```
fclose (Arquivo);
```

```
fclose(Arquivo);
```

## Manipulação de Arquivos

**Funções e Procedimentos da Linguagem para manipular arquivos**

**fopen - Abrir arquivo**

**Abre um arquivo para leitura, escrita ou alteração**

```
FILE *fopen (const char *filename, const char *mode);    /* modes: b, r, w, a, + */
```

```
Exemplo:  arqptr = fopen (NomeArq, "w");
```

```
          if (arqptr == NULL) { printf ("\n\nErro no arquivo!\n\n"); exit(0); }
```

```
FILE *fopen ( <nome_arquivo>, <modo_abertura> )
```

```
Exemplo: FILE *arquivo;  arquivo = fopen ("c:\arquivo.txt", "rt");
```

```
Parâmetros: modo_abertura => r (read), w (write), a (append), + (r/w)
```

```
          b(binário), t (texto). Exemplos: "rt", "wb", "r+" ...
```

```
Retorno:  NULL se ocorrer um erro na abertura, !(NULL) se funcionar
```

**fclose - Fechar arquivo**

**Fecha um arquivo que foi previamente aberto com o fopen**

**Não esqueça de fechar arquivos "w"! Salva o conteúdo, data, tamanho,...**

```
int fclose (FILE *file);
```

```
Exemplo:  arqptr = fopen (NomeArq, "w");    ...  fclose (arqptr);
```

```
          arqptr = fopen (NomeArq, "r");    ...  fclose (arqptr);
```

## Manipulação de Arquivos

```
FILE *fopen ( <nome_arquivo>, <modo_abertura> )
```

### Modo de abertura:

**r** : Leitura

**w** : Escrita

**r+** : Leitura e Escrita

**w+** : Leitura e Escrita. Caso o arquivo não existe, ele é criado.

**a** : Escrita inicia no fim do arquivo. Caso o arquivo não existe, ele é criado.

**a+** : Leitura e escrita (no fim do arquivo). Caso o arquivo não existe, ele é criado.

## Manipulação de Arquivos

Funções e Procedimentos da Linguagem para manipular arquivos

**fprintf** - Escrever no arquivo [printf]

```
int fprintf (FILE *file, const char *format, ...);
```

```
Exemplo: fprintf (arqptr, "Gravando 1, 2, %d no arquivo: %s",3,NomArq);
```

**fscanf** - Ler do arquivo [scanf]

```
int fscanf (FILE *file, const char *format, ...);
```

```
Exemplo: fscanf (arqptr,"% d %d %d", &dia, &mes, &ano);
```

**feof** - Testar se chegou ao final do arquivo

```
int feof (FILE *file);
```

```
Exemplo: while ( !feof(arqptr) )  
        { fscanf(arqptr,"%c",&character); printf("%c",character); }
```

## Manipulação de Arquivos

### Exemplo:

```
#include <stdio.h>
#define NomeArq "Arquivo.txt"

FILE *ArqIn;
char character;

int main()
{
    if ( ArqIn=fopen(NomeArq,"r") )
        {
            while ( !feof(ArqIn) )
                {
                    fscanf(ArqIn,"%c",&character);
                    printf("%c",character);
                }
            fclose(ArqIn);
        }
    else printf ("Erro na leitura do arquivo de entrada!");
}
```

exibe-texto.c

le-palavras.c

```
#include <stdio.h>
FILE *arqtxt;

int main () {
    char palavra[256];
    arqtxt = fopen ( "arquivo.txt", "r");

    if (arqtxt == NULL){

        printf ("ERRO!\n");
        exit(1);
    }
    while ( ! feof ( arqtxt ) ){
        fscanf (arqtxt, "%s", palavra);
        printf ("%s ",palavra);
    }

    fclose (arqtxt);
}
```

grava-palavras.c

```
#include <stdio.h>
FILE *arqtxt;

int main () {
    char palavra[256];
    arqtxt = fopen ( "grava.txt", "w");

    if (arqtxt == NULL){
        printf ("ERRO!\n");
        exit(1);
    }
    while ( ! feof ( arqtxt ) ){
        scanf ("%s",palavra);
        fprintf(arqtxt, "%s ", palavra);
        if (strcmp(palavra,"FIM") == 0)
            break;
    }
    fclose (arqtxt);
}
```

salva-texto.c

## Exemplo:

```
#include <stdio.h>

#define NomeArq "Texto.txt"

int main(){
    FILE *ArqOut;
    char Texto[100];
    char codigo;
    if ( ArqOut=fopen(NomeArq,"w") ){
        codigo=' ';
        while (codigo != '#'){
            scanf("%s",Texto);
            fprintf(ArqOut,"%s\n",Texto);
            codigo=Texto[0];
        }
        fclose(ArqOut);
    }else
        printf ("Erro na criacao do arquivo de saida!");
}
```

## Manipulação de Arquivos

**Funções e Procedimentos da Linguagem para manipular arquivos**

**fgets - Ler uma linha de texto do arquivo [gets]**

Lê uma string terminada por um '\n'. Acrescenta um '\0' na string.

