



## ENSINO MÉDIO INTEGRADO - INFORMÁTICA

### Disciplina de Linguagem de Programação

Aula 15: Arquivo Texto

---

*Gil Eduardo de Andrade*

#### Conceitos Preliminares

<https://docs.microsoft.com/pt-br/cpp/c-language/?view=msvc-170>

### Introdução

Na linguagem C um arquivo pode ser utilizado para representar vários recursos, desde um arquivo de texto armazenado em disco até uma terminal de comandos ou impressora. Toda vez que um programa em C inicia uma comunicação (abre) com um arquivo, um fluxo de dados (stream) com o mesmo é criado.

### Definição

Um arquivo de texto é um recurso computacional estruturado em linhas que permite o armazenamento de dados. O arquivo de texto é normalmente utilizado para armazenar um ou mais caracteres, contendo também caracteres especiais, como o marcador de fim de arquivo (EOF);

### Ponteiro de Arquivo - Funcionalidade

Um ponteiro de arquivo aponta para as informações que definem o arquivo apontado, tais como: nome, status e posição atual. O ponteiro possibilita identificar um arquivo específico dentro de um disco rígido, permitindo que operações sejam efetuadas (leitura, escrita, etc.). O ponteiro para arquivo é uma variável do tipo FILE.

### Sintaxe

**FILE \*fp;**

- **FILE \*:** indica o tipo do ponteiro que está sendo criado;

- **fp**: nome da variável ponteiro que está sendo declarada;

## Abrindo um Arquivo

É chamado de “abertura de arquivo” o processo que permite criar um canal de comunicação (fluxo de dados - stream) com um arquivo em específico. Apenas a partir da abertura é possível obter informações sobre o mesmo e também executar operações de escrita e leitura. A função em C que permite abrir uma arquivo chama-se ***fopen()*** (*file open*);

## Sintaxe

➤ Protótipo:

**FILE\*** **fopen**(const char \***nome**, const char\* **modo**);

- **nome**: é um nome válido para um arquivo que pode ou não incluir seu caminho.
- **modo**: determina como o arquivo será aberto, as opções são mostradas a seguir.

## Modos de Abertura

| MODO | DESCRIÇÃO   |
|------|---|
| r    | Abre um arquivo texto já existente para leitura.  |
| w    | Abre/Cria um arquivo texto para escrita apagando seu conteúdo   |
| r+   | Abre um arquivo texto já existente para leitura e escrita.  |
| w+   | Abre/Cria um arquivo texto para escrita e leitura apagando seu conteúdo   |
| a+   | Abre/Cria um arquivo texto para escrita e leitura mantendo seu conteúdo – ponteiro é colocado no início do arquivo para leitura e no final para escrita |

## Lendo um Arquivo Texto

Após um arquivo texto ser aberto é possível efetuar operações como leitura e escrita. A leitura permite que o conjunto de caracteres armazenados no arquivo seja lido pela aplicação C. A função mais elementar que permite ler um arquivo texto, caractere a caractere, é a ***fgetc()***.

### Sintaxe

➤ Protótipo:

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

- **fp**: é o ponteiro que referencia o arquivo aberto durante a chamada da função *fopen()*;
- **int**: a função *fgetc()* retorna o código ascii (int) dos caracteres que estão sendo lidos;
- **EOF**: último caractere a ser lido, indica que o final do arquivo foi alcançado (end of file);

A linguagem C também possui uma função mais elaborada que permite ler um conjunto de caracteres de uma só vez. Essa função chama-se ***fgets()***. A função ***fgets()*** lê um conjunto de caracteres para um dado arquivo até que o caractere de nova linha '\n' seja encontrado ou que a quantidade de caracteres especificado (que deseja-se obter) tenha sido alcançada.

### Sintaxe

➤ Protótipo:

```
FILE* fgets(char* str, int quantidade, FILE* fp);
```

- **str**: variável ***string*** que receberá o conjunto de caracteres que será lido do arquivo;
- **quantidade**: determina a quantidade de caracteres que deverá ser lida do arquivo;
- **fp**: é o ponteiro que referencia o arquivo aberto durante a chamada da função *fopen()*;

## Escrevendo num Arquivo Texto

A escrita permite que um conjunto de caracteres seja armazenado no arquivo aberto pela aplicação C. A função mais elementar que permite escrever, caractere a caractere, em um arquivo texto é a *fputc()*.

