



Professor
Maromo

LINGUAGEM DE PROGRAMAÇÃO

Material 009



GitHub
maromo71

C



Agenda



Arquivos:

- **Arquivos textos**

Material: LP_010



Arquivos

Em C, um arquivo é um conjunto de bytes colocados um após o outro de forma sequencial.



Utilização

- 1) Fonte de dados para o programa (entrada- input).
- 2) Destino de dados gerados pelo programa (saída- output)



Tipos de periféricos

Dados de entrada podem ser fornecidos por periféricos de entrada: teclado, mouse, leitor de código de barras.

Dados de saída podem ser enviados por diversos periféricos de saída: tela do computador, impressora, etc.



Streams

Conjunto sequencial de caracteres, sem qualquer estrutura interna. Os programas ou comandos vêem apenas um conjunto de caracteres. Mesmo os caracteres especiais.

Sempre quando se falar em arquivos está se pensando em streams, que corresponde a conjunto de dados existentes em suporte magnético (disco, pen-drive, disquetes, etc).



Operações Básicas sobre Arquivos

- Abertura / Fechamento de um arquivo
- Argumentos na linha de comando para processar arquivos.
- Arquivos textos e binários.
- Detecção de fim de arquivo

Abertura de um arquivo

A abertura permite associar um arquivo existente em disco a uma variável do nosso programa. C possui apenas os tipos **char**, **int**, **float** e **double**, os quais não se referem a arquivos.

Para podermos utilizar um arquivo tem-se que declarar uma variável do tipo FILE (ou melhor um ponteiro para FILE).

A declaração de variáveis para o processamento de arquivos faz-se usando o tipo FILE definido no arquivo <stdio.h> [FILE] deve ser escrito em maiúsculo.



Declaração

```
int x,y;  
float k;  
FILE *fp; /* fp - file pointer */
```



Abertura de um arquivo

A abertura de um arquivo é realizada pela função `fopen`, cujo protótipo se encontra no arquivo `<stdio.h>`, pois se trata de uma operação-padrão de entrada/saída.
Sintaxe:

```
FILE * fopen(const char *filename, const char *mode)
```

Nota: a função recebe dois parâmetros.

filename: String contendo o nome físico do arquivo (Ex: dados.dat).

mode: String contendo o Modo de abertura do arquivo.



Nome do arquivo

É armazenado numa string e deve representar fielmente e na totalidade o nome do arquivo tal como é visto pelo sistema operacional que está se operando.

EX: C:\Temp\Dados.dat

Escrevemos em C, "**C:\\Temp\\Dados.dat**" pois "\\" é um caracter especial em C.



Modos de Abertura

r (read) abre um arquivo existente para leitura.

w (write) abre um arquivo para escrita. Se o arquivo não existir, o sistema operacional tentará criá-lo. Se o arquivo existir, todo o seu conteúdo será substituído pelo novo conteúdo.

a (append) abre um arquivo para inserção. Se o arquivo não existir, o sistema operacional tentará criá-lo.

r+ abre um arquivo existente para leitura e gravação.

w+ abre um arquivo para leitura e escrita. Se o arquivo não existir, o sistema operacional tentará criá-lo. Se o arquivo existir, todo o seu conteúdo será substituído pelo novo conteúdo.

a+ abre um arquivo para inserção e leitura. Se o arquivo não existir, o sistema operacional tentará criá-lo.

Exemplo arq1.c

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

int main(){
    FILE *arquivo = fopen("c:\\testes.txt", "r");

    // testa se o arquivo foi aberto com sucesso
    if(arquivo != NULL)
        printf("Arquivo foi aberto com sucesso.");
    else
        printf("Nao foi possível abrir o arquivo.");

    printf("\n");
    system("PAUSE");
    return 0;
}
```

OBS: troque pelo caminho onde o arquivo esteja gravado.



Modo Texto / Modo Binário

Por padrão, a abertura de um arquivo é realizada considerando este arquivo como sendo do **tipo texto**.

Para se abrir um arquivo no modo binário é preciso acrescentar um **"b"** ao modo de abertura apresentado anteriormente.

Ex: wb, rb, a+b, etc.



Descrição do arquivo Texto

Contituído de caracteres perceptíveis pelos humanos. Ex: letras do alfabeto, números, caracteres vulgares de nossa escrita &%^*.,, e ainda pelos separadores espaço em branco, tab e NewLine.



Descrição do arquivo binário

Podem ser constituídos por quaisquer caracteres existentes na tabela ASCII.



Fechamento de um arquivo

O Fechamento de um arquivo retira a ligação entre a nossa variável e o arquivo existente no disco. Antes de o arquivo ser fechado são gravados, fisicamente todos os dados que possam ainda existir em buffers associados ao arquivo. É liberada a memória alocada pela função fopen.

Sintaxe:

