Linguagem C: Arquivos

Sumário

- Introdução;
- Streams e Arquivos;
- Fundamentos do Sistema de Arquivos;
  - Abrindo um Arquivo;
  - Fechando um Arquivo;
  - Escrevendo um caracter;

Introdução

- A linguagem C não contém nenhum comando de E/S. Ao contrário, todas as operações de E/S ocorrem mediante chamadas de biblioteca C padrão;
- Essa abordagem faz o sistema de arquivos de C extremamente poderoso e flexível;
- O sistema de E/S de C é único, porque dados podem ser transferidos na sua representação binária interna ou em um formato de texto legível por humanos;

Streams e Arquivos

- Antes de começar a falar do sistema de arquivos C ANSI, é importante entender a diferença entre os termos streams e arquivos;

- O sistema de arquivos de C é projetado para trabalhar com uma ampla variedade de dispositivos, incluindo terminais, acionadores de disco e acionadores de fita;

- Embora cada um dos disposistivos seja muito diferente, o sistema de arquivo com buffer transforma-os em um dispositivo lógico chamado de stream;

Streams e Arquivos

- Portanto, o stream é uma abstração do dispositivo real chamado de arquivo;
- Pelo fato de as streams serem amplamente independentes do dispositivo, a mesma função pode escrever em um arquivo em disco ou em algum outro dispositivo, como o console;
- Existem dois tipos de streams:
  - Texto e binária

Streams e Arquivos

- Uma stream de texto é uma sequência de caracteres.
  - O padrão C ANSI permite (mas não exige) que uma stream de texto seja organizada em linhas terminadas por um caractere de nova linha;
  - Porém, o caracter de nova linha é opcional na última linha e é determinado pela implementação;

- Uma stream binária é uma sequência de bytes com uma correspondencia de um para com aqueles encontrados no dispositivo externo;

Streams e Arquivos

- Em C, um arquivo pode ser qualquer coisa, desde um arquivo em disco até um terminal ou uma impressora;
- Você associa uma stream com um arquivo específico realizando uma operação de abertura;
- Uma vez o arquivo aberto, informações podem ser trocadas entre ele e o seu programa;

Streams e Arquivos

- Cada stream associada a uma arquivo tem uma estrutura de controle de arquivo do tipo FILE;
  - Essa estrutura é definida no cabeçalho STDIO.H
- O sistema de arquivos C ANSI é composto de diveras funções inter-relacionadas;
- A maioria das funções começa com a letra "f". Isso é uma conversão do padrão C UNIX, que definiu dois sistemas de arquivos;

Fundamentos do Sistema de Arquivos

- As funções mais comuns do sistema de arquivo com buffer;

Nome	Função
fopen()	Abre um arquivo
fclose()	Fecha um arquivo
putc()	Escreve um caracter em um arquivo
fputc()	O mesmo que putc()
getc()	Lê um caracter de um arquivo
fgetc()	O mesmo que getc()
fseek()	Posiciona o arquivo em um byte específico
fprintf()	O printf de um arquivo
feof()	Devolve verdade se fim de arquivo for atingigo
ferror()	Devolve verdadeiro se ocorreu um erro
fewind()	Recoloca o indicador de posição de arquivo no início do arquivo
remove()	Apaga um arquivo
fflush()	Descarrega um arquivo

Fundamentos do Sistema de Arquivos

- O ponteiro é o meio comum que une o sistem C ANSI de E/S;
- Um ponteiro de arquivo é um ponteiro para informações que definem várias coisas sobre o arquivos como:
  - Nome, status e posição atual do arquivo;
- Basicamente, o ponteiro de arquivo identifica um arquivo específico em disco e é usado pela stream associadas para direcionar as operações das funções de E/S;

Abrindo um arquivo

- Um ponteiro de arquivo é uma variável ponteiro do tipo FILE;

```
FILE *fp;
```

- Para ler ou escrever arquivos, seu programa precisa usar ponteiros de arquivos;

- A função fopen() abre uma stream para uso e associa um arquivo a ela;

Abrindo um arquivo

FILE *fopen(const char* nomearq, const char* modo);

