Programação

Arquivos em C

Luiz Eduardo da Silva

Universidade Federal de Alfenas

Agenda

Agenda

Introdução

- O sistema de arquivo C é definido para trabalhar com uma série de dispositivos: **impressora**, **terminais**, **disco**, etc... Estes dispositivos são vistos como arquivos lógico em C denominados STREAM. (abstração do dispositivo).
- O dispositivo real é denominado ARQUIVO (impressora, disco, console). Um STREAM é associado a um ARQUIVO por uma operação de abertura do arquivo.
- Existem dois conjuntos de funções de E/S em C:
  - E/S ANSI (com buffer ou formatada) e a
  - E/S UNIX (sem buffer ou não formatada).
- Daremos uma ênfase maior ao primeiro conjunto pela portabilidade deste sistema de entrada e saída com arquivos.

Funções

Função	Descrição
fopen()	abre um arquivo (associa stream ao arquivo)
fclose()	fecha um arquivo
putc() e fputc()	escreve um caracter no arquivo
getc() e fgetc()	lê um caracter do arquivo
fseek ()	posiciona num registro do arquivo
fprintf()	escrita formatada no arquivo
fscanf()	leitura formatada no arquivo
fputs()	escrita de um string do arquivo
fgets()	leitura de um string do arquivo
feof()	verdadeiro se o fim do arquivo é alcançado
fwrite()	escrita de registros no arquivo
fread( )	leitura de registrso do arquivo

Ponteiro para arquivo

Ponteiro

A estrutura FILE, predefinida em stdio.h, contem informações sobre o arquivo, incluindo nome, status (aberto, fechado) e posição atual sobre o arquivo. Para se definir uma variável ponteiro de arquivo usa-se o seguinte comando:

```
FILE *arq;
```

Abrindo arquivos

fopen()

A função que abre arquivo em C é a função fopen (). Ela devolve NULL (nulo) ou um ponteiro associado ao arquivo e deve ser passado para função o nome físico do arquivo e o modo como este arquivo deve ser aberto. Exemplo:

```
arq = fopen ("teste", "w");
```

- O modo é definido pela combinação das letras r (read), w (write), a (append), b (binary), t (text) e + (leitura e escrita)

Modo	Significado
r	abre um arquivo texto (que já deve existir) para leitura
w	ou "wt"cria um arquivo texto para escrita (sobrescreve)
a	abre um arquivo texto (que já deve existir) para anexar
rb, wb ou ab	abre, cria ou anexa arquivo binário
r+, w+ ou a+	abre, cria ou anexa arquivo texto para leitura e escrita
r+b, w+b ou a+b	abre, cria ou anexa arquivo binário para leitura e escrita

Fechando arquivos

fclose()

A função que fecha e esvazia o buffer de um arquivo é a função fclose (). Exemplo:

```
fclose (arq);
```

```
1 #include <stdio.h>
2 int main () {
3     FILE *arq;
4     arq = fopen ("teste.txt", "rt");
5     if (arq == NULL)
6         puts ("Arquivo não existe");
7     else
8         puts ("Arquivo aberto");
9     fclose (arq);
10 }
```

Leitura e escrita

Leitura e escrita

- As funções fgetc, fputc, fscanf, fprintf, fgets e fputs tem um funcionamento similiar a leitura/escrita pelo teclado. A diferença é que tem um parâmetro a mais que especifica onde o dado será lido ou escrito.

```
1 #include <stdio.h>
2 int main() {
3   FILE *arq;
4   arq = fopen("texto.txt", "rt");
5   if (arq == NULL)
6     puts("Arquivo não existe");
7   do {
8     char c = fgetc(arq);
9     if (!feof(arq))
10      putchar(c);
11  } while (!feof(arq));
12  fclose(arq);
13 }
```

Leitura e escrita

Leitura e escrita

- As funções fread() e fwrite() servem para ler e escrever estruturas maiores. Os parâmetros dessas funções são:

```
fread (buffer, tamanho, quantidade, arquivo)
fwrite (buffer, tamanho, quantidade, arquivo)
```

- buffer- é um endereço na memória da estrutura para onde deve ser lido ou de onde serão escritos os valores (fread () e fwrite () respectivamente).
- tamanho - é um valor numérico que define o tamanho em bytes da estrutura que deve ser lida/escrita.
- quantidade - número de estrutura que serão lidas ou escritas em cada processo de fread ou fwrite.
- arquivo - é o ponteiro do arquivo onde será lida ou escrita uma estrutura.

fseek

protótipo

```
int fseek(FILE *arquivo, int deslocamento, int origem);
```

Esta função reposiciona o indicador de posição de fluxo (entrada e saída), [deslocamentos] posições a partir da [origem]. Mais indicada para arquivos binários.

Onde:

- arquivo - é a stream que terá o indicador de posição alterado
- deslocamento - diferença da posição relativa a origem
- origem - de onde partirá o deslocamento. Pode ser:
  - SEEK_SET - começo do arquivo
  - SEEK_CUR - posição atual
  - SEEK_END - fim do arquivo