

MCTA028-15: Programação Estruturada



Aula 10: Arquivos

Wagner Tanaka Botelho

wagner.tanaka@ufabc.edu.br / wagtanaka@gmail.com

Universidade Federal do ABC (UFABC)

Centro de Matemática, Computação e Cognição (CMCC)

Introdução

Introdução

- Por que usar **arquivos**?
 - Os **dados** permanecem **disponíveis** para uso, mesmo que o **programa** que o gerou já tenha sido **encerrado**;
 - Permitem armazenar **grande** quantidade de **informação**;
 - Acesso **concorrente** aos dados, ou seja, **mais** de um **programa** pode utilizá-lo ao mesmo tempo;
 - ...
- A **Linguagem C** possui **funções**:
 - **CRIAR** e **ESCREVER** em novos arquivos;
 - **LER** o seu **conteúdo**.

Introdução

- A **Linguagem C** trabalha com apenas **DOIS** tipos de arquivos:
 - **Texto:**
 - Será considerado nesta aula.
 - **Binários:**
 - Arquivos **executáveis**, **compactados**, de **registros**, etc.

Abrir/Fechar um Arquivo

Abrir um Arquivo

```
FILE *fopen(char *nome_do_arquivo, char *modo)
```

➤ **nome_do_arquivo:**

➤ *String* contendo o **nome do arquivo** que deverá ser **ABERTO**.

➤ **modo:**

➤ *String* contendo o **MODO** de **ABERTURA DO ARQUIVO**.

➤ **Retorna:**

➤ **NULL**: caso ocorra algum **ERRO**;

➤ **Ponteiro** para o **arquivo aberto**.

Abrir um Arquivo: Modo

- O **MODO DE ABERTURA** do arquivo determina que **TIPO** de uso será feito nele;
- Os seguintes **modos de abertura** de um arquivo são válidos:

Modo	Arquivo	Função
"r"	Texto	Leitura. Arquivo deve existir.
"w"	Texto	Escrita. Cria arquivo, se não houver. Apaga o anterior, se ele existir.
"a"	Texto	Escrita. Os dados serão adicionados no fim do arquivo.
"r+"	Texto	Leitura/escrita. O arquivo deve existir e pode ser modificado.
"w+"	Texto	Leitura/escrita. Cria arquivo se não houver. Apaga o anterior se ele existir.
"a+"	Texto	Leitura/escrita. Os dados serão adicionados no fim do arquivo.

```
FILE *fopen(char *nome_do_arquivo, char *modo)
```



```
FILE *fp;  
fp = fopen("C:\\Projetos\\NovoProjeto\\arquivo.txt", "w");
```

Um detalhe, se o **arquivo txt** estiver na **MESMA** pasta do **arquivo .c**, basta:

```
FILE *fp;  
fp = fopen("arquivo.txt", "w");
```


Fechar um Arquivo

➤ Sempre que **TERMINAMOS** de usar um **arquivo**, devemos **FECHÁ-LO**:

➤ Deve-se usar a função **fclose()**:

```
int fclose(FILE *fp)
```

➤ **RECEBE** como **parâmetro** o **ponteiro fp** que determina o **arquivo** a ser **fechado**;

➤ Como **resultado**, a função retorna:

➤ Um **valor inteiro** igual a **ZERO**:

➤ No caso de **SUCESSO** no **fechamento do arquivo**.

➤ Um **valor DIFERENTE** de **ZERO**:

➤ Houve **ERRO** no fechamento.

*Ex_05.c X

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void main() {
5      FILE *fp;
6      //fp = fopen("D:\\UFABC\\Disciplinas\\2021-2025\\Q1\\PE\\Aulas\\10\\Codigos\\Aula10\\Teste.txt", "rw");
7      fp = fopen("Teste.txt", "r");
8
9      if (fp == NULL) {
10         printf("Erro na abertura do arquivo.\n");
11     }
12     else {
13         fclose(fp);
14     }
15 }
```

Nome

- bin
- obj
- Aula10
- Aula10.depend
- Aula10.layout
- Ex_01
- Ex_02
- Ex_02x
- Ex_03
- Ex_03x
- Ex_04
- Ex_04x
- Ex_05
- Ex_05xx
- Teste

Aula10

Sources

- Ex_01.c
- Ex_02.c
- Ex_02x.c
- Ex_03.c
- Ex_03x.c
- Ex_04.c
- Ex_04x.c
- Ex_05.c
- Ex_05xx.c

Others

Teste.txt

Escrita e Leitura

Escrita e Leitura

- A **Linguagem C** possui várias **funções** prontas para a **manipulação de arquivos**;
- Deve-se utilizar a **BIBLIOTECA PADRÃO** de **entrada** e **saída** (**stdio.h**);
- As **operações** de **ESCRITA** e **LEITURA** em arquivos se limitam a **operações**:
 - Abrir/fechar;
 - Ler/escrever.
- Todas as **funções** de **manipulação de arquivos** trabalham com o conceito de **ponteiro de arquivos**:
 - **Ponteiro** aponta para o primeiro registro no arquivo;
 - Indica quando chegamos ao **FINAL** de um arquivo.

Escrevendo um Caractere

- Para **ESCREVER** um **caractere** em um arquivo, deve-se usar a função `fputc()`:

