



Algoritmos e Programação II

Prof. Joilson dos Reis Brito

Roteiro da Aula

- ✓ Correção exercícios
- ✓ Continuação Arquivos
- ✓ Desenvolvimento de Exercícios - Lista08

Lista 11 Exercício 3

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
#include<locale.h>
#define TAM 100
int main(int argc,char *argv[])
{
    system("cls");
    setlocale(LC_ALL,"Portuguese");
    FILE *PonteiroArquivo;
    char NomeArq[TAM];
    char Character;
    int l = 0, TotalA=0,TotalE=0,Totall=0,TotalO=0,TotalU=0;
    strcpy(NomeArq,argv[1]);
    PonteiroArquivo = fopen(NomeArq,"r");
    if(PonteiroArquivo == NULL)
    {
        printf("\n Nao consigo abrir o arquivo ! ");
        exit(1);
    }
}
```

Lista 11 Exercício 3 - Continuação

```
else
{
    Character = fgetc(PonteiroArquivo);
    while(Character != EOF)
    {
        Character = toupper(Character);
        switch (Character)
        {
            case 'A': TotalA ++;break;
            case 'E': TotalE ++;break;
            case 'I': TotalI ++;break;
            case 'O': TotalO ++;break;
            case 'U': TotalU ++;break;
        }
        Character = fgetc(PonteiroArquivo);
    }
}
printf("\nNo arquivo %s tem:\n",NomeArq);
printf("\n%i vogal A ou a",TotalA);
printf("\n%i vogal E ou e",TotalE);
printf("\n%i vogal I ou i",TotalI);
printf("\n%i vogal O ou o",TotalO);
printf("\n%i vogal U ou u\n",TotalU);
fclose(PonteiroArquivo);
}
```

Lista 11 Exercício 4

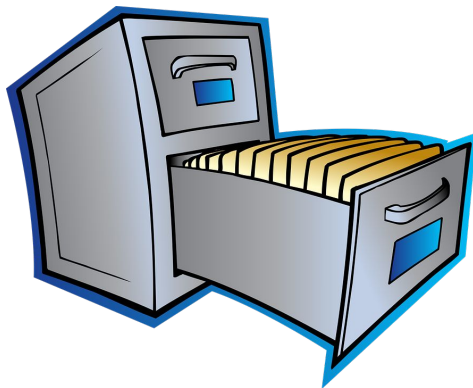
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<locale.h>
#define TAMANHO 100
int main(int argc,char *argv[])
{
    system("cls");
    setlocale(LC_ALL,"Portuguese");
    FILE *PonteiroArquivo;
    char String[TAMANHO], StringAux[TAMANHO], NomeArq[TAMANHO];
    char Character;
    int l = 0, Contador = 0;
    strcpy(NomeArq,argv[1]);
    strcpy(String,argv[2]);
    PonteiroArquivo = fopen(NomeArq,"r");
    if(PonteiroArquivo == NULL)
    {
        printf("\n Nao consigo abrir o arquivo ! ");
        exit(1);
    }
}
```

Lista 11 Exercício 4 - Continuação

```
else
{
    Character = fgetc(PonteiroArquivo);
    while(Character != EOF)
    {
        if(Character != ' ' && Character != '\n' && Character != ',')
        {
            StringAux[I] = Character;
            I++;
        }
        else
        {
            StringAux[I] = '\0';
            if(stricmp(String,StringAux) == 0)
                Contador++;
            I = 0;
        }
        Character = fgetc(PonteiroArquivo);
    }
}

printf("\n\nA palavra %s aparece %i vezes no arquivo %s.\n\n",String,Contador,NomeArq);
fclose(PonteiroArquivo);
system("Pause");
return 0;
}
```

CONTINUAÇÃO ARQUIVOS



Leitura de strings de Arquivo

fgets

Função que realiza a leitura de uma string em um arquivo.

Protótipo:

char *fgets (char *str, int tamanho, FILE *fp);

A função fgets recebe 3 argumentos:

- a string a ser lida,
- o limite máximo de caracteres a serem lidos;
- e o ponteiro para FILE, que está associado ao arquivo de onde a string será lida.

Leitura de strings de Arquivo

- A função lê a string até que um caracter de nova linha seja lido ou tamanho-1 caracteres tenham sido lidos.
- Se o caracter de nova linha ('\n') for lido, ele fará parte da string, o que não acontecia com gets.
- A string resultante sempre terminará com '\0' (por isto somente tamanho-1 caracteres, no máximo, serão lidos).

Exemplo 1

```
#include<stdio.h>
#include<locale.h>
#include<stdlib.h>
int main()
{
    FILE *PonteiroArquivo;
    char String[100];
    char NomeArquivo[100];
    setlocale(LC_ALL,"Portuguese");
    printf("Digite o nome do arquivo: ");
    gets(NomeArquivo);
    ponteiroArquivo = fopen(NomeArquivo,"r");
    if(PonteiroArquivo == NULL)
        printf("Impossível abrir o arquivo %s!\n",NomeArquivo);
    else
    {
        printf("Palavras armazenadas no arquivo: \n");
        while(!feof(PonteiroArquivo))
        {
            fgets(String, 99, PonteiroArquivo);
            printf("%s",String);
        }
        fclose(PonteiroArquivo);
        printf("\n");
        system("Pause");
    }
    return 0;
}
```

Escrita de strings de Arquivo

fputs

Função que escreve uma string em um arquivo.

Protótipo:

char *fputs (char *str, FILE *fp);

A função grava a string **str** no arquivo **fp**.

Verificação de Erros ao Acessar o Arquivo

ferror

Verifica se um acesso ao arquivo obteve sucesso.

