



## USP - ICMC - SSC SSC 0301 - 2o. Semestre 2013

### Disciplina de Introdução à Computação para Engenharia Ambiental

**Prof. Dr. Fernando Santos Osório / Prof. Artur Lovato Cunha**

**LRM - Laboratório de Robótica Móvel do ICMC / CROB-SC**

**Email: [fosorio@icmc.usp.br](mailto:fosorio@icmc.usp.br) ou [fosorio@gmail.com](mailto:fosorio@gmail.com)**

**Email: [arturlc@icmc.usp.br](mailto:arturlc@icmc.usp.br)**

**Página Pessoal: <http://www.icmc.usp.br/~fosorio/>**

**Material on-line:**

**Wiki ICMC - <http://wiki.icmc.usp.br/index.php>**

**Wiki SSC0301 - [http://wiki.icmc.usp.br/index.php/SSC-301-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-301-2013(fosorio))**

## **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**

**Informações Complementares e Atualizadas:**  
**Consulte REGULARMENTE**  
**O material disponível na COTEIA**

## 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 [nós vamos usar arquivos texto!]

### Funções de Manipulação de Arquivos:

Abertura de arquivos ..... fopen

Leitura e Escrita ..... fscanf / fprintf + fgets / fputs (*texto*)

Leitura e Escrita ..... fwrite / fread (*binário*)

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 ) ~ Monitor

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 ← *Tipo de arquivo a ser usado nesta disciplina*

### 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 → *Uso opcional*

#### 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**, 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

## 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;
```

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

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

```
fclose (Arquivo);
```

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

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

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

```
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, t, r, w, a, + */
```

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



## 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, t, r, w, a, + */
```

```
Exemplo: arqptr = fopen (NomeArq, "wt");  
if (arqptr == NULL) { printf ("\n\nEr
```

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

```
Exemplo: FILE *arquivo; arquivo = fopen ("c:\
```

```
Parâmetros: modo_abertura => r (read), w (write)
```

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

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

Modos:

**b = binário** Ex.: 123 => Binário (byte)

**t = texto** Ex.: 123 => ASCII (texto)

**r = read** '1' '2' '3'

**w = write**

**a = append**

**+ = read/write**

#### 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, "wt"); ... fclose (arqptr);
```

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

## Manipulação de Arquivos: Texto

Funções e Procedimentos da Linguagem para manipular arquivos texto  
Arquivos Texto → Serão adotados nesta disciplina

**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: Binários

Funções e Procedimentos da Linguagem para manipular arquivos binário  
Arquivos Binários → Seu uso será optativo em nossa disciplina

**fwrite**- Escrever no arquivo binário

```
size_t fwrite(void *ptr, size_t size, size_t nmemb, FILE *stream);
```

```
Exemplo: fwrite (Vetor, sizeof(int), 10, ArqPtr);
```

**fread** - Ler do arquivo binário

```
size_t fread(const void *ptr, size_t size, size_t nmemb, FILE *stream);
```

```
Exemplo: fread (Vetor, sizeof(int), 10, ArqPtr);
```

**Exemplos:**

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

```
FILE *ArqPtr;
```

```
int VetInt[5] = {1,2,3,4,5}
```

```
ArqPtr=fopen("arq.bin","rb");  
fread(VetInt,sizeof(int),5,ArqPtr);  
fclose(ArqPtr);
```

```
ArqPtr=fopen("arq.bin","wb");  
fwrite(VetInt,sizeof(int),5,ArqPtr);  
fclose(ArqPtr);
```

## Manipulação de Arquivos: Texto

### Exemplo:

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

FILE *ArqIn;
char character;

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

exibe-texto.c

## Manipulação de Arquivos

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

main ()
{
    char palavra[256];

    arqtxt = fopen ( "arquivo.txt", "rt");
    if (arqtxt == NULL)
    {
        printf ("ERRO!\n");
        exit(1);
    }

    while ( ! feof ( arqtxt ) )
    {
        fscanf (arqtxt, "%s", palavra);
        printf ("%s ",palavra);
    }

    fclose (arqtxt);
}
```

le-palavras.c

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

main ()
{
    char palavra[256];

    arqtxt = fopen ( "grava.txt", "wt");
    if (arqtxt == NULL)
    {
        printf ("ERRO!\n");
        exit(1);
    }

    for ( ; ; )
    {
        scanf ("%s",palavra);
        fprintf(arqtxt, "%s ", palavra);
        if (strcmp(palavra,"FIM") == 0)
            break;
    }

    fclose (arqtxt);
}
```

grava-palavras.c

## Manipulação de Arquivos

### Exemplo:

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

FILE *ArqOut;
char Texto[100];
char codigo;

main()
{
    if ( ArqOut=fopen(NomeArq,"wt") )
    {
        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!");
}
```

salva-texto.c

## Manipulação de Arquivos: Texto

**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);

## Manipulação de Arquivos: Texto

### Exemplo:

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

FILE *ArqOut;
char Texto[100];
char codigo;

main()
{
    if ( ArqOut=fopen(NomeArq,"wt") )
    {
        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>

FILE *Arquivo;

int main(int argc, char *argv[])
{

    Arquivo=fopen("saida.txt","wt");
    if (Arquivo == NULL) exit(0);

    fprintf(Arquivo,"\n Hello World! \n");

    fclose(Arquivo);

    system("PAUSE");
    return 0;
}
```

Hello-Arquivo.c  
Arq.: saida.txt

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

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

FILE *Arquivo;

int main(int argc, char *argv[])
{
    char nome[30];

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

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

    fclose(Arquivo);

    system("PAUSE");
    return 0;
}
```

Entrada-Arquivo.c  
Arq.: entrada.txt

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

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

```
FILE *Arquivo;
int Vetor[10];
```

```
int main(int argc, char *argv[])
{
    int i;
```

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

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

Le-Numeros.c  
Arq.: numeros.txt

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

    fclose(Arquivo);

    system("PAUSE");
    return 0;
}
```

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

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

FILE *Arquivo;
int Vetor[10];

int main(int argc, char *argv[])
{
    int i;

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

    /* Le 10 numeros digitados pelo teclado */
    for (i=0; i < 10; i++)
    {
        printf ("Entre com o nro. %d : ", i);
        scanf("%d", &Vetor[i]);
    }
}
```

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

```
/* Grava em disco os 10 numeros lidos */
for (i=0; i < 10; i++)
    fprintf(Arquivo,"%d\n", Vetor[i]);

fclose(Arquivo);

system("PAUSE");
return 0;
}
```

## Exercícios - Uso de Arquivos:

1. Faça um programa que leia um arquivo texto contendo uma sequência de valores, com um valor em cada linha do arquivo. Exibir os valores lidos do arquivo aplicando um reajusta de 10% sobre cada valor.

Arquivo de Entrada:

```
123.00  
456.00  
100.00
```

Saída na tela:

```
135.30  
501,60  
110.00
```

2. Faça um programa que leia um arquivo texto do disco, lendo linha a linha, e exibindo cada uma das linhas numeradas na tela. A idéia é podermos pegar um arquivo um arquivo texto qualquer (pode ser inclusive o arquivo do programa fonte – arquivo “.c” – de um destes exercícios) e mostrar na tela com as linhas numeradas. As primeiras linhas do arquivo iriam ser exibidas na tela da seguinte forma:

```
1: #include <stdio.h>  
2: #include <conio.h>  
3:  
4: FILE *ArqEntrada;  
5: ...
```

Usar: fgets e fputs

3. Faça um programa que peça para o usuário entrar um texto que deseja procurar (uma palavra) e que em seguida leia um arquivo texto do disco e procure por esta palavra no arquivo. Caso seja encontrada a palavra digitada pelo usuário em alguma das linhas do arquivo texto lido do disco, o programa deverá exibir na tela o número da linha do arquivo onde encontrou esta ocorrência da palavra, bem como o conteúdo da linha onde se encontra a palavra procurada. Dica: use o comando “strstr” para procurar a palavra.



## INFORMAÇÕES SOBRE A DISCIPLINA

**USP - Universidade de São Paulo - São Carlos, SP**  
**ICMC - Instituto de Ciências Matemáticas e de Computação**  
**SSC - Departamento de Sistemas de Computação**

**Prof. Fernando Santos OSÓRIO**

**Web institucional: <http://www.icmc.usp.br/>**

**Página pessoal: <http://www.icmc.usp.br/~fosorio/>**

**Página do Grupo de Pesquisa: <http://www.lrm.icmc.usp.br/>**

**E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com**

**Disciplina de Introdução a Computação – Eng. Ambiental**

**WIKI - [http://wiki.icmc.usp.br/index.php/SSC-301-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-301-2013(fosorio))**

- > Programa, Material de Aulas, Critérios de Avaliação,**
- > Trabalhos Práticos, Datas das Provas, Notas**