

Fundamentos de Programação 1

Linguagem C “Arquivos Seqüências ou de Texto”.

Slides 18

Prof. SIMÃO

Arquivo de Escrita 1

```
fopen ( "nome.txt", "w" );
```

```
fputc ( caracter, arquivo);
```

Arquivo de Escrita 1

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;

    char ch;

    fp = fopen ( "teste.txt", "w" );

    if ( NULL == fp )
    {
        printf ( "O arquivo não pode ser aberto. \n" );
        system ( "Pause" );
        exit (1);
    }

    printf ( " Programa para gravar caracteres em um arquivo chamado teste.txt." );
    printf ( " Digite $ para fechar o arquivo." );

    printf ( " \n \n " );

    printf ( "Digite os caracteres: \n" );

    do
    {
        ch = getchar();
        fputc ( ch, fp );
    } while ( ch != '$' );

    fclose ( fp );

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

Arquivo de Leitura 1

```
fopen ( "nome.txt", "r" );
```

```
fgetc ( arquivo );
```

Arquivo de Leitura 1

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

int main()
{
    FILE *fp;
    char ch;
    char nome[100];

    printf ( "Informe o nome do arquivo a ser lido: \n" );
    gets ( nome );

    // nome = strcat(nome, ".txt");

    fp = fopen ( nome, "r" );

    if ( NULL == fp)
    {
        printf ( "O arquivo não pode ser aberto. \n" );
        system ( "Pause" );
        exit (1);
    }

    ch = fgetc ( fp );

    while ( ch != EOF )
    {
        putchar( ch ); /* Imprime na tela */
        ch = fgetc ( fp );
    }

    printf ("\n");
    system ("Pause");

    return 0;
}
```

Arquivo de Escrita 2

Arquivo de Escrita 2

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

int main()
{
    FILE *fp;
    char ch;

    fp = fopen ( "c:/simao/simao2007/teste.txt", "w" );

    if ( fp == NULL )
    {
        printf ( "O arquivo não pode ser aberto. \n" );
        system ( "Pause" );
        exit (1);
    }

    printf ( "Programa para gravar caracteres em um arquivo chamado teste.txt." );
    printf ( "\n \n" );

    printf ( "Digite os caracteres: \n" );

    do
    {
        ch = getchar();
        fputc ( ch, fp );
    } while ( ch != '\n' );

    fclose ( fp );

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

Arquivo de Escrita 3

Arquivo de Escrita 3

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

```
int main()
{
    FILE *fp;
    char ch;
    char nome[100];

    printf ( "Qual é o nome do arquivo! \n" );
    gets ( nome );

    fp = fopen ( nome, "w" );

    if ( fp == NULL )
    {
        printf ( "O arquivo não pode ser aberto. \n" );
        system ("Pause");
        exit (1);
    }

    printf ( "\n \n" );

    printf ("Digite os caracteres: \n" );
    do
    {
        ch = getchar();
        fputc ( ch, fp );
    } while ( ch != '\n' );

    fclose ( fp );

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

Arquivo de Escrita 4

Utilizando *argc* e *argv*.

Arquivo de Escrita 4

```
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char *argv[])
{
    FILE *fp;
    char ch;

    if ( argc != 2 )
    {
        printf ( "Você esqueceu de digitar o nome do arquivo! \n" );
        system ( "Pause" );
        exit ( 1 );
    }

    fp = fopen ( argv[1], "w" );

    if ( fp == NULL )
    {
        printf ( "O arquivo não pode ser aberto. \n" );
        system ( "Pause" );
        exit ( 1 );
    }

    printf ( "\n \n" );
    printf ( "Digite os caracteres: \n" );
    do
    {
        ch = getchar();
        fputc ( ch, fp );
    } while (ch != '\n');

    fclose ( fp );

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

```
C:\SIMAO>dir e*.*
Volume in drive C has no label.
Volume Serial Number is FC94-EF9D

Directory of C:\SIMAO

14/06/2007  14:32           20.873 ExemploArquivoEscrita1.exe
14/06/2007  14:56           20.873 ExemploArquivoEscrita4.exe
             2 File(s)           41.746 bytes
             0 Dir(s)  19.071.557.632 bytes free

C:\SIMAO>ExemploArquivoEscrita4 TesteConsole.txt

Digite os caracteres:
Exemplo de utilizacao de Argc e Argv. Bem por isto que se chama o programa a par
tir do console do DOS!
Press any key to continue . . .