- Onde *nomearq* é um ponteiro para uma cadeia de caracteres que forma o nome válido de arquivo e pode incluir especificação de caminho de pesquisa(path);
- A string apontada por *modo* determina como o arquivo será aberto, a tabela ao lado mostra os valores legais para *modo*;

Modo	Significado
r	Abre um arquivo-texto para leitura
w	Cria um arquivo-texto para escrita
a	Anexa a um arquivo-texto
rb	Abre um arquivo binário para leitura
wb	Cria um arquivo binário para escrita
ab	Anexa a um arquivo binário
r+	Abre um arquivo-texto para leitura/escrita
w+	Cria um arquivo-texto para leitura/escrita
a+	Anexa ou cria um arquivo-texto para leitura/escrita
r+b	Abre um arquivo binário para leitura/escrita
w+b	Cria um arquivo binário para leitura/escrita
a+b	Anexa a um arquivo binário para leitura/escrita

Abrindo um arquivo

- Para abrir um arquivo chamado Teste, permitindo escrita, pode-sedigitar:

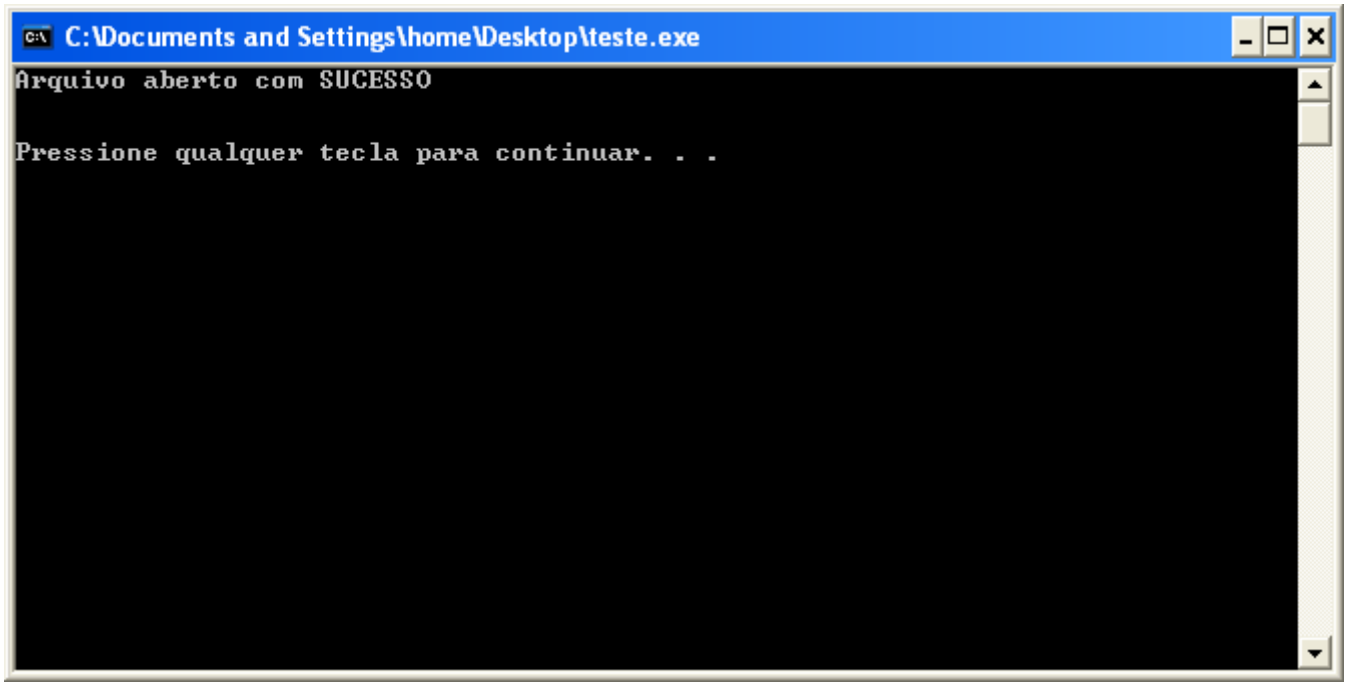
```
#include<stdio.h>
int main()
{
    FILE *f;
    f = fopen("teste", "w");
    system("pause");
}
```

