

Tipos de dados

Streams são uma abstração usado nas linguagens C e C++ para operações de entrada e saída através de dispositivos de E/S baseados em caracteres, como arquivos, teclado, impressora, tela e porta de E/S.

O Ponteiro de arquivo

Um ponteiro de arquivo identifica um arquivo específico. A stream é representada pelo ponteiro de arquivo do tipo FILE que contém informações sobre as propriedades e indicadores de um arquivo. Para obter um variável ponteiro de arquivos, usa-se o comando :

```
FILE * pF;
```

Propriedades

Um stream possui algumas propriedades que definem qual função pode ser utilizada ou como a função tratará o stream. Muitas funções são definidas no parâmetro MODE quando a função de abertura de arquivo, fopen, for chamada.

Acesso

Especifica se a operação a ser executada com o stream será para leitura e/ou gravação no arquivo

Texto / Binário

Arquivo texto é aquele cujas linhas são delimitadas por um caractere especial de fim de linha (EOL). O fim do arquivo texto é definido pela ocorrência de um caractere de fim de arquivo (EOF).

Arquivo binário é um arquivo onde cada byte é lido ou gravado com um caractere, nenhuma transformação é executada e o final do arquivos coincide com o fim físico do arquivo.

Buffer

Um buffer é um bloco de memória onde os dados são acumulados antes de serem fisicamente lidos ou gravados no arquivo.

Indicadores

Um stream tem alguns indicadores que especificam o estado corrente dele. Os indicadores são modificados internamente e afetam o comportamento das funções de E/S.

Error indicator

Este indicador é ativado quando ocorre um erro na operação relacionada com o stream. O indicador pode ser checado utilizando a função `ferror`, e pode ser alterado utilizando a função `clearerr` ou por funções de reposicionamento (`rewind`, `fseek`, `fsetpos`).

End-Of-File

Quando este indicador está ativo, a última operação de leitura ou gravação realizada encontrou o fim do arquivo associado com o stream. Pode ser checado com a função `feof`, e pode ser alterado utilizando a função `clearerr` ou por funções de reposicionamento (`rewind`, `fseek`, `fsetpos`).

Position Indicator

Este indicador é um ponteiro interno que aponta para o próximo caractere dentro do stream que está sendo lido ou gravado pela próxima operação de E/S. Este valor pode ser obtido pela função `ftell` e `fgetpos` e pode ser chamado pelas funções `rewind`, `fseek` e `fsetpos`.

Standard Stream

Quando um programa inclui `cstdio` no começo da execução três streams predefinidos são abertos:

stdin

Este é o stream padrão para entrada. Por default corresponde ao teclado, mas pode ser redirecionado pelo sistema operacional.

stdout

Este é o stream padrão para saída. Por default corresponde à tela, mas pode ser redirecionado para um arquivo ou para outro dispositivo de saída.

stderr

Este é o stream padrão para erros.

Tipos de dados

Abrindo um arquivo

A função `fopen()` abre um arquivo o qual é armazenado com o nome informado e retorna um ponteiro. Operações permitidas para o arquivo são definidas como parametros de modo de abertura.

Protótipo

`FILE * fopen (const char * nome_arquivo, const char * modo_abertura);`

Parâmetros

Nome_arquivo identifica o nome do arquivo que será aberto. Este parâmetro segue as especificações do sistema operacional e pode incluir um caminho de localização (path).

Modo_abertura é o tipo de acesso requerido , e pode ser :

Modo	Descrição	Situação
r	Abrir um arquivo para leitura	O arquivo deve existir
w	Criar um arquivo vazio para gravação	Se existir o arquivo sera zerado
a	Acrescenta dados no final do arquivo	Se não existir arquivo sera criado
r+	Abrir um arquivo para leitura e gravação	O arquivo deve existir
w+	Cria um arquivo vazio para leitura e gravação	Se existir o arquivo sera zerado
a+	Abrir um arquivo para leitura e acrescentar dados	Se não existir arquivo sera criado

O parametro modo_abertura também serve para especificar se arquivo sera texto ou binario adicionando as letras t e b no modo de abertura

Tipo	Descrição
t	Modo texto, o final é do arquivo é assumido ao encontrar CTRL-Z
b	Modo binário, o final do arquivo é assumido ao encontrar o último caractere

Valor de Retorno

Se o arquivo foi aberto com sucesso a função retorna um ponteiro para o arquivo, caso contrario é retornado um ponteiro NULO.

Exemplo:

```
FILE * pF;  
pF = fopen ("c:/temp/arquivo.txt", "w");           // sem avaliar  
abertura  
if ( ( pF = fopen ( fopen ("c:/temp/arquivo.txt", "w") )    = = NULL )  
{  
    cout << "\n erro de abertura ";  
    return -1;  
}
```

Fechando um arquivo

A função fclose() fecha uma stream que foi aberta por meio de uma chamada a função fopen().

