

# Arquivos



adrierj@gmail.com (não compartilhado) [Alternar conta](#)



**\*Obrigatório**

## Definição

### Em C

Um arquivo é uma sequência de Bytes colocados uns após os outros de forma sequencial.

É também uma fonte de dados, servindo como entrada e saída de dados.

C processa os bytes por meio de streams (conjunto sequencial de caracteres ou bytes sem qualquer estrutura interna)

Cada stream está ligada a um arquivo.

Padrão stream:

- stdin - entrada do teclado
- stdout - saída na tela

Operações básicas:

- Abertura
- Utilização
- Fechamento



## Funções de manipulação de arquivos

Função	Ação
<b>fopen()</b>	Abre um arquivo
<b>Fclose()</b>	Fecha um arquivo
<b>putc()</b> e <b>fputc()</b>	Escreve um caractere em um arquivo
<b>getc()</b> e <b>fgetc()</b>	Lê um caractere de um arquivo
<b>fseek()</b>	Posiciona em um registro de um arquivo
<b>fprintf()</b>	Efetua impressão formatada em um arquivo
<b>fscanf()</b>	Efetua leitura formatada em um arquivo
<b>feof()</b>	Verifica o final de um arquivo
<b>fwrite()</b>	Escreve tipos maiores que 1 byte em um arquivo
<b>fread()</b>	Lê tipos maiores que 1 byte de um arquivo

## &lt;stdio.h&gt;

Para abrir um arquivo, declaramos uma variável FILE (stdio.h).

FILE \*fp //um ponteiro

Exemplo:

```
fp = fopen ("texto.txt","w"); //cria arquivo
```



## Modos de abertura

Modo	Significado
"r"	Abre um arquivo texto para leitura. O arquivo deve existir antes de ser aberto.
"w"	Abrir um arquivo texto para gravação. Se o arquivo não existir, ele será criado. Se já existir, o conteúdo anterior será destruído.
"a"	Abrir um arquivo texto para gravação. Os dados serão adicionados no fim do arquivo ("append"), se ele já existir, ou um novo arquivo será criado, no caso de arquivo não existente anteriormente.
"rb"	Abre um arquivo binário para leitura. Igual ao modo "r" anterior, só que o arquivo é binário.
"wb"	Cria um arquivo binário para escrita, como no modo "w" anterior, só que o arquivo é binário.
"ab"	Acrescenta dados binários no fim do arquivo, como no modo "a" anterior, só que o arquivo é binário.
"r+"	Abre um arquivo texto para leitura e gravação. O arquivo deve existir e pode ser modificado.
"w+"	Cria um arquivo texto para leitura e gravação. Se o arquivo existir, o conteúdo anterior será destruído. Se não existir, será criado.
"a+"	Abre um arquivo texto para gravação e leitura. Os dados serão adicionados no fim do arquivo se ele já existir, ou um novo arquivo será criado, no caso de arquivo não existente anteriormente.
"r+b"	Abre um arquivo binário para leitura e escrita. O mesmo que "r+" acima, só que o arquivo é binário.
"w+b"	Cria um arquivo binário para leitura e escrita. O mesmo que "w+" acima, só que o arquivo é binário.
"a+b"	Acrescenta dados ou cria um arquivo binário para leitura e escrita. O mesmo que "a+" acima, só que o arquivo é binário.

## Exemplo

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    // criando a variável ponteiro para o arquivo
    FILE *pont_arq;

    //abrindo o arquivo
    pont_arq = fopen("arquivo.txt", "a");

    // fechando arquivo
    fclose(pont_arq);

    //mensagem para o usuário
    printf("O arquivo foi criado com sucesso!");

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

## Fechando arquivo

Utilizamos a função `fclose()`, que escreve no disco qualquer dado que ainda possa existir em buffer, e o fecha no sistema operacional.

sintaxe:

```
int fclose(FILE * fp); //retorno zero se o fechamento ocorrer sem problemas.
```



## Escrevendo

Essa operação sobrescreve o conteúdo já existente no arquivo. Primeiro é necessário abrir o arquivo em modo que permita a escrita, em seguida gravar o texto.

A função `fprintf` armazena dados em um arquivo. Seu funcionamento é muito semelhante ao `printf`, a diferença principal é a existência de um parâmetro para informar o arquivo onde os dados serão armazenados.

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    FILE *pont_arq; // cria variável ponteiro para o arquivo
    char palavra[20]; // variável do tipo string

    //abrindo o arquivo com tipo de abertura w
    pont_arq = fopen("arquivo_palavra.txt", "w");

    //testando se o arquivo foi realmente criado
    if(pont_arq == NULL)
    {
        printf("Erro na abertura do arquivo!");
        return 1;
    }

    printf("Escreva uma palavra para testar gravacao de arquivo: ");
    scanf("%s", palavra);

    //usando fprintf para armazenar a string no arquivo
    fprintf(pont_arq, "%s", palavra);

    //usando fclose para fechar o arquivo
    fclose(pont_arq);

    printf("Dados gravados com sucesso!");

    return(0);
}
```



### Exemplo: exibindo informações

Leitura de strings – Função fgets()

É utilizada para leitura de strings em um arquivo. Realiza a leitura dos caracteres até o final da linha quando encontra o caracter \n. A leitura é efetuada de tal forma que a string lida é armazenada em um ponteiro do tipo char. A função pode ser finalizada quando encontrar o final do arquivo, neste caso retorna o endereço da string lida. Se ocorrer algum erro na leitura do arquivo, a função retorna NULL.

Adicione ao código anterior:

```
char texto_str[20];

//abrindo o arquivo_frase em modo "somente leitura"
pont_arq = fopen("arquivo_palavra.txt", "r");

//enquanto não for fim de arquivo o looping será executado
//e será impresso o texto
while(fgets(texto_str, 20, pont_arq) != NULL)
printf("%s", texto_str);

//fechando o arquivo
fclose(pont_arq);
```

### Escrevendo, mas mantendo o que existia

Operação de anexação:

append

Primeiro abrimos um arquivo em modo que permita adicionar conteúdo, depois gravamos o texto.

Exemplo:

```
pont_arq = fopen("arquivo_palavra.txt", "a");
```

Implemente os exemplos até aqui demonstrados \*

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários

