



FeMASS

FACULDADE PROF. MIGUEL ÂNGELO DA SILVA SANTOS

Arquivos

Ana Maria M. Moura

professora.anamoura@gmail.com

Arquivos - Agenda

- Declaração
- Abrindo arquivos
- Fechando arquivos
- Escrevendo em arquivos
- Lendo o conteúdo de arquivos
- Posicionando o ponteiro do arquivo
- Verificando final de arquivo
- Exemplos

Declaração

- Para criar/abrir um arquivo é necessário utilizar um ponteiro para arquivo.
 - FILE *arq;
- Biblioteca
 - stdio

Abrindo arquivos

- Para abrir um arquivo é preciso utilizar a função `fopen("caminho/nome.ext","modo")`.
 - `arq = fopen("alunos.txt","a+");`
- O primeiro parâmetro é o nome do arquivo com o caminho onde o arquivo deve ser ou está armazenado.
- O segundo parâmetro é o modo de abertura do arquivo. O modo determina as operações que poderemos fazer com o arquivo após a abertura.
 - Texto
 - `w` (cria o arquivo e permite escrita)
 - `r` (abre um arquivo existente e permite leitura)
 - `a` (abre um arquivo para edição – escrita e leitura)
 - `w+` (cria/sobrescreve o arquivo e permite escrita e leitura)
 - `r+` (abre um arquivo existente e permite leitura e escrita)
 - `a+` (abre / cria um arquivo para edição – escrita e leitura)
 - Binário
 - `wb` (cria o arquivo e permite escrita)
 - `rb` (abre um arquivo existente e permite leitura)
 - `Ab` (abre um arquivo para edição – escrita e leitura)
 - `wb+` (cria/sobrescreve o arquivo e permite escrita e leitura)
 - `rb+` (abre um arquivo existente e permite leitura e escrita)
 - `ab+` (abre / cria um arquivo para edição – escrita e leitura)

Fechando arquivos

- Para fechar um arquivo é preciso utilizar a função `fclose(ponteiro_arquivo)`
 - Ex.: `fclose(arq);`
- Se após fechar o arquivo, você desejar não usar mais o ponteiro é possível atribuir `NULL` ao ponteiro.
 - `Arq=NULL;`

Escrevendo em arquivos

- Para escrever em um arquivo, é possível utilizar as funções abaixo:
 - `fprintf(ponteiro_arquivo,"conteúdo"[,variavel]);`
 - Ex.: `fprintf(arq,"\\n%d",i);` - escreve o valor da variável como inteiro em uma nova linha.
 - `fputs(ponteiro_string, ponteiro_arquivo);`
 - Ex.: `fputs("olá mundo!", arq);` - escreve o conteúdo da variável como um texto no arquivo.
 - `fputc(ponteiro_char, ponteiro_arquivo);`
 - Ex.: `fputs('A', arq);` - escreve a letra A no arquivo.
 - `fwrite(ponteiro_conteúdo, tamanho_conteúdo, qtde_conteúdo, ponteiro_arquivo);`
 - Ex.: `fwrite(&a,sizeof(aluno),1,arq);` - escreve o conteúdo de uma variável aluno usando o tamanho da estrutura aluno no arquivo.

Escrevendo em arquivos

- Para escrever em um arquivo, é possível utilizar as funções abaixo:
 - `fscanf(ponteiro_arquivo,"caracter de controle",ponteiro_variavel);`
 - Ex.: `fscanf(arq,"%d",&i);` - lê o valor da variável como inteiro a partir de uma linha do arquivo.
 - `fgets(ponteiro_string, qytd_caracteres, ponteiro_arquivo);`
 - Ex.: `fgets(linha, 100, arq);` - lê uma linha do arquivo e coloca o conteúdo na variável linha que deve ser um vetor de char.
 - `fread(ponteiro_conteúdo, tamanho_conteúdo, qtde_conteúdo, ponteiro_arquivo);`
 - Ex.: `fread(&a,sizeof(aluno),1,arq);` - lê o conteúdo de um aluno de um arquivo e coloca o conteúdo na variável a do tipo aluno.

Posicionando o ponteiro do arquivo

- `rewind(ponteiro_arquivo)`
 - Posiciona o ponteiro no início do arquivo

Verificando final de arquivo

- `feof(ponteiro_arquivo)`
 - Retorna 0 enquanto não atingiu o final do arquivo;
 - Retorna um inteiro > 0 ao chegar no final do arquivo.

```
while(feof(arq)==0){  
    fgets(linha,100,arq);  
    printf("%s\n",linha);  
}
```

```
while(!feof(arq)){  
    fgets(linha,100,arq);  
    printf("%s\n",linha);  
}
```

Exemplos

- Neste exemplo, é criado um arquivo texto com números de 1 a 10, sendo um número em cada linha do arquivo.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
main(){
    setlocale(LC_ALL,"portuguese");
    int i;
    FILE *arq = fopen("numeros.txt","w+");
    if (arq==NULL){
        printf("O arquivo não pode ser criado.");
        exit(1);
    }
    for(i=1;i<=10;i++){
        fprintf(arq,"%d\n",i);
    }
    fclose(arq);
}
```

Exemplos

- Neste exemplo, é lido um arquivo com números inteiros sem saber o tamanho total do arquivo e os números são apresentados na tela.
- Quando o arquivo chega ao fim, a função fcanf retorna 0.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
main(){
    setlocale(LC_ALL,"portuguese");
    int i;
    FILE *arq = fopen("numeros.txt","r");
    if (arq==NULL){
        printf("O arquivo não pode ser localizado.");
        exit(1);
    }
    while(fscanf(arq,"%d",&i)!=0){
        printf("\n%d",i);
    }
    fclose(arq);
}
```

Exemplos

- Neste exemplo, o usuário informa os nomes e notas de 10 alunos. Os dados são gravados em uma estrutura e esta é escrita em um arquivo.
- Neste caso, o arquivo é binário e, por isso, foi aberto com o modo ab.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>
struct Aluno{
    char nome[50];
    float n1, n2;
};
main(){
    setlocale(LC_ALL, "portuguese");
    int i;
    Aluno a;
    FILE *arq = fopen("alunos.txt", "ab");
    if (arq==NULL){
        printf("O arquivo não pode ser localizado ou criado.");
        exit(1);
    }
    for(i=0; i<10; i++){
        printf("\nNome:"); fflush(stdin);
        gets(a.nome);
        printf("\nNota1: "); scanf("%f", &a.n1);
        printf("\nNota2: "); scanf("%f", &a.n2);
        fwrite(&a, sizeof(Aluno), 1, arq);
    }
    fclose(arq);
}
```

Exemplos

- Neste exemplo, um arquivo com estruturas é lido e as informações são mostradas para o usuário.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>
struct Aluno{
    char nome[50];
    float n1, n2;
};
main(){
    setlocale(LC_ALL,"portuguese");
    int i;
    Aluno a;
    FILE *arq = fopen("alunos.txt","ab+");
    if (arq==NULL){
        printf("O arquivo não pode ser localizado ou criado.");
        exit(1);
    }
    while(fread(&a,sizeof(Aluno),1,arq)!=0){
        printf("\nNome: %s ",a.nome);
        printf("\nNota1: %f ",a.n1);
        printf("\nNota2: %f ",a.n2); |
    }
    fclose(arq);
}
```