- Embora tecnicamente correto, você geralmene verá o código anterior escrito desta forma:

```
FILE *f;
if((f = fopen("teste", "w")) == NULL)
{
      printf("Arquivo nao pode ser aberto\n\n\n");
      exit(1);
}
else
      printf("Arquivo aberto com SUCESSO\n\n\n");
```

Abrindo um arquivo

- Caso o arquivo teste esteja na mesma pasta que o código fonte teremos:



A função exit()

void exit(int codigo_de_retorno);

- A função exit() aborda a execução do programa. Ela pode ser chamada de qualquer ponto do programa e retorna ao sistema operacional o código de retorno;

- A convenção mais usada é que um programa retorne zero no caso de um término normal e retorne um número não nulo no caso de ter ocorrido um problema;

Fechando um arquivo

- A função fclose() fecha uma stream que foi aberta por meio de uma chamada fopen();
- Ela escreve qualquer dados que ainda permanece no buffer de disco no arquivo e, então, fecha normalmente o arquivo em nível de sistema operacional
- A função fclose() tem o protótipo a seguir:

int fclose(FILE *fp);

  - Onde fp é o ponteiro de arquivo devolvido pela chamada a fopen(); Um valor de retorno zero significa uma operações de fechamento bem-sucedida;

Escrevendo um caracter

- O padrão C ANSI define duas funções equivalentes para escrever caracteres em um arquivo:
  - putc();
  - fputc();

- A função putc() escreve em um arquivo que foi previamente aberto para escrita por meio da função fopen();

- O protótipo para essa função é:

int putc(char* ch, FILE *fp);

  - Onde fp é um ponteiro de arquivo devolvido por fopen() e ch é o caracter a ser escrito;

Escrevendo um caracter

- Vejamos um exemplo de escrita em arquivo:

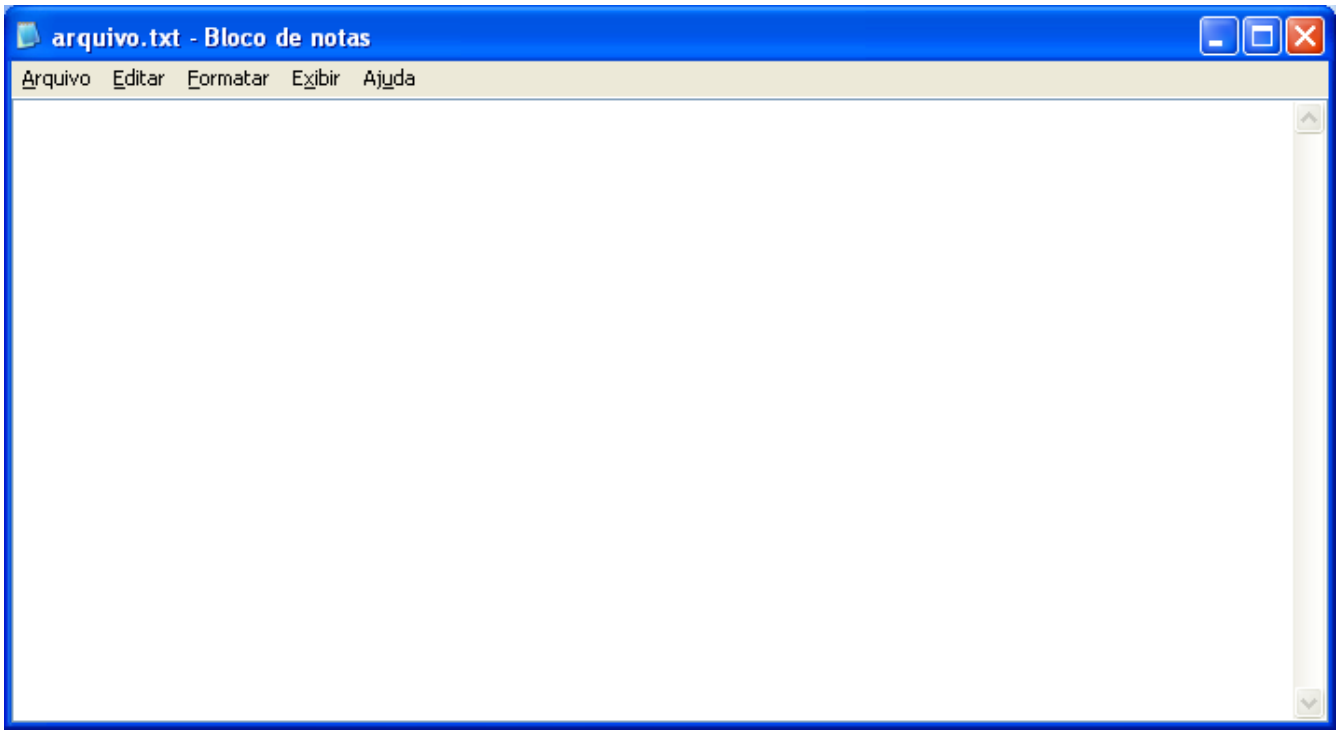
```c
#include<stdio.h>

int main()
{
    FILE *fp;
    char nomeArq[30]="teste.txt", nomePessoa[30];
    int i=0, j=10;

    printf("Este Prog. demonstra a ciração e gravação de dados em um arquivo");
    if(fp = fopen(nomeArq, "w") != NULL)
    {
        printf("\n\nArquivo criado com sucesso!!!");
        printf("\n\nPor favor, digite seu nome: ");
        gets(nomePessoa);
        fp = fopen(nomeArq, "w");
        for(i=0; nomePessoa[i]!='\0'; i++)
            fputc(nomePessoa[i], fp);
    }
    else
    {
        printf("Falha ao criar o arquivo");
        exit(1);
    }
    printf("\nLocalize o arquivo na pasta em que o prog. está rodando e abra-o\n\n");
    fclose(fp);
    system("pause");
}
```

