

Linguagem e Técnicas de Programação



Manipulação de Arquivos em Linguagem C

Nós iremos aprender como realizar a manipulação de arquivos em Linguagem C, incluindo a **abertura, leitura, escrita** e fechamento de arquivos.

Vamos explorar três exemplos práticos para ilustrar esses conceitos.

Arquivos

Mas antes, veja:

- Em Linguagem C, a manipulação de arquivos é feita usando ponteiros para arquivos (**FILE pointers**).
- Para trabalhar com arquivos, você precisa incluir a biblioteca **stdio.h**.

Arquivos

Abrindo e Fechando Arquivos

Para abrir um arquivo, você pode usar a função **fopen()**. Ela retorna um ponteiro para o arquivo aberto.

```
1  
2  
3     FILE *arquivo;  
4     arquivo = fopen("gostosinho.txt", "r");  
5  
6
```

// Abre o arquivo "gostosinho.txt" para leitura

- Use a função **fclose()** para fechar um arquivo quando terminar de usá-lo.



```
fclose(arquivo);
```

Arquivos

Modos de Leitura:

- **"r"**: Modo de leitura (abre o arquivo para leitura). O arquivo deve existir previamente, caso contrário, a função retornará NULL.
- **"rb"**: Modo de leitura binária (para ler arquivos binários). Semelhante a "r", mas para arquivos binários.

Arquivos

Modos de Escrita:

- **"w"**: Modo de escrita (abre o arquivo para escrita). Se o arquivo já existir, seu conteúdo será truncado (apagado).
- **"wb"**: Modo de escrita binária (para escrever arquivos binários). Semelhante a "w", mas para arquivos binários.
- **"a"**: Modo de anexação (abre o arquivo para escrita, mas não apaga seu conteúdo). Os novos dados serão adicionados ao final do arquivo.
- **"ab"**: Modo de anexação binária (para anexar a um arquivo binário). Semelhante a "a", mas para arquivos binários.

Modos de Leitura e Escrita:

- **"r+"**: Modo de leitura e escrita (abre o arquivo para leitura e escrita). O arquivo deve existir previamente.
- **"rb+"**: Modo de leitura e escrita binária (para leitura e escrita em arquivos binários). Semelhante a "r+", mas para arquivos binários.
- **"w+"**: Modo de leitura e escrita (abre o arquivo para leitura e escrita). Se o arquivo já existir, seu conteúdo será truncado.

- **"wb+":** Modo de leitura e escrita binária (para leitura e escrita em arquivos binários). Semelhante a "w+", mas para arquivos binários.
- **"a+":** Modo de leitura e anexação (abre o arquivo para leitura e escrita, anexando no final). Os novos dados podem ser escritos ao final do arquivo existente.

Outros Modos Especiais:

- **"t"**: Texto (modo padrão). Quando não especificado, o modo de texto é usado.
- **"b"**: Binário. Indica que o arquivo está no modo binário. Por exemplo, "rb" indica leitura de arquivo binário.

Leitura de Arquivos

Para ler dados de um arquivo, use as funções **fscanf()** ou **fgets()**.

Arquivos

```
1
2 FILE *arquivo;
3 arquivo = fopen("gostosinho.txt", "r");
4 if (arquivo == NULL) {
5     printf("O arquivo não foi tão gostosinho assim. Não abriu! :/");
6     return 1;
7 }
8
9 char linha[100];
10 while (fgets(linha, sizeof(linha), arquivo) != NULL) {
11     printf("%s", linha);
12 }
13
14 fclose(arquivo);
15
16
```

Função fgets deve receber 3 parâmetros obrigatórios

Ex.: **fgets(variavel, qtdCaracteres, file)**

Escrita em Arquivos

Para escrever em um arquivo, use as funções **fprintf()** ou **fputs()**.

Arquivos

```
2 FILE *arquivo;
3 arquivo = fopen("saida.txt", "w");
4 if (arquivo == NULL) {
5     printf("Erro ao abrir o arquivo.\n");
6     return 1;
7 }
8
9 fprintf(arquivo, "Olá, mundo!\n");
10 fclose(arquivo);
```

Função **fgets** deve receber 3 parâmetros obrigatórios:

Ex.: **fgets(variavel, qtdCaracteres, file)**

Exemplos:

1. Crie um script em C, capaz de realizar a Leitura de um arquivo e contar a palavras.

Arquivos



Exemplos:

1. Crie um script em C, capaz de realizar a Leitura de um arquivo e contar a palavras.

RESPOSTA

```
1  #include <stdio.h>
2
3  int main() {
4      FILE *arquivo;
5      arquivo = fopen("texto.txt", "r");
6      if (arquivo == NULL) {
7          printf("Erro ao abrir o arquivo.\n");
8          return 1;
9      }
10
11     char palavra[100];
12     int contagem = 0;
13     while (fscanf(arquivo, "%s", palavra) != EOF) {
14         contagem++;
15     }
16
17     printf("O arquivo contém %d palavras.\n", contagem);
18     fclose(arquivo);
19
20     return 0;
21 }
```

*EOF – End of File

Exemplos:

2. Crie um script em C que crie e escreva a frase: "O professor é um gostosinho!" em um arquivo.

Arquivos



Exemplos:

2. Crie um script em C que crie e escreva a frase: "O professor é um gostosinho!" em um arquivo.

RESPOSTA

```
1  #include <stdio.h>
2
3  int main() {
4      FILE *arquivo;
5      arquivo = fopen("saida.txt", "w");
6      if (arquivo == NULL) {
7          printf("Erro ao abrir o arquivo.\n");
8          return 1;
9      }
10
11     fprintf(arquivo, "O professor é um gostosinho!\n");
12     fclose(arquivo);
13
14     return 0;
15 }
```

Exemplos:

3. Crie um script que copie o texto de um arquivo (origem.txt) para outro (destino.txt).

Arquivos



Exemplos:

3. Crie um script que copie o texto de um arquivo (origem.txt) para outro (destino.txt).

RESPOSTA

```
1  #include <stdio.h>
2
3  int main() {
4      FILE *arquivo_origem, *arquivo_destino;
5      arquivo_origem = fopen("origem.txt", "r");
6      arquivo_destino = fopen("destino.txt", "w");
7      if (arquivo_origem == NULL || arquivo_destino == NULL) {
8          printf("Erro ao abrir os arquivos.\n");
9          return 1;
10     }
11
12     char caractere;
13     while ((caractere = fgetc(arquivo_origem)) != EOF) {
14         fputc(caractere, arquivo_destino);
15     }
16
17     fclose(arquivo_origem);
18     fclose(arquivo_destino);
19
20     printf("Arquivo copiado com sucesso.\n");
21     return 0;
22 }
```