



INF101202

Algoritmos e Programação

Modalidade Ead – Turma H

Material de apoio: arquivos texto
Processamento de arquivo texto:
com entrada e saída formatada

Arquivos texto: funções fscanf e fprintf

Para se **ler conteúdos/valores** de uma string/linha de um arquivo texto , pode-se usