



Arquivos - Programação II

Autor Prof.: Delcino Picinin Júnior

Manipulação de Arquivo

As operações de entrada e saída do C, incluindo as relacionadas a arquivos, encontram-se na biblioteca `stdio.h`. Essa biblioteca também define o EOF, que indica o fim de arquivo. Além disso, é nela que está definido o tipo `FILE`.

As principais funções relacionadas a manipulação de arquivos existentes na biblioteca `stdio.h` são:

- **fopen()** - Abre um arquivo.
- **fclose()** - Fecha o arquivo garantindo a transferência do buffer.
- **fflush()** - Descarrega o buffer.
- **fscanf()** - Leitura de entrada formatada (semelhante ao `scanf()`).
- **fprintf()** - Escrita de saída formatada (semelhante ao `printf()`).
- **fgets()** - Obtém uma string do arquivo.
- **fgetc()** - Obtém um caracter do arquivo.
- **fputs()** - Insere uma string no arquivo.
- **fputc()** - Insere um caracter no arquivo.
- **fread()** - Lê um bloco de dados do arquivo.
- **fwrite()** - Escreve um bloco de dados no arquivo.
- **fseek()** - Reposiciona o ponteiro.
- **rewind()** - Reposiciona o ponteiro para o início do arquivo.
- **ftell()** - Retorna a posição do ponteiro.

Abertura de Arquivo

O primeiro passo para trabalhar com um arquivo é fazer a associação do arquivo físico com um ponteiro. Para isso utilizamos o tipo FILE. A abertura/associação do arquivo é feita pela função `fopen(const char* arquivo, const char* modo)`. O modo que a abertura é feita podem ser:

- "r- Abre o arquivo somente para leitura, a partir do início. O arquivo deve existir.
- "w- Cria um arquivo vazio para escrita. Se já havia o arquivo, ele é perdido.
- "a- Adiciona no final do arquivo. Se o arquivo não existir, a função o cria.
- "r+- Abre o arquivo para leitura e escrita, a partir do início. O arquivo deve existir.
- "w+" Cria um arquivo vazio para leitura e escrita. Se já havia o arquivo, ele é perdido.
- "a+" Abre para adição ou leitura no final do arquivo. Se o arquivo não existir, a função o cria.

Código 1: Exemplo de abertura com a+

```
1 #include <stdio.h>
2 void main(int argc, char *argv[]) {
3     FILE *fp;
4     char nomePessoa[10];
5     if ((fp=fopen("file.txt", "a+")) != NULL) {
6         printf("Digite seu nome ");
7         scanf("%s", nomePessoa);
8         fprintf(fp, "O nome digitado é %s\n", nomePessoa);
9         fclose(fp);
10    } else {
11        printf("Problema na abertura do arquivo");
12    }
13 }
```

Fprintf e Fscanf

A função fprintf() funciona como a função printf(). A diferença é que a saída de fprintf() é um arquivo e não a tela do computador.

Código 2: Exemplo escrita com fprintf

```
1 #include <stdio.h>
2 void main(int argc, char *argv[]) {
3     FILE *fp;
4     int op;
5     char nomePessoa[10];
6     if ((fp=fopen("file.txt", "a+")) != NULL) {
7         printf("\n Quer cadastrar 1—sim 2—não");
8         scanf("%d",&op);
9         while(op==1){
10             printf("Digite seu nome ");
11             scanf("%s",nomePessoa);
12             fprintf(fp, "O nome digitado é %s\n",nomePessoa);
13             printf("\n Quer cadastrar novamente 1—sim 2—não");
14             scanf("%d",&op);
15         }
16         fclose(fp);
17     } else {
18         printf("Problema na abertura do arquivo");
19     }
20 }
```

A função fscanf() funciona como a função scanf(). A diferença é que fscanf() lê de um arquivo e não do teclado do computador.

Código 3: Exemplo leitura com fscanf

```
1 #include <stdio.h>
2
3 void main(int argc, char *argv[]) {
4     FILE *fp;
5     char t1[10], t2[10], t3[10], t4[10];
6     char nomePessoa[10];
7     if ((fp=fopen("file.txt", "a+")) != NULL) {
8         while (!feof(fp)){
9             fscanf(fp, "%s %s %s %s %s",t1,t2,t3,t4,nomePessoa);
10             if (!feof(fp)){
11                 printf("\n %s %s %s %s %s",t1,t2,t3,t4,nomePessoa);
12             }
13         }
14         fclose(fp);
15     } else {
16         printf("Problema na abertura do arquivo");
17     }
18 }
```

Fread e Fwrite

Podemos escrever e ler blocos de dados. Para tanto, temos as funções fread() e fwrite().

Código 4: Exemplo leitura com fread

```
1 #include <stdio.h>
2
3 int main( int argc, char** argv ){
4     if( argc > 1 ){
5         FILE *fp;
6         char conteudo[200];
7         fp = fopen(argv[1], "r");
8         fread(&conteudo, sizeof(char), 100,fp);
9         printf("\nO CONTEÚDO DO ARQUIVO É:\n%s\n", conteudo);
10        fclose(fp);
11    }else{
12        printf("Informe o arquivo. Ex.: ./prog file .txt\n");
13    }
14    return(0);
15 }
```

Código 5: Exemplo leitura com fwrite

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main( int argc, char** argv ){
5     if( argc > 1 ){
6         FILE *fpr, *fpw;
7         char conteudo[200];
8         fpr = fopen(argv[1], "r");
9         fpw = fopen(argv[2], "w");
10        fread(&conteudo, sizeof(char), 200, fpr);
11
12        for( int x=0; x<strlen(conteudo); x++){
13            if( conteudo[x]>96){
14                conteudo[x]=conteudo[x]-32;
15            }
16        }
17        fwrite(&conteudo, sizeof(char), strlen(conteudo),fpw);
18        fclose(fpr);
19        fclose(fpw);
20    }else{
21        printf("Informe o arquivo. Ex.: ./prog file .txt\n");
22    }
23    return(0);
24 }
```