```
int fputc(int c, FILE *fp);
```

- **c**: caractere a ser **ESCRITO** no arquivo;
- **fp**: variável que está associada ao arquivo onde o caractere será escrito;
- **RETORNA**:
 - **CONSTANTE** *End Of File* (**EOF**), em geral, -1, se houver **ERRO** na **escrita**;
 - O próprio **caractere**, se ele foi escrito com **SUCESSO**.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  void main() {
5      FILE *arq;
6      char string[100];
7      int i;
8
9      arq = fopen("Teste.txt", "w");
10
11     if (arq == NULL) {
12         printf("Erro na abertura do arquivo");
13         exit(1);
14     }
15
16     printf("Entre com a string a ser gravada no arquivo:\n");
17     gets(string);
18
19     //Grava a string, caractere a caractere
20     for (i = 0; i < strlen(string); i++) {
21         fputc(string[i], arq);
22     }
23     fclose(arq);
24 }
```

D:\UFABC\Disciplinas\2021-2025\Q1\PE\Aulas\10\Codigos'

Entre com a string a ser gravada no arquivo:
Teste!!!!

Teste.txt x

Teste!!!!

Lendo um Caractere

➤ A função `fgetc ()` faz a **leitura de caracteres**:

```
int fgetc(FILE *fp);
```

➤ **Recebe** como **parâmetro de entrada** apenas a variável associada ao arquivo de onde o caractere será lido;

➤ **RETORNA:**

➤ Constante **EOF** (em geral, -1), se houver **ERRO** na **LEITURA**;

➤ O **caractere** lido do arquivo, na forma de um valor inteiro, se ele foi lido com sucesso.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void main() {
4      FILE *arq;
5      char c;
6
7      arq = fopen("teste.txt", "r");
8
9      if(arq == NULL) {
10         printf("Erro na abertura do arquivo");
11         exit(1);
12     }
13
14     for(int i = 0; i < 5; i++) {
15         c = fgetc(arq);
16         printf("%c", c);
17     }
18     fclose(arq);
19 }
```

D:\UFABC\Disciplinas\2021-2025

UFABC
Process returned 0 (0x0)
Press any key to continue.

Teste.txt X

UFABC

Lendo cinco caracteres de um arquivo.


```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void main() {
4      FILE *arq;
5      char c;
6
7      arq = fopen("teste.txt", "r");
8
9      if(arq == NULL) {
10         printf("Erro na abertura do arquivo");
11         exit(1);
12     }
13
14     while((c = fgetc(arq)) != EOF) {
15         printf("%c", c);
16     }
17
18     fclose(arq);
19 }
```

D:\UFABC\Disciplinas\2021-2025

UFABC
Process returned 0 (0x0)
Press any key to continue.

Teste.txt X

UFABC

`fgetc()` é utilizada juntamente com a constante **EOF** para ler não apenas alguns caracteres, mas para continuar lendo caracteres enquanto não encontrar o final do arquivo.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void main() {
4      FILE *fp;
5      char c;
6
7      fp = fopen("Teste.txt", "r");
8
9      if (fp == NULL) {
10         printf("Erro na abertura do arquivo\n");
11         exit(1);
12     }
13
14     while (!feof(fp)) {
15         c = fgetc(fp);
16         printf("%c", c);
17     }
18     fclose(fp);
19 }
```

`feof()` retorna um valor inteiro **IGUAL** a **ZERO** se ainda **NÃO** tiver atingido o **final do arquivo**. Um valor de retorno **DIFERENTE** de **ZERO** significa que o final do arquivo já foi atingido.

Teste.txt X

```
1  UFABC
2  Santo Andre
3  Sao Bernardo do Campo
4
```

UFABC

Santo Andre

Sao Bernardo do Campo

Process returned 0 (0x0)

Press any key to continue.

Referências

- Slides do Prof. Luiz Rozante;
- SALES, André Barros de; AMVAME-NZE, Georges. Linguagem C: roteiro de experimentos para aulas práticas. 2016;
- BACKES, André. Linguagem C Completa e Descomplicada. Editora Campus. 2013;
- SCHILDT, Herbert. C Completo e Total. Makron Books. 1996;
- DAMAS, Luís. Linguagem C. LTC Editora. 1999;
- DEITEL, Paul e DEITEL, Harvey. C Como Programar. Pearson. 2011;
- BACKES, André. Estrutura de Dados Descomplicada em Linguagem C. GEN LTC. 2016.