

Aula 16

Funções de manipulação de arquivos

Os direitos desta obra foram cedidos à Universidade Nove de Julho

Aula 16 – Funções de manipulação de arquivos

Caro aluno,

Se você for recortar as soluções dos programas deste arquivo e colá-las no compilador DevC++, tome cuidado pois as aspas e o sinal de menos precisam ser substituídos, senão vai dar erro.

Outra dica: não use acentos nas palavras, pois não funcionam nos programas.

Lista de exercícios da aula 16

Vamos praticar um pouco mais usando outras funções de manipulação de arquivos. Você deve tentar fazer os exercícios sozinho e somente depois ver as respostas, ok? Boa sorte!

1. Faça um programa em C que peça e leia o nome de um arquivo. Em seguida, crie o arquivo texto, grave alguns dados nele usando a função `fprint()` e feche o arquivo. Depois abra o arquivo e mostre na tela os dados gravados, lendo-os com a função `fscanf()`.

2. Faça o que se pede:

Leia sobre a função `ferror()`:

- Arquivo cabeçalho a ser incluído: **<stdio.h>**.
- Recebe como argumento um ponteiro para arquivo.
- Retorna verdadeiro na ocorrência de um erro durante a última operação no arquivo e falso, caso contrário. Deve ser chamada imediatamente após cada operação, porque a condição de erro é constantemente modificada.

Leia sobre a função `rewind()`:

- Arquivo cabeçalho a ser incluído: **<stdio.h>**.
- Reposiciona o indicador de posição do arquivo no início do arquivo.
- Recebe por argumento um ponteiro para arquivo.

Agora implemente e teste o programa a seguir, mostrando os resultados que serão apresentados na tela.

```
/* Error.c - Ilustra o uso da função ferror() */

#include<stdio.h>

#include<stdlib.h>

int main()
{
    FILE *fp;

    char nome_arq[20];

    char str[50];

    system("color 1e");

    puts("\nForneca o nome de um arquivo a ser criado:\n");

    gets(nome_arq);

    puts("\nAbrindo o arquivo no modo de escrita");

    if( (fp=fopen(nome_arq,"w")) == NULL)
    { puts("\nNao foi possivel abrir o arquivo.");
      getch();
      exit(1);
    }

    puts("\nVamos escrever qualquer coisa no arquivo. Por ex:");
    puts("NONONONONONO\n");
    fprintf(fp,"NONONONONONO");

    puts("\nEsperamos que esta operacao nao resulte em erro.");
    puts("Afimal o arquivo foi mesmo aberto para escrita.\n");
    puts("Vamos testar a condicao de erro:");

    if (ferror(fp))

        printf("\nErro de escrita no arquivo: %s\n", nome_arq);
```

```

else

    puts("\nNenhum erro ocorreu.");

puts("\nAgora, vamos rebobinar o arquivo.\n");

rewind(fp);

puts("\nAgora, vamos ler o que foi escrito no arquivo.");

fscanf(fp,"%s", str);

puts("\nVamos testar novamente a condicao de erro:");

if (ferror(fp))
{ printf("\nErro de leitura no arquivo: %s\n", nome_arq);
  puts("\nIsto era mesmo esperado, pois o arquivo foi aberto apenas para escrita.\n");
}
else

    puts("\nNenhum erro ocorreu.");

puts("\n\nAu revoir!"); getch();

return 0;

}

```

3. Faça o que se pede:

Leia sobre a função `remove()`:

A função `remove()` permite que se apague/remova um arquivo. Vejamos as características da função `remove()`:

- Arquivo cabeçalho a ser incluído: **<stdio.h>**.
- Recebe como argumento o nome do arquivo a ser removido.
- Devolve zero se for bem sucedida e um valor diferente de zero, caso contrário.