### Sintaxe

➤ Protótipo:

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

- **fp**: é o ponteiro que referencia o arquivo aberto durante a chamada da função *fopen()* ;
- **int**: a função *fputc()* retorna o código ascii (int) do caractere que foi escrito ou EOF se a escrita não for bem sucedida;
- **c**: caractere que será inserido no arquivo;

A linguagem C também possui uma função mais elaborada que permite escrever um conjunto de caracteres de uma só vez. Essa função chama-se *fputs()*. A função *fputs()* escreve um conjunto de caracteres especificado para um dado arquivo.

### Sintaxe

➤ Protótipo:

```
char* fputs(const char* str, FILE* fp);
```

- **str**: variável *string* que será escrita no arquivo texto;
- **fp**: é o ponteiro que referencia o arquivo aberto durante a chamada da função *fopen()* ;

## Removendo/Excluindo um Arquivo Texto

### Sintaxe

➤ Protótipo:

```
int remove(const char* nome);
```

- **int:** a função `remove()` retorna o valor "0" se o arquivo foi removido com sucesso, ou "outro valor" se não foi possível remover o arquivo;
- **nome:** nome (com caminho) do arquivo que será removido;

## Movendo Ponteiro para o Início do Arquivo

### Sintaxe

➤ Protótipo:

```
void rewind(FILE* fp);
```

- **fp:** ponteiro para o arquivo que foi aberto anteriormente e que deseja-se mover novamente ao início, permitindo efetuar a leitura novamente;



## Codificação – Linguagem de Programação C

---

### Abrir Arquivo para Leitura

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

int main() {
    FILE *fp;

    fp = fopen("arquivo.txt", "r");

    if(fp == NULL) {
        printf("Não foi possível abrir o arquivo!\n");
        return 0;
    }
}
```

```
printf("Arquivo aberto com sucesso!");  
  
fclose(fp);  
  
printf("\n");  
  
return 0;  
  
}
```

Função *fopen()* - Modo de Abertura "r"

---

(Arquivo-fonte: 17 - Arquivo Texto/abrir\_arquivo\_r.c)

## Abrir/Criar Arquivo para Escrita (apaga conteúdo)

```
#include <stdio.h>  
  
#include <stdio_ext.h>  
  
#include <stdlib.h>  
  
int main() {  
  
    FILE *fp;  
  
    fp = fopen("arquivo.txt", "w");  
  
    if(fp == NULL) {  
  
        printf("Não foi possível abrir o arquivo!\n");  
  
        return 0;  
  
    }  
  
    printf("Arquivo aberto/criado com sucesso!");  
  
    fclose(fp);  
  
    printf("\n");  
  
    return 0;  
  
}
```

Função *fopen()* - Modo de Abertura "w"

(Arquivo-fonte: 17 - Arquivo Texto/abrir\_arquivo\_w.c)

## Abrir/Criar Arquivo para Escrita (mantém conteúdo)

```
#include <stdio.h>

#include <stdio_ext.h>

#include <stdlib.h>

int main() {

    FILE *fp;

    fp = fopen("arquivo.txt", "a+");

    if(fp == NULL) {

        printf("Não foi possível abrir o arquivo!\n");

        return 0;

    }

    printf("Arquivo aberto/criado com sucesso!");

    fclose(fp);

    printf("\n");

    return 0;

}
```

Função *fopen()* - Modo de Aberto "a+"

(Arquivo-fonte: 17 - Arquivo Texto/abrir\_arquivo\_a+.c)

## Lendo Todo o Arquivo - *fgetc()*

```
#include <stdio.h>

#include <stdio_ext.h>

#include <stdlib.h>

int main() {
```

```
FILE *fp;

char c;

fp = fopen("arquivo.txt", "a+");

if(fp == NULL) {

    printf("Não foi possível abrir o arquivo!\n");

    return 0;

}

do {

    c = fgetc(fp);

    if(c != EOF) {

        printf("%c", c);

    }

} while(c != EOF);

fclose(fp);

printf("\n");

return 0;