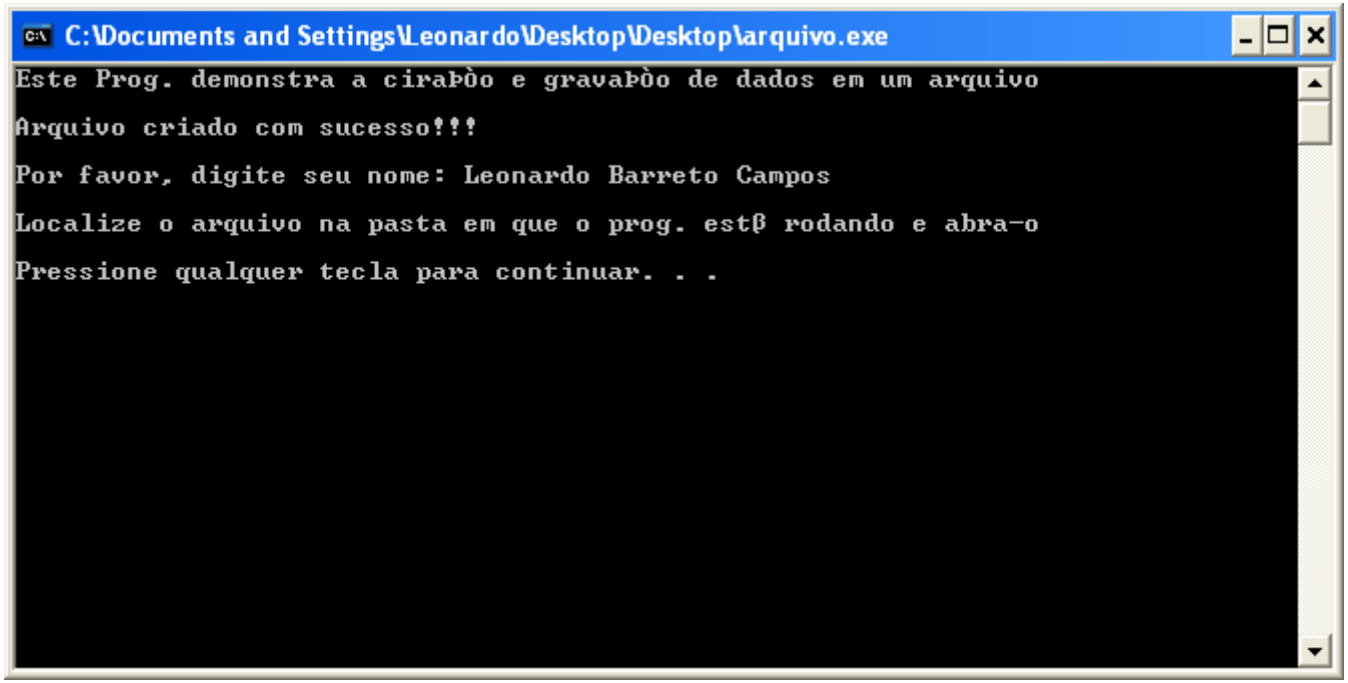
Escrevendo um caracter

- Vejamos o conteúdo de teste.txt antes da execução do programa:



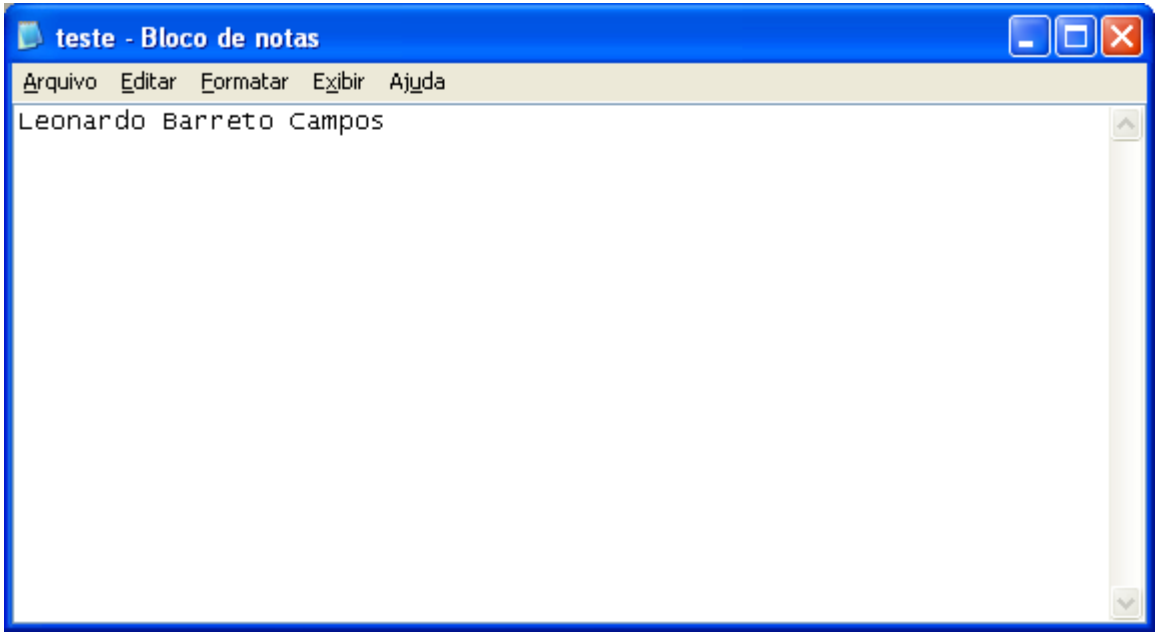
Escrevendo um caracter

- A saída no console para o programa anterior será:



Escrevendo um caracter

- Vejamos o conteúdo de teste.txt após a execução do programa:



Lendo um caracter

- O padrão C ANSI define duas funções equivalentes para ler caracteres de um arquivo:
  - getc();
  - fgetc();

- A função getc() lê caracteres de um arquivo aberto no modo leitura por fopen();

- O protótipo de getc() é:

```
int getc(FILE *fp);
```

  - Onde fp é um ponteiro de arquivo do tipo FILE devolvido por fopen();