```
int fclose(FILE *arq)
```



Exemplo: `arq02.c`

O próximo programa indica se podemos ou não abrir um determinado arquivo.

Exemplo arq02.c

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

int main(){
    FILE *fp;
    char s[100];
    printf("Digite o nome do arquivo: ");
    gets(s);

    /* Abrir o arquivo */
    fp = fopen(s,"r");

    /* Verificar se a abertura foi efetuada com sucesso */
    if(fp==NULL)
        printf("Nao foi possivel abrir arquivo\n\n");
    else{
        printf("Arquivo %s aberto com sucesso\n\n",s);
        fclose(fp); /*Fechamos o arquivo */
    }
    system("PAUSE");
    return 0;
}
```



Leitura de Caracteres

Dentre as diversas funções existentes para leitura de caracteres. A mais usada é:

```
int fgetc(FILE *arq)
```

Essa função lê um caractere no arquivo passado como parâmetro (previamente aberto pela função **fopen**).



Arquivo `arq03.c`

Neste código poderemos implementar o comando `type` do DOS passando o nome do arquivo na linha de parâmetro.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    FILE *fp;
    int ch; /*Inteiro para ler os caracteres */
    /* Testar a linha de comando */
    if(argc!=2){
        printf("Sintaxe: \n\n%s Arquivo \n\n", argv[0]);
        system("PAUSE");
        exit(1); /* Termina o programa */
    }
    /* Abrir o Arquivo com o nome em argv[1] */
    fp = fopen(argv[1], "r");
    if(fp==NULL){
        printf("Impossível abrir o arquivo %s\n", argv[1]);
        system("PAUSE");
        exit(2);
    }
    while ((ch=fgetc(fp))!=EOF)
        putchar(ch);
    fclose(fp);
    return 0;
}
```




Explicando

Os parâmetros referem-se a:

- 1 – Nome do executável
- 2 – Nome do arquivo a ser aberto.



Ainda sobre o exemplo anterior

O comando:

```
while((ch=fgetc(fp))!=EOF)  
    putchar(ch);
```

Serve para leitura dos caracteres um a um, e coloca-os na tela por meio da função `putchar(ch)`.

Se o resultado for EOF (fim de arquivo) interrompe-se o laço.



Escrita de Caracteres num Arquivo

A escrita de caracteres em arquivo pode ser realizada pela função:

```
int fputc(int ch, FILE *arq)
```

Essa função escreve o caractere `ch` no arquivo `arq`. Essa função devolve o caractere `ch`, em caso de sucesso, ou `EOF`, no caso contrário.



Problema

Implementar um utilitário que permita copiar um arquivo para outro arquivo.


```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    FILE *fin, *fout; /* fin para entrada, fout para saída */
    int ch; /*Inteiro para ler os caracteres */
    /* Testar a linha de comando */
    if(argc!=3) {
        printf("Sintaxe: \n\n%s Origem Destino \n\n", argv[0]);
        exit(1); /* Termina o programa */
    }
    /* Abrir o Arquivo argv[1] (Origem) */
    fin = fopen(argv[1], "rb");

    if(fin==NULL) {
        printf("Impossível abrir o arquivo %s\n", argv[1]);
        exit(2);
    }

    /* Abrir o Arquivo argv[2] (destino) */
    if ((fout = fopen(argv[2], "wb"))==NULL) {
        printf("Impossível criar o arquivo %s\n", argv[2]);
        exit(3);
    }
    while ((ch=fgetc(fin))!=EOF) fputc(ch, fout);

    fclose(fin);
    fclose(fout);
    return 0;
}
```



Explicando

O comando do bloco abaixo:

