



# Programação Estruturada

---

## Aula 03 - Arquivos

Bacharelado em Sistemas de Informação  
Prof.º Philippe Leal  
[philippeleal@yahoo.com.br](mailto:philippeleal@yahoo.com.br)



# Definição

---

- **Arquivo** é o mecanismo utilizado quando o programa trabalha com uma grande quantidade de dados de entrada e/ou saída, sendo inviável utilizar apenas teclado e tela.
- **Arquivo de entrada:** o programa faz a leitura de dados do arquivo.
- **Arquivo de saída:** o programa escreve (salva) dados no arquivo.



# Tipos de Arquivo

---

- Na maioria dos sistemas operacionais um arquivo pode ser visto de duas maneiras:

- **Modo texto**

O Arquivo é visto como um texto composto por uma sequência de caracteres;

- **Modo binário**

O Arquivo é visto como uma sequência de bytes.



# Como Utilizar um Arquivo

---

- Em linhas gerais para usar um arquivo temos que executar os seguintes passos:
  - 1.** Abrir o arquivo (para leitura e/ou escrita);
  - 2.** Usar (ler e/ou escrever) o arquivo;
  - 3.** Fechar o arquivo.
- As funções para manipulação de arquivos encontram-se na biblioteca **<stdio.h>**.



# Abrindo um Arquivo

---

- Para abrir um arquivo utiliza-se a função **fopen( )**:
  - Esta função recebe dois parâmetros:  
o **nome do arquivo** e o **modo de abertura**;
  - Esta função retorna um ponteiro para o tipo **FILE**;
  - Um ponteiro é um tipo de variável que serve para armazenar endereços de memória;
  - Por hora, basta saber que para declarar uma variável ponteiro adiciona-se um asterisco antes do nome da variável.

**Ex.:** FILE \*arq;



# Abrindo um Arquivo

---

- Modos de abertura:
  - “r”: abre um arquivo para leitura. O arquivo deve existir ou um erro ocorre;
  - “w”: cria um arquivo vazio para escrita. Caso um arquivo com o mesmo nome exista, o seu conteúdo é apagado;
  - “a”: adiciona ao final do arquivo. O arquivo é criado caso ele não exista.
- Existem outros modos de abertura, mas inicialmente vamos nos limitar a esses.



# Abrindo um Arquivo

---

- Exemplo:

- Declaração da variável ponteiro para arquivo:

**FILE \*arqEntrada, \*arqSaida;**

- Se fosse abrir o arquivo chamado "Entrada.txt" para leitura:

**arqEntrada = fopen("Entrada.txt", "r");**

- Se fosse abrir o arquivo chamado "Saida.txt" para escrita:

**arqSaida = fopen("Saida.txt", "w");**



# Abrindo um Arquivo

---

- Exemplo:

- Se der algum erro na abertura do arquivo, a função **fopen( )** retorna a constante simbólica pré-definida **NULL**:

...

```
arqEntrada = fopen("Entrada.txt", "r");
```

```
if (arqEntrada == NULL){
```

```
    printf("\n Erro na abertura do arquivo! \n");
```

```
    exit(1); /* Aborta o programa */
```

```
}
```

...





# Fechando um Arquivo

---

- O valor retornado pela função **fopen( )** deve ser passado para as demais funções que manipulam o arquivo;
- Ao final, o arquivo deve ser fechado usando-se a função **fclose( )**:

...

```
arqEntrada = fopen("Entrada.txt", "r");
```

...

```
/* Código de manipulação do arquivo */
```

...

```
fclose(arqEntrada);
```

...



# Fim de Arquivo

---

- A função **feof( )** é usada para detectar o fim de arquivo;
- Recebe como parâmetro um ponteiro para arquivo e retorna não-zero se já chegou o final do arquivo ou zero caso contrário:

**Ex.:** while( ! feof(arqEntrada) ) {  
    ...  
}



# Voltando ao Início de um Arquivo

---

- A função **rewind( )** permite recolocar o cursor no início do arquivo.
- Recebe como parâmetro um ponteiro de arquivo e não retorna nada.

**Ex.:** `rewind(arqEntrada);`



# Lendo de um Arquivo

---

- A principal função para ler de um arquivo é **fscanf( )**;
- Análoga a função scanf( ), porém recebe um ponteiro de arquivo como parâmetro:

...

```
int a;
```

```
float b;
```

```
FILE *arqEntrada;
```

```
arqEntrada = fopen("Entrada.txt", "r");
```

...

```
fscanf(arqEntrada, "%d %f", &a, &b); /* Espera ler um valor inteiro e um real */
```

...

```
fclose(arqEntrada);
```



# Escrevendo em um Arquivo

---

- A principal função para escrever em um arquivo é **fprintf( );**
- Análoga a função printf( ), porém recebe um ponteiro de arquivo como parâmetro:

...

```
int a = 10;
```

```
float b = 5.0;
```

```
FILE *arqSaida;
```

```
arqSaida = fopen("Saida.txt", "w");
```

...

```
fprintf(arqSaida, "Valores: a = %d e b = %.2f \n", a, b);
```

...

```
fclose(arqSaida);
```



# Exemplo 01

---

- Calcular a média das notas dos alunos a partir do arquivo "**Notas.txt**" abaixo, onde a primeira linha contém o número de notas e, depois, uma nota por linha.