**char \*fgets (char \*buffer, int maxlength, FILE \*file);**

**Exemplo: fgets (texto, 255, arqptr);**

**fputs - Escrever uma linha de texto no arquivo [puts]**

Escreve uma string, substituindo o '\0' por um '\n'.

**int fputs (const char \*string, FILE \*file);**

**Exemplo: fputs (texto, arqptr);**

**NULL = Ponteiro nulo ( \0 ) - Erro de abertura de arquivo [fopen]**

**fwrite, fread - Arquivos Binários (blocos de dados)**

**sprintf - Escrever para dentro de uma string [printf]**

**int sprintf(char \*buffer, const char \*format, ...);**

**Exemplo: sprintf (texto, "Valor total - R\$%.2lf", VTotal);**

**sscanf - Ler dados de dentro de uma string [scanf]**

**int sscanf (const char \*string, const char \*format, ...);**

**Exemplo: sscanf(texto, "%d %d %d", &dia, &mes, &ano);**

```

#include <stdio.h>
#define NomeArq "Texto.txt"

int main() {
    FILE *ArqOut;
    char Texto[100];
    char codigo;

    if ( ArqOut=fopen(NomeArq,"w") ){
        codigo=' ';
        while (codigo != '#'){
            gets(Texto);
            fputs(Texto, ArqOut);
            fputs("\n",ArqOut);
            codigo=Texto[0];
        }
        fclose(ArqOut);
    }else
        printf ("Erro na criacao do arquivo de saida!");
}

```

salva-texto1.c

## Exemplos de Uso de Arquivos: Grava no arquivo “Hello World!”

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

int main(){
    FILE *Arquivo;
    Arquivo=fopen("saida.txt","w");
    if (Arquivo == NULL)
        exit(0);

    fprintf(Arquivo,"\n Hello World! \n");
    fclose(Arquivo);
    return 0;
}
```

Hello-Arquivo.c  
Arq.: saida.txt

## Exemplos de Uso de Arquivos: Lê uma palavra do arquivo

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

```
Entrada-Arquivo.c
Arq.: entrada.txt
```

```
int main(){
    FILE *Arquivo;
    char nome[30];

    Arquivo=fopen("entrada.txt","r");
    if (Arquivo == NULL)
        exit(0);

    printf("Qual o seu nome: ");
    fscanf(Arquivo,"%s",nome);
    printf("Nome: %s\n",nome);

    fclose(Arquivo);
    return 0;
}
```

## Exemplos de Uso de Arquivos: Lê 10 números de um arquivo

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

Le-Numeros.c  
Arq.: numeros.txt

```
int main(){
    FILE *Arquivo;
    int Vetor[10], i;

    Arquivo=fopen("numeros.txt","r");
    if (Arquivo == NULL) exit(0);

    for (i=0; i < 10; i++) /* Le 10 numeros de um arquivo */
        fscanf(Arquivo,"%d", &Vetor[i]);

    for (i=0; i < 10; i++) /* Exibe na tela os numeros lidos */
        printf("Numero [%d]: %d\n", i, Vetor[i]);

    fclose(Arquivo);
    return 0;
}
```

## Uso de Arquivos: Grava 10 números em um arquivo

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

Salva-Numeros.c  
Arq.: nros-salvos.txt

```
int main(){
    FILE *Arquivo;
    int Vetor[10], i;

    Arquivo=fopen("nros-salvos.txt","w");
    if (Arquivo == NULL) exit(0);

    for (i=0; i < 10; i++){ /* Le 10 numeros digitados pelo teclado */
        printf ("Entre com o nro. %d : ", i);
        scanf("%d", &Vetor[i]);
    }
    for (i=0; i < 10; i++) /* Grava em disco os 10 numeros lidos */
        fprintf(Arquivo,"%d\n", Vetor[i]);

    fclose(Arquivo);
    return 0;
}
```

## Exercício

1) Um arquivo contém 10 linhas e em cada linha temos 20 inteiros separados por espaços. Leia esses números e coloque em uma matriz  $M[10][20]$ . Salve essa matriz em outro arquivo chamado `matriz.txt`