}
```

Função *fgetc()* - Lendo arquivo até encontrar o caractere EOF.

---

(Arquivo-fonte: 17 - Arquivo Texto/ler\_todo\_arquivo\_fgetc.c)

## Lendo uma Linha Específica do Arquivo - *fgetc()*

```
#include <stdio.h>

#include <stdio_ext.h>

#include <stdlib.h>

int main() {

    FILE *fp;

    char c;
```

```
int linha, cont;

fp = fopen("arquivo.txt", "a+");

if(fp == NULL) {

    printf("Não foi possível abrir o arquivo!\n");

    return 0;

}

printf("Linha: ");

scanf("%i", &linha);

cont = 1;

do {

    c = fgetc(fp);

    if(c == '\n' || c == EOF) {

        cont++;

    }

    else if(cont == linha) {

        printf("%c", c);

    }

} while(c != EOF);

fclose(fp);

printf("\n");

return 0;

}
```

Função *fgetc()* - Lendo uma linha especificada pelo usuário.

---

(Arquivo-fonte: 17 - Arquivo Texto/ler\_linha\_arquivo\_fgetc.c)

## Lendo uma String do Arquivo - *fgets()*

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

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

int main() {

    FILE *fp;

    char str[10];

    fp = fopen("arquivo.txt", "a+");

    if(fp == NULL) {

        printf("Não foi possível abrir o arquivo!\n");

        return 0;

    }

    fgets(str, 10, fp);

    printf("Caracteres lidos: %s", str);

    fclose(fp);

    printf("\n");

    return 0;

}
```

Função *fgets()* - Lendo uma quantidade específica de caracteres.

---

(Arquivo-fonte: 17 - Arquivo Texto/ler\_string\_arquivo\_fgets.c)

## Lendo Todo o Arquivo - *fgets()*

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

int main() {

    FILE *fp;

    char str[100];

    fp = fopen("arquivo.txt", "a+");
```

```
if(fp == NULL) {  
    printf("Não foi possível abrir o arquivo!\n");  
    return 0;  
}  
  
while(fgets(str, 100, fp) != NULL) {  
    printf("%s", str);  
}  
  
fclose(fp);  
  
printf("\n");  
  
return 0;  
}
```

Função *fgets()* - Lendo arquivo até encontrar NULL.

---

(Arquivo-fonte: 17 - Arquivo Texto/ler\_todo\_arquivo\_fgets.c)

## Escrevendo uma String no Arquivo - *fputc()*

```
#include <stdio.h>  
  
#include <stdio_ext.h>  
  
#include <stdlib.h>  
  
int main() {  
  
    FILE *fp;  
  
    char str[100];  
  
    int a;  
  
    fp = fopen("nomes.txt", "a+");  
  
    if(fp == NULL) {  
  
        printf("Não foi possível abrir o arquivo!\n");  
  
        return 0;  
  
    }  
}
```

```
printf("Digite um nome: ");  
  
gets(str);  
  
for(a=0; str[a] != '\0'; a++) {  
    fputc(str[a], fp);  
}  
  
fputc('\n', fp);  
  
fclose(fp);  
  
printf("\n");  
  
return 0;  
}
```

Função *fputc()* - Escrevendo uma string no arquivo.

---

(Arquivo-fonte: 17 - Arquivo Texto/escrever\_string\_arquivo\_fputc.c)

## Escrevendo uma String no Arquivo - *fputs()*

```
#include <stdio.h>  
  
#include <stdio_ext.h>  
  
#include <stdlib.h>  
  
int main() {  
  
    FILE *fp;  
  
    char str[100];  
  
    int a;  
  
    fp = fopen("nomes.txt", "a+");  
  
    if(fp == NULL) {  
  
        printf("Não foi possível abrir o arquivo!\n");  
  
        return 0;  
  
    }  
  
    printf("Digite um nome: ");
```

```
gets(str);  
  
fputs(str, fp);  
  
putc('\n', fp);  
  
fclose(fp);  
  
printf("\n");  
  
return 0;  
}
```

Função *fputs()* - Escrevendo uma string no arquivo.

---

(Arquivo-fonte: 17 - Arquivo Texto/escrever\_string\_arquivo\_fputs.c)