
Fundamentos de Programação

Aula 14 - Arquivos - Parte 2 (funções auxiliares)

Arnaldo Barreto Vila Nova

Sumário

- feof
 - rewind
 - fflush
 - fseek
 - remove
 - ferror
-

feof(FILE *stream)

- Retorna *true* se o ponteiro está no EOF (final do arquivo)
- Retorna *false* se não estiver

```
FILE *arquivo = fopen("texto.txt", "r");
char c;
printf("Conteúdo do arquivo:\n\n");
while(!feof(arquivo))
{
    c = fgetc(arquivo);
    printf("%c", c);
}
fclose(arquivo);
```

feof(FILE *stream)

- Retorna *true* se o ponteiro está no EOF (final do arquivo)
- Retorna *false* se não estiver

```
FILE *arquivo = fopen("texto.txt", "w");
char str[200];
printf("Escreva o texto: (finalize com EOF)\n\n");
while(!feof(stdin))
{
    scanf("%s",&str);
    fprintf(arquivo, "%s", str);
}
fclose(arquivo);
```

rewind(FILE *stream)

- Faz com que o stream do ponteiro volte ao início do arquivo

```
printf("Quer imprimir o conteúdo do arquivo? (s ou n)\n");
scanf("%c", &resp);
if(resp == 's')
{
    rewind(arquivo);
    while(!feof(arquivo))
    {
        printf("%c", fgetc(arquivo));
    }
}
```

fflush(FILE *stream)

- Atualiza o arquivo relacionado ao stream, mesmo sem o fechamento (fclose) do arquivo
- Não é muito aconselhável o uso (pode se tornar uma brecha de segurança do código)

```
while(!feof(stdin))
{
    printf("Digite um nome: (finalize com EOF)\n");
    gets(nome);
    fprintf(arquivo, "%s", nome);
    fflush(arquivo);
}
```

fseek(FILE *stream, long qt_bytes, int pos)

- Posiciona o stream em um local específico do arquivo
 - **qt_bytes** é a quantidade de bytes percorridos a partir da posição **pos** informada
 - temos 3 opções para **pos**:
 - **SEEK_SET** - movimenta qt_bytes a partir do início do arquivo
 - **SEEK_CUR** - movimenta qt_bytes a partir da posição atual do stream
 - **SEEK_END** - movimenta qt_bytes retrocedendo a partir do final do arquivo
-

fseek(FILE *stream, long qt_bytes, int pos)

- Exemplo:

```
FILE *arquivo = fopen("texto.txt", "w");
fputs("Dono do arquivo: Arnaldo", arquivo);
fflush(arquivo);
system("pause");
fseek(arquivo, 17, SEEK_SET);
fputs("Deadpool", arquivo);
fflush(arquivo);
system("pause");
fseek(arquivo, -8, SEEK_END);
fputs("Chimichangas", arquivo);
fclose(arquivo);
```

fseek(FILE *stream, long qt_bytes, int pos)

- Exemplo:

```
FILE *arquivo = fopen("pessoas.txt", "a");
struct pessoa p;
printf("Digite o nome e idade da pessoa a ser adicionada");
scanf("%s %d", &p.nome, &p.idade);
int pos;
printf("Quer guardar em qual posição do arquivo?");
scanf("%d", &pos);
fseek(arquivo, pos * sizeof(struct pessoa), SEEK_SET);
fwrite(&p, sizeof(struct pessoa), 1, arquivo);
fclose(arquivo);
```

remove(char *nome_arquivo)

- Deleta o arquivo (o arquivo tem que estar fechado)
- Retorna 0 (zero) se o arquivo for deletado com sucesso

```
printf("Você tem certeza que quer apagar o  
arquivo?\n");  
scanf("%c", &resp);  
if (resp == 's')  
{  
    if (remove("texto.txt") == 0)  
        printf("Arquivo deletado com sucesso");  
    else  
        printf("O arquivo não foi deletado");  
}
```

ferror(FILE *stream)

- Detecta se houve algum erro na última operação realizada com o stream;
- Retorna 0 (zero) se não houve erro

```
while(!feof(stdin))
{
    printf("Digite um nome: (finalize com EOF)\n");
    gets(nome);
    fprintf(arquivo, "%s", nome);
    if (ferror(arquivo))
        printf("\nHouve algum erro na operação fprintf");
    fflush(arquivo);
}
```