```
while ((ch=fgetc(fin))!=EOF) fputc(ch, fout);
```

O laço é exatamente igual ao do exercício anterior, colocando o caractere lido no arquivo informado em destino.



Entradas e Saídas Formatadas

É possível ler ou escrever dados de forma formatada em arquivos através das funções ***fscanf*** e ***fprintf***. Essas funcionam da mesma forma que as funções ***scanf*** e ***printf***, tendo apenas mais um parâmetro inicial que corresponde ao arquivo onde o processamento irá ser realizado.



Entrada e Saídas Formatadas

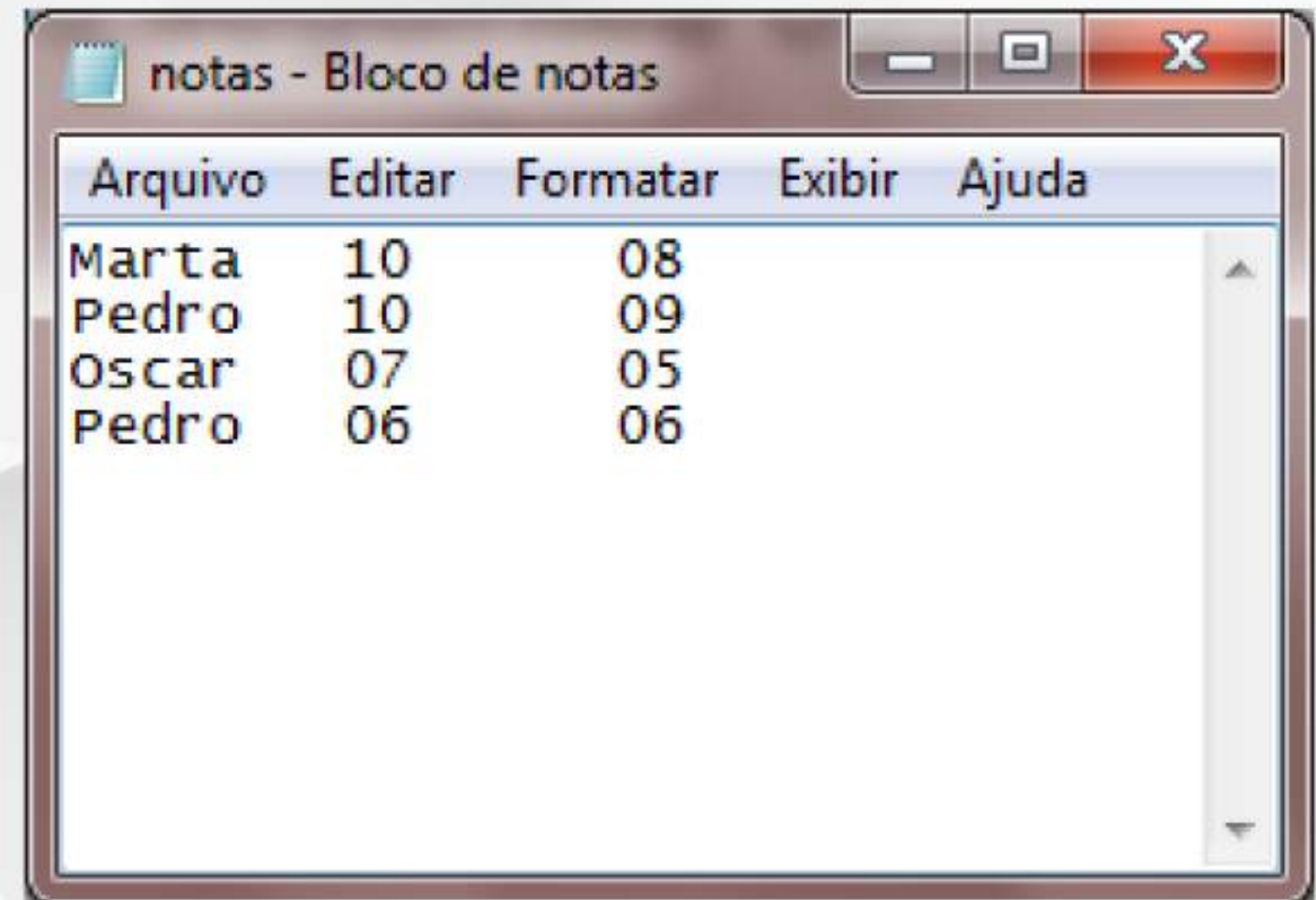
```
int fscanf(FILE *arq, const char *format, ...)
```

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


Problema

Você tem um arquivo sequencial formato txt com os seguintes dados.

1. Crie um programa que calcule e mostre média dos alunos.
2. O programa deve informar se o aluno está aprovado ou reprovado de acordo com o seguinte critério:
aprovado: média ≥ 7 , caso contrário: **reprovado**



Arquivo	Editar	Formatar	Exibir	Ajuda
Marta	10	08		
Pedro	10	09		
Oscar	07	05		
Pedro	06	06		

Arquivo arq05.c

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

int main(int argc, char *argv[])
{
    FILE *fp;
    char Nome[50];
    float nota1, nota2, media;

    /* Testar a linha de comando */
    if(argc!=2){
        printf("Sintaxe: \n\n%s Arquivo\n\n", argv[0]);
        exit(1);
    }
}
```



Arquivo: arq05.c

```
/* Abrir o Arquivo */
if((fp = fopen(argv[1], "r"))==NULL){
    printf("Impossivel abrir arquivo %d\n", argv[1]);
    exit(2);
}

/* Teste da nota */
while (fscanf(fp, "%s %f %f", Nome, &nota1, &nota2)!=EOF) {
    media= (nota1 + nota2)/2;
    if(media>=7)
        printf("%s sua media %f situacao=aprovado\n", Nome, media);
    else
        printf("%s sua media %f situacao=reprovado\n", Nome, media);
}
fclose(fp);
return 0;
}
```

Referências

.....

DAMAS, L. M. D. Linguagem C. LTC, 2007.

.....

HERBERT, S. C completo e total. 3a. ed. Pearson, 1997.

.....