Observe o trecho de código a seguir e transforme-o em 2 programas: um que peça o nome, crie um arquivo texto com o nome fornecido e insira qualquer dado nele; e outro que peça

o nome de um arquivo e, caso ele exista, remova-o, senão imprima uma mensagem dizendo que o arquivo não existe. Você pode mostrar os arquivos do diretório com a função `system("dir *.*");`

```
FILE *fp1;

char nome_arquivo[30];

...

if ((fp1=fopen(nome_arquivo,"w")) == NULL)
{
    printf("\nArquivo nao pode ser criado!\n");
    getch();
    exit(1);
}

. . .

if(remove(nome_arquivo))
{
    printf("\nArquivo nao pode ser apagado!\n");
    getch();
    exit(1);
}
else
{
    printf("\nArquivo %s removido com sucesso. ",
           nome_arquivo);
    getch();
}
```

Respostas dos exercícios

1.

```
/* FprintfFscanf.c: Ilustra o uso das funções fprintf() e fscanff() para gravação e
leitura de arquivos */
#include<stdio.h>
#include<stdlib.h>

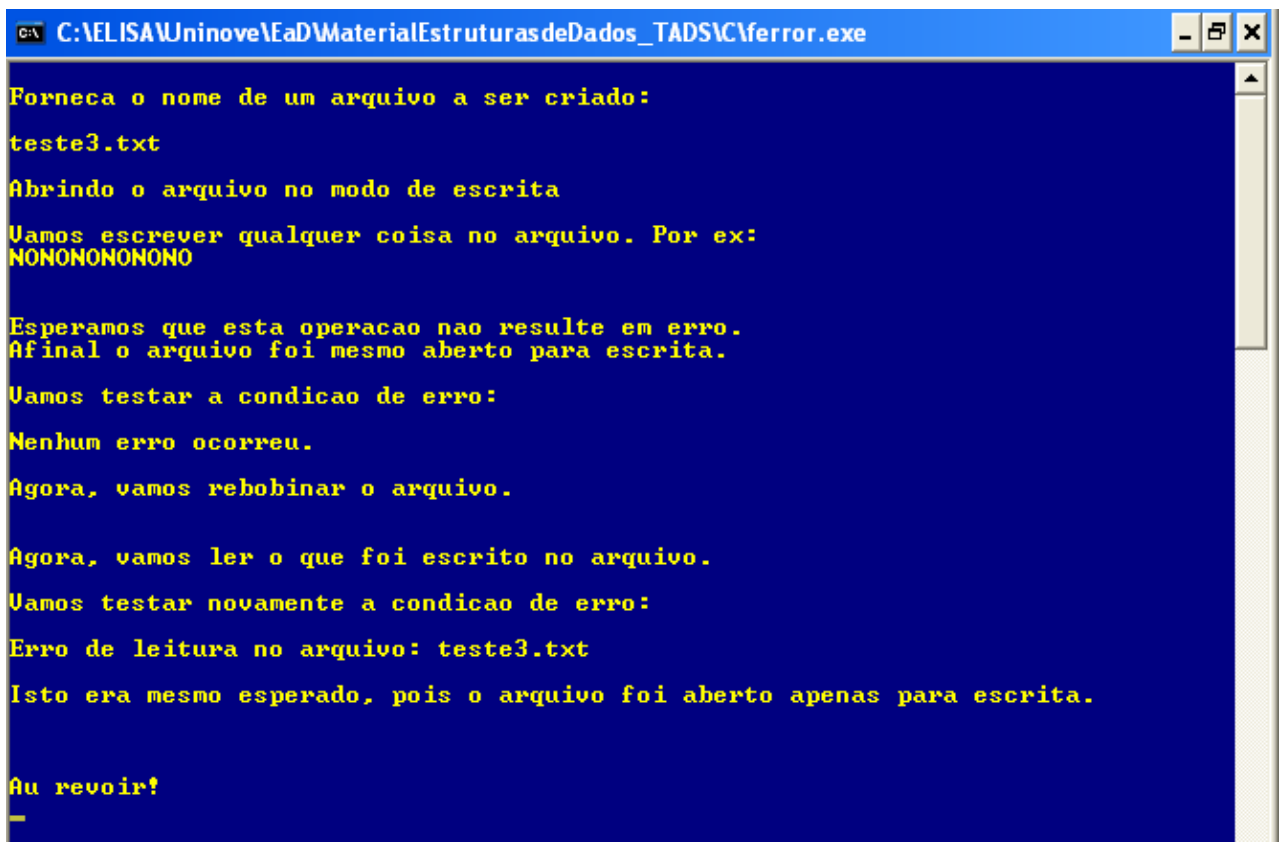
int main()
{
    FILE *fp;
    float db=10.1010, db1;
    int i=10, i1;
    char nome_arq[20];

    puts("Forneca o nome do arquivo a ser criado:");
    scanf("%s",nome_arq);
    puts("\nAbrindo o arquivo no modo escrita:");
    if( (fp=fopen(nome_arq, "w"))== NULL)
    { puts("O arquivo nao pode ser aberto!");
      puts("O programa sera abortado.");
      exit(1);
    }

    printf("\nEscrevendo com fprintf() no arquivo %s o conteúdo de db=%f e
i=%d\n", nome_arq, db, i);
    fprintf(fp, "%f %d", db, i);
    puts("Fechando o arquivo.");
    fclose(fp);
    puts("Abrindo-o agora no modo de leitura.");
    if( (fp=fopen(nome_arq, "r"))== NULL)
    { puts("O arquivo nao pode ser aberto!");
      puts("O programa sera abortado.");
      exit(1);
    }

    printf("\nLendo com fscanff() do arquivo %s para as variaveis db1 e i1\n",
nome_arq);
    fscanff(fp,"%f %d", &db1, &i1);
    printf("\nOs valores lidos sao: db1 = %f e i1 = %d\n",db1, i1);
    puts("\nFechando o arquivo...");
    fclose(fp);
    getch();
    return 0;
}
```

2.



```
C:\ELISA\Uninove\EaD\MaterialEstruturasdeDados_TADS\CVerror.exe

Forneca o nome de um arquivo a ser criado:
teste3.txt
Abrindo o arquivo no modo de escrita
Vamos escrever qualquer coisa no arquivo. Por ex:
NONONONONONO

Esperamos que esta operacao nao resulte em erro.
Afiml o arquivo foi mesmo aberto para escrita.

Vamos testar a condicao de erro:
Nenhum erro ocorreu.
Agora, vamos rebobinar o arquivo.

Agora, vamos ler o que foi escrito no arquivo.
Vamos testar novamente a condicao de erro:
Erro de leitura no arquivo: teste3.txt
Isto era mesmo esperado, pois o arquivo foi aberto apenas para escrita.

Au revoir!
_
```

3.

```
//criaarq.c: cria um arquivo texto e insere qualquer dado nele
// verifica se o arquivo foi criado pela função system dir

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

FILE *fp1;
char nome_arquivo[30], frase[100]="Escrevendoqualquerdadonoarquivo";

int main()
{
    system("color 1e");
    printf("\nDigite o nome do arquivo texto a ser criado: ");
    scanf("%s",nome_arquivo);

    if ((fp1=fopen(nome_arquivo,"w")) == NULL)
    {
        printf("\nArquivo nao pode ser criado!\n");
        getch();
        exit(1);
    }

    fprintf(fp1, "%s", frase);

    fclose(fp1);

    printf("\n Arquivo %s criado com sucesso! ",nome_arquivo);
}
```

```

printf("\n\nMostra os arquivos do diretorio: \n");
system("dir *.*");
getch();

printf("\n\n Fim do programa");
getch();
return 0;
}

```

```

C:\ELISA\Uninove\EaD\MaterialEstruturasdeDados_TADS\C\criaarq.exe
Digite o nome do arquivo texto a ser criado: z.tst
Arquivo z.tst criado com sucesso!
Mostra os arquivos do diretorio:
O volume na unidade C não tem nome.
O número de série do volume é F417-A65E

Pasta de C:\ELISA\Uninove\EaD\MaterialEstruturasdeDados_TADS\C
22/06/2012  16:51                31 z.tst
               1 arquivo(s)                31 bytes
               0 pasta(s) 419.483.086.848 bytes disponíveis

Fim do programa

```

Caro aluno, muito cuidado para não remover arquivos importantes!