Notas.txt

6
7.5
8.4
9.1
4.2
5.5
4.8

# Resposta Exemplo 01

```
1 #include<stdio.h>
2 |
3 int main(){
4
5     int i, n;
6     float nota, soma = 0.0;
7     FILE *arqEntrada;
8
9     /* Abertura do arquivo para leitura */
10    arqEntrada = fopen("Notas.txt", "r");
11
12    if(arqEntrada == NULL){
13        printf("\nErro na abertura do arquivo! \n\n");
14        return 1; /* Aborta o programa */
15    }
16
17    /* Leitura da quantidade de notas no arquivo */
18    fscanf(arqEntrada, "%d", &n);
19
20    /* Loop para leitura de cada nota */
21    for(i = 1; i <= n; i++){
22        fscanf(arqEntrada, "%f", &nota);
23        soma = soma + nota;
24    }
25
26    /* Fechamento do arquivo */
27    fclose(arqEntrada);
28
29    /* Impressao do valor da media na tela */
30    printf("\nMedia = %.2f \n\n", soma/n);
31
32    return 0;
33 }
```



# Exemplo 02

---

- Calcular a média das notas dos alunos a partir do arquivo “**Notas.txt**” abaixo. Semelhante ao Exemplo 01, porém agora não sabemos a quantidade de notas no arquivo.

Notas.txt

7.5
8.4
9.1
4.2
5.5
4.8



# Resposta Exemplo 02

```
1 #include<stdio.h>
2 |
3 int main(){
4
5     int cont = 0;
6     float nota, soma = 0.0;
7     FILE *arqEntrada;
8
9     /* Abertura do arquivo para leitura */
10    arqEntrada = fopen("Notas.txt", "r");
11
12    if(arqEntrada == NULL){
13        printf("\nErro na abertura do arquivo! \n\n");
14        return 1; /* Aborta o programa */
15    }
16
17
18    /* Loop para leitura de cada nota */
19    fscanf(arqEntrada, "%f", &nota);
20    while(!feof(arqEntrada)){
21
22        soma = soma + nota;
23        cont++; //Conta quantas notas sao lidas
24        fscanf(arqEntrada, "%f", &nota);
25
26    }
27
28    /* Fechamento do arquivo */
29    fclose(arqEntrada);
30
31    /* Impressao do valor da media na tela */
32    printf("\nMedia = %.2f \n\n", soma/cont);
33
34    return 0;
35 }
```



# Exemplo 03

---

- Ler as notas dos alunos a partir do arquivo "**Notas.txt**" abaixo e imprimir a média das notas no arquivo "**Media.txt**".

Notas.txt

7.5
8.4
9.1
4.2
5.5
4.8

# Resposta Exemplo 03

```
1 #include<stdio.h>
2 |
3 int main(){
4
5     int cont = 0;
6     float nota, soma = 0.0;
7     FILE *arqEntrada, *arqSaida;
8
9     /* Abertura do arquivo para leitura */
10    arqEntrada = fopen("Notas.txt", "r");
11
12    if(arqEntrada == NULL){
13        printf("\nErro na abertura do arquivo! \n\n");
14        return 1; /* Aborta o programa */
15    }
16
17
18    /* Loop para leitura de cada nota */
19    fscanf(arqEntrada, "%f", &nota);
20    while(!feof(arqEntrada)){
21
22        soma = soma + nota;
23        cont++; //Conta quantas notas sao lidas
24        fscanf(arqEntrada, "%f", &nota);
25
26    }
27
28    /* Fechamento do arquivo */
29    fclose(arqEntrada);
30
31    /* Abertura do arquivo para escrita */
32    arqSaida = fopen("Media.txt", "w");
33
34    /* Impressao do valor da media no arquivo "Media.txt" */
35    fprintf(arqSaida, "\nMedia = %.2f \n\n", soma/cont);
36
37    /* Fechamento do arquivo */
38    fclose(arqSaida);
39
40    return 0;
41 }
```



# Exercício

---

- Escreva um programa que leia o arquivo "**Notas.txt**" contendo as notas dos alunos (uma nota por linha) e determine:
  - a) A nota máxima e nota mínima;
  - b) O número de alunos aprovados, dado que "Aprovado" significa nota maior ou igual a 5.0.