C:\SIMAO>dir Tes*.*
Volume in drive C has no label.
Volume Serial Number is FC94-EF9D

Directory of C:\SIMAO

14/06/2007  14:42             6 teste.txt
14/06/2007  14:58            41 TesteC.txt
10/10/2007  09:50           104 TesteConsole.txt
             3 File(s)           151 bytes
             0 Dir(s)  19.071.561.728 bytes free

C:\SIMAO>type TesteConsole.txt
Exemplo de utilizacao de Argc e Argv. Bem por isto que se chama o programa a par
tir do console do DOS!

C:\SIMAO>
```

Arquivo de Escrita 5

fputs (caracter, arquivo);

Arquivo de Escrita 5

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

int main()
{
    char str [180];

    FILE *fp;

    if ( ( fp = fopen ( "TEST2.TXT", "w" ) ) == NULL )
    {
        printf ( "O arquivo nao pode ser aberto. \n" );
        system ("Pause");
        exit (1);
    }

    printf ( "Digite uma ou mais linhas (linha em branco para sair) \n" );

    do
    {
        gets (str);
        strcat (str, "\n");
        fputs (str, fp);

    }while ( *str != '\n' );        // ( str[0] != '\n' )

    return 0;
}
```

Arquivo de Escrita 5

```
fgets ( str, 179, arquivo);
```

Arquivo de Escrita e Leitura

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char str [180];
    FILE *fp;

    if ( ( fp = fopen ("TEST2.TXT", "w+") ) == NULL )
    {
        printf ( "O arquivo nao pode ser aberto. \n" );
        system ( "Pause" );
        exit (1);
    }

    // Gravar arquivo
    printf ( "Digite uma ou mais linhas (linha em branco para sair) \n" );

    do
    {
        gets (str);
        strcat ( str, "\n");
        fputs ( str, fp );
    } while ( *str != '\n' );

    // Ler arquivo
    rewind ( fp );

    while ( !feof ( fp ) )
    {
        fgets ( str, 179, fp );
        puts ( str );
    }
    fclose ( fp );
    system ( "Pause" );
    return 0;
}
```

Modos de utilização de arquivos	
Modo	Significado
r	Abre um arquivo texto para leitura.
w	Cria um arquivo texto para escrita.
a	Anexa a um arquivo-texto.
rb	Abre um arquivo binário para leitura.
wb	Cria um arquivo binário para escrita.
ab	Anexa um arquivo binário.
r+	Abre um arquivo-texto para leitura/escrita.
w+	Cria um arquivo-texto para leitura/escrita.
a+	Anexa ou cria um arquivo-texto para leitura/escrita.
r+b	Abre um arquivo binário para leitura/escrita.
w+b	Cria um arquivo binário para leitura/escrita.
a+b	Anexa a um arquivo binário para leitura/escrita.

Exercício

Lista Postal

```
/* Um programa de lista postal muito simples */
```

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

```
#define TAM 2
```

```
struct Elemento
```

```
{  
  
    char nome [100];  
    char rua [100];  
    char cidade [100];  
    char estado [2];  
    char cep [10];  
  
};
```

```
struct Elemento Lista [TAM];
```

```
char menu ();
```

```
void inicia_lista ();
```

```
void cadastra ();
```

```
void mostra ();
```

```
void salva ();
```

```
void carrega ();
```

```
int main()
```

```
{  
    char escolha;  
    inicia_lista();  
  
    for ( ;; )  
    {  
        escolha = menu();  
        switch (escolha)  
        {  
            case 'c':  
            case 'C': { cadastra(); } break;  
  
            case 'm':  
            case 'M': { mostra(); } break;  
  
            case 's':  
            case 'S': { salva(); } break;  
  
            case 'a':  
            case 'A': { carrega(); } break;  
  
            case 't':  
            case 'T': { exit (0 ); } break;  
  
            default : { printf ( "Opcao invalida. \n" ); }  
  
            printf ( "\n \n \n" );  
        }  
        system ( "Pause" );  
        return 0;  
    }  
}
```

```
/* Um programa de lista postal muito simples */
```

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

```
#define TAM 2
```

```
struct Elemento
```

```
{  
  
    char nome [100];  
    char rua [100];  
    char cidade [100];  
    char estado [2];  
    char cep [10];
```