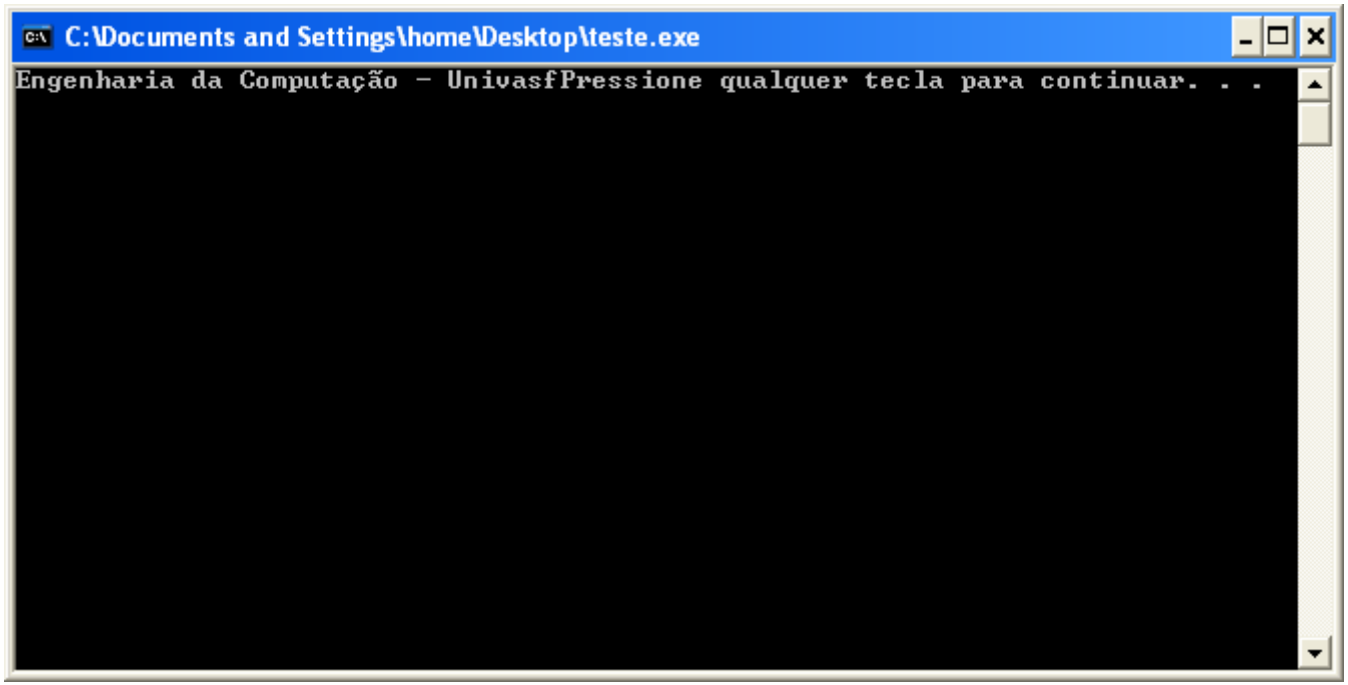
Lendo um caracter

- A função getc() devolve EOF (end of file) quando o final do arquivo for alcançado; Vejamos um exemplo:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
     FILE *fp;
     char ch;
     fp = fopen("arquivo.txt", "r");
     if(!fp)
     {
           printf("Arquivo nao pode ser aberto\n\n\n");
           exit(1);
     }
     while( (ch=getc(fp) ) != EOF)
       printf("%c", ch);
     fclose(fp);
     system("pause");
}
```

Lendo um caracter

- A saída no console para o programa anterior será:



Considerações Finais

- As funções fopen(), getc(), putc() e fclose() constituem o conjunto mínimo de rotinas de arquivos;
- Pode-se dizer que vimos 20% da linguagem C (sua sintaxe e seu poder no mundo da programação);
- Obrigado pela paciência e até a próxima disciplina;
- O SUCESSO ESTÁ EM SUAS MÃOS.

Bibliografia

- SCHILDT H. "*C Completo e Total*", Makron Books. SP, 1997.

- UFMG "Curso de Linguagem C", Universidade Federal de Minas Gerais.