Protótipo:

int ferror (FILE *fp);

A função retorna:

- zero, se nenhum erro ocorreu;
- um número diferente de zero se algum erro ocorreu durante o acesso ao arquivo.

Verificação de Erros ao Acessar o Arquivo

error

Se torna muito útil quando queremos verificar se cada acesso a um arquivo teve sucesso, de modo que consigamos garantir a integridade dos nossos dados. Na maioria dos casos, se um arquivo pode ser aberto, ele pode ser lido ou gravado. Porém, existem situações em que isto não ocorre.

Por exemplo, pode acabar o espaço em disco enquanto gravamos, ou o disco pode estar com problemas e não conseguimos ler, etc.

Exemplo 2

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

#include <locale.h>int main()
{
    FILE *PonteiroArquivo;
    char String[100];
    PonteiroArquivo = fopen("Arquivo.txt","w");
    if(PonteiroArquivo == NULL)
    {
        printf("\n Nao consigo abrir o arquivo ! ");
        exit(1);
    }
    do
    {
        printf("\nDigite uma nova string. Para terminar, digite <enter>: ");
        gets(String);
        if(String[0] != '\n')
        {
            strcat(String,"\n");
            fputs(String, PonteiroArquivo);
        }
        if(ferror(PonteiroArquivo))
        {
            printf("Erro na gravacao\n");
            fclose(PonteiroArquivo);
            exit(1);
        }
    } while (String[0] != '\n');
    fclose(PonteiroArquivo);
    system("Pause"); return 0;
}
```

E se fosse necessário adicionar neste mesmo código a impressão do arquivo que foi lido?

Voltar para o Início do Arquivo

rewind

Retorna a posição corrente do arquivo para o início.

Protótipo:

void rewind (FILE *fp);

Exemplo 3

```
{
FILE *PonteiroArquivo;
char String[100];
PonteiroArquivo = fopen("frutas.txt","a+");
if(PonteiroArquivo == NULL)
{
    printf("\n Nao consigo abrir o arquivo ! ");
    exit(1);
}
do
{
    printf("\nDigite o nome de uma fruta. Para terminar, digite <enter>: ");
    gets(String);
    if(String[0] != '\0')
    {
        strcat(String,"\n");
        fputs(String, PonteiroArquivo);
        if(ferror(PonteiroArquivo))
        {
            printf("Erro na gravacao\n");
            fclose(PonteiroArquivo);
            exit(1);
        }
    }
} while (String[0] != '\0');
system("cls");
printf("Frutas armazenadas no arquivo: \n");
rewind(PonteiroArquivo);
while(!feof(PonteiroArquivo))
{
    fgets(String, 99, PonteiroArquivo);
    if(!feof(PonteiroArquivo))
        printf("%s",String);
}
fclose(PonteiroArquivo);
system("Pause");
return 0;
}
```

Entradas e Saídas Formatadas

- É possível ler ou escrever dados de forma formatada em arquivos usando as funções: ***fprintf*** e ***fscanf***.
- O funcionamento de `fprintf` e `fscanf` é similar às funções `printf` e `scanf`, tendo apenas mais um parâmetro inicial, que corresponde ao arquivo onde o processamento irá ser realizado.
- Protótipos:

`int fscanf(Arquivo, Código de Formação Variável, ...)`

`int fscanf(Arquivo, Código de Formação Variável, ...)`

A função `fscanf` retorna EOF se tiver detectado End-of-File, ou retorna o número de parâmetros que conseguiu ler com sucesso.

Entradas e Saídas Formatadas

As funções que permitem ler e escrever dados de forma formatada em arquivos devem ser utilizadas apenas para arquivos de texto.

Exemplo:

Suponha um arquivo que contém em cada linha o nome de um aluno (apenas uma palavra) e sua respectiva nota (0 ... 10).

Maria	7,1
João	2,9
José	1,2
Henrique	5,7

Exemplo 1 - Programa que cadastra os nomes e as notas dos alunos no arquivo alunos.txt

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#define Tam 100
int main()
{
    FILE *PontArq;
    char Nome[Tam];
    float Nota;
    setlocale(LC_ALL,"Portuguese");
    PontArq = fopen("Alunos.txt","w");
    if(PontArq == NULL)
    {
        printf("Impossível Abrir o arquivo!\n");
        exit(1);
    }
    else
    {
        printf("Digite o nome de um aluno ou pressione <enter> para finalizar: ");
        fflush(stdin);
        gets(Nome);
        while(Nome[0] != '\0')
        {
            printf("Digite a nota do um aluno: ");
            scanf("%f",&Nota);
            fprintf(PontArq, "%s %f\n",Nome, Nota);
            printf("Digite o nome de um aluno ou pressione <enter> para finalizar: ");
            fflush(stdin);
            gets(Nome);
        }
        fclose(PontArq);
    }
    system("Pause");
    return 0;
}
```

Exemplo 2 - Programa que lê o arquivo Alunos.txt e imprime os nomes e as notas dos alunos que tiraram mais de 6.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
int main()
{
    FILE *PontArq;
    char Nome[100];
    float Nota;
    setlocale(LC_ALL,"Portuguese");
    PontArq = fopen("Alunos.txt","r");
    if(PontArq == NULL)
    {
        printf("Impossível Abrir o arquivo!\n");
        exit(1);
    }
    while(fscanf(PontArq,"%s %f",Nome, &Nota) != EOF)
    {
        if(Nota>=6)
        {
            printf("Nome: %s - Nota: %.2f\n",Nome,Nota);
        }
    }
    system("Pause");
    return 0;
}
```