//removearq.c: mostra o diretório pela função system dir
 // pede o nome e remove o arquivo se ele existir

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

```

```

FILE *fp1;
char nome_arquivo[30];

```

```

int main()
{
    system("color 1e");

    printf("\n\nMostra os arquivos do diretorio antes da remocao: \n");
    system("dir *.*");
    getch();

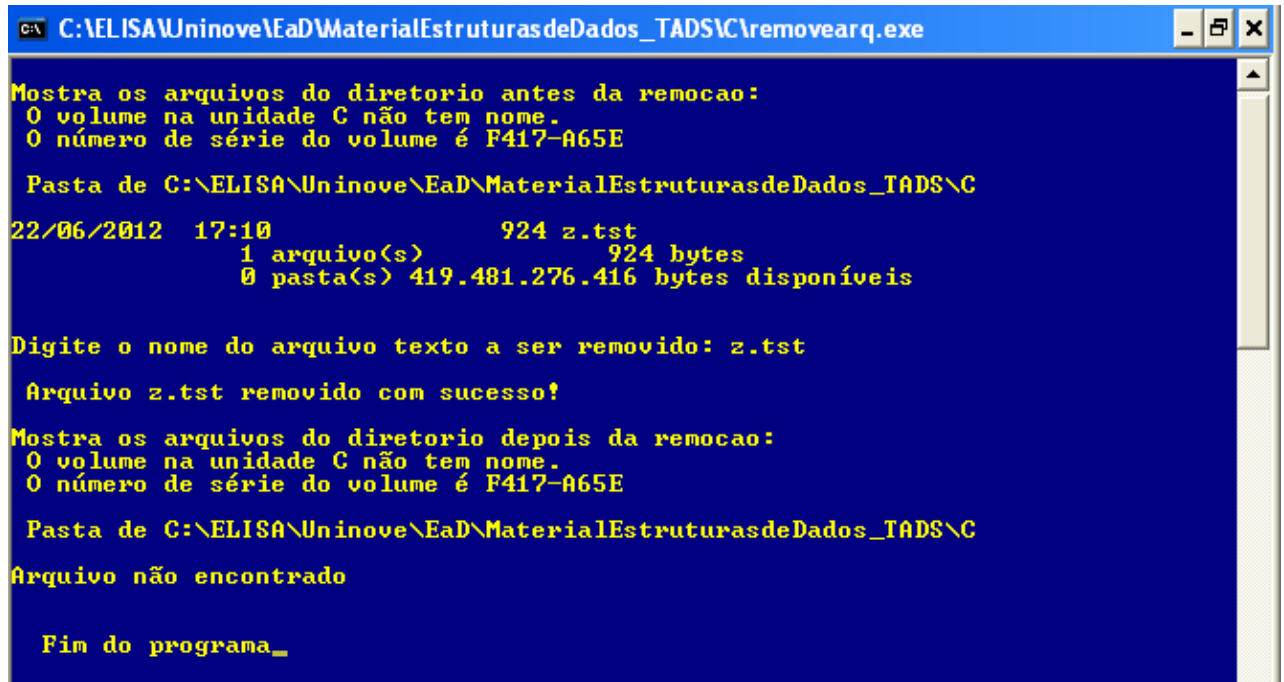
    printf("\n\nDigite o nome do arquivo texto a ser removido: ");
    scanf("%s",nome_arquivo);

    if (remove(nome_arquivo)==1)
    {
        printf("\nArquivo nao existe!\n");
        getch();
        exit(1);
    }
    else
    {
        printf("\n Arquivo %s removido com sucesso! ",nome_arquivo);
        printf("\n\nMostra os arquivos do diretorio depois da remocao: \n");
    }
}

```



```
        system("dir *.*");  
        getch();  
    }  
  
    printf ("\n\n Fim do programa");  
    getch();  
    return 0;  
}
```



```
C:\ELISA\Uninove\EaD\MaterialEstruturasdeDados_TADS\C\removearq.exe  
Mostra os arquivos do diretorio antes da remocao:  
O volume na unidade C não tem nome.  
O número de série do volume é F417-A65E  
  
Pasta de C:\ELISA\Uninove\EaD\MaterialEstruturasdeDados_TADS\C  
22/06/2012  17:10                924 z.tst  
          1 arquivo(s)                924 bytes  
          0 pasta(s) 419.481.276.416 bytes disponíveis  
  
Digite o nome do arquivo texto a ser removido: z.tst  
Arquivo z.tst removido com sucesso!  
Mostra os arquivos do diretorio depois da remocao:  
O volume na unidade C não tem nome.  
O número de série do volume é F417-A65E  
  
Pasta de C:\ELISA\Uninove\EaD\MaterialEstruturasdeDados_TADS\C  
Arquivo não encontrado  
  
Fim do programa_
```