Protótipo

```
int fclose (FILE * stream);
```

Parâmetros

Stream é o ponteiro do tipo FILE especificando a stream que será fechada.

Exemplo;

```
fclose(pF) ;
```

Tipos de dados

Escrevendo um caractere

A função fputc() grava um caractere na posição especificado no stream e incrementa o ponteiro de posicionamento para o proximo caractere.

Protótipo

```
int fputc (int caractere, FILE * stream);
```

Parâmetros

Caractere que sera gravado, embora declara como int , a função converte-o para unsigned char antes de grava-lo

Stream é um ponteiro para um arquivo aberto

Valor de retorno

Se não existir erro o caractere é retornado, se um erro ocorrer , EOF é retornado

Exemplo: File_0301

Objetivo: Uso da função fputc

```
#include <stdio.h>
```

```
#include <iostream.h>
```

```
int main ()
```

```
{
```

```
    FILE * pFile;
```

```
    char cValor;
```

```
    pFile = fopen ("c:/temp/dados.txt","w");
```

```
    if (pFile!=NULL)
```

```
    {
```

```
        for (cValor = 'A' ; cValor <= 'Z' ; cValor++)
```

```
        {
```

```
            fputc (cValor , pFile);
```

```
            cout << cValor; // caracteres gravados no arquivo
```

```
        }
```

```
    fclose (pFile);
```

```
    }
```

```
    return 0;
```

```
}
```

Tipos de dados

Lendo um caractere

A função fgetc() retorna o proximo caractere do stream e incrementa para ponteiro para o caractere seguinte.

Protótipo

```
int fgetc (FILE * stream);
```

Parâmetros

Stream é um ponteiro para um arquivo aberto

Valor de retorno

O caractere lido é retornado como int, se um erro ocorrer , EOF é retornado

Exemplo: File_0401

Objetivo: Uso da função fgetc

```
#include <stdio.h>
```

```
#include <iostream.h>
```

```
int main ()
```

```

{
    FILE * pFile;
    char cValor;
    int n = 0;
    pFile=fopen ("c:/temp/dados.txt","r");
    if (pFile==NULL) perror ("Erro na abertura do arquivo");
    else
    {
        do {
            cValor = fgetc (pFile);
            cout << cValor;
            n++;
        } while (cValor != EOF);
        fclose (pFile);
        cout << "\n Arquivo contem " << n-1 << " caracteres \n";
    }
    return 0;
}
/*
ABCDEFGHGIJKLMNOPQRSTUVWXYZ
Arquivo contem 26 caracteres
Press any key to continue
*/

```

Tipos de dados

Final de arquivo

A função `feof()` verifica se o final do arquivo foi encontrado.

Protótipo

```
int feof (FILE * stream);
```

Parâmetros

Stream é um ponteiro para um arquivo aberto

Valor de retorno

Um valor diferente de zero é retornado quando o indicador de posição encontrar final de arquivo, caso contrário retornará 0.

```

Exemplo: File_0501
Objetivo: Uso da função feof
#include <stdio.h>
#include <iostream.h>
int main ()
{
    FILE * pFile;
    char cValor;
    int n = 0;

```

```

pFile=fopen ("c:/temp/dados.txt","r");
if (pFile==NULL) perror ("Erro na abertura do arquivo");
else
{
    while (!feof(pFile)){
        cValor = fgetc (pFile);
        cout << cValor;
        n++;
    }
    fclose (pFile);
    cout << "\n Arquivo contem " << n-1 << " caracteres \n";
}
return 0;
}/*
ABCDEFGHGIJKLMNOPQRSTUVWXYZ
Arquivo contem 26 caracteres
Press any key to continue
*/

```

Tipos de dados

Lendo um bloco de dados

Protótipo

```
int feof (FILE * stream);
```

Parâmetros

Stream é um ponteiro para um arquivo aberto

Valor de retorno

Um valor diferente de zero é retornado quando o indicador de posição encontrar final de arquivo, caso contrário retornará 0.

```

Exemplo: File_0501
Objetivo: Uso da função feof
#include <stdio.h>
#include <iostream.h>
int main ()
{
    FILE * pFile;
    char cValor;
    int n = 0;
    pFile=fopen ("c:/temp/dados.txt","r");
    if (pFile==NULL) perror ("Erro na abertura do arquivo");
    else
    {

```

```
while (!feof(pFile)){
    cValor = fgetc (pFile);
    cout << cValor;
    n++;
}
fclose (pFile);
cout << "\n Arquivo contem " << n-1 << " caracteres \n";
}
return 0;
}/*
ABCDEFGHIIJKLMNOPQRSTUVWXYZ
Arquivo contem 26 caracteres
Press any key to continue
*/
```