```
} Lista [TAM];
```

```
char menu ();
```

```
void inicia_lista ();
```

```
void cadastra ();
```

```
void mostra ();
```

```
void salva ();
```

```
void carrega ();
```

```
int main()  
{  
    char escolha;  
    inicia_lista();  
  
    for ( ;; )  
    {  
        escolha = menu();  
        switch (escolha)  
        {  
            case 'c':  
            case 'C': { cadastra(); } break;  
  
            case 'm':  
            case 'M': { mostra(); } break;  
  
            case 's':  
            case 'S': { salva(); } break;  
  
            case 'a':  
            case 'A': { carrega(); } break;  
  
            case 't':  
            case 'T': { exit (0 ); } break;  
  
            default : { printf ( "Opcao invalida. \n" ); }  
  
            printf ( "\n \n \n" );  
        }  
        system ( "Pause" );  
        return 0;  
    }  
}
```

```

char menu()
{
    printf ("\n \n \n");
    char opcao;

    printf ( " (C)adastrar.  \n" );
    printf ( " (M)ostrar.   \n" );
    printf ( " C(A)arregar. \n" );
    printf ( " (S)alvar.     \n" );
    printf ( " (T)erminar.  \n" );

    fflush ( stdin );
    scanf ( "%c", &opcao );

    return opcao;
}

```

```

void mostra()
{
    printf ("\n \n \n");

    register int t;

    for( t = 0; t < TAM; t++ )
    {
        if (Lista[t].nome[0] != '\0')
        {
            printf ( "%s \n", Lista[t].nome);
            printf ( "%s \n", Lista[t].rua);
            printf ( "%s \n", Lista[t].cidade);
            printf ( "%s \n", Lista[t].estado);
            printf ( "%s \n", Lista[t].cep);
        }
        printf ("\n");
    }
}

```

```

void inicia_lista()
{
    register int t;
    for (t = 0; t < TAM; t++)
    {
        *Lista[t].nome = '\0'; // Lista[t].nome[0] = '\0';
    }
}

```

```

void cadastra ()
{
    printf ("\n \n \n");

    for ( int i = 0; i < TAM; i++ )
    {
        printf ( "Nome: \n" );
        fflush ( stdin );
        gets ( Lista[i].nome );

        printf ( " Rua: \n" );
        fflush ( stdin );
        gets ( Lista[i].rua );

        printf ( "Cidade: \n" );
        fflush ( stdin );
        gets ( Lista[i].cidade );

        printf ( "Estado: \n" );
        fflush ( stdin );
        gets ( Lista[i].estado );

        printf ( "CEP: \n" );
        fflush ( stdin );
        gets ( Lista[i].cep );
    }
}

```

```
void salva ()
{
    printf ("\n \n \n");

    // Implementar usando archivos de texto.

}
```

```
void carrega ()
{

    // implementar usando archivos de texto.

}
```

Exercício 1

- 1.1. Refazer o projeto com ao menos um arquivo do tipo *.h* e dois arquivos do tipo *.c*.
- 1.2 Transformar o vetor *Lista* em variável não global, a ser declarada na função *main* e passada por parâmetro para as demais funções. Não usar nenhuma variável global enfim.
- 1.3 Implementar as *funções* *salva* e *recupera* usando arquivos de texto.
- 1.4 Adicionar na estrutura *Elemento* um campo ou variável inteira chamada *num*. Isto feito, fazer os tratamentos apropriados nas demais funções para considerar esta variável.
 - Dica, pesquisar função de conversão de inteiro para caractere.

Exercício 2

- 2.1 Modificar o exercício anterior de maneira tal que exista um estrutura (struct) chamada *Data*, com três campos inteiros chamados *dia*, *mes* e *ano*.
- 2.2 Fazer com que a estrutura *Elemento* do exercício anterior tenha um vetor *Vet_Datas* de estrutura *Data* de tamanho quatro. A primeira posição armazena a data de nascimento, a segunda posição a data de casamento ou equivalente, a terceira posição a data de divórcio ou equivalente (-1 para todos os campos caso não haja ocorrido) e a quarta posição a data de falecimento (idem).
- 2.3 Readequar as funções *cadastra*, *mostra*, *salva* e *recupera* do exercício anterior levando em conta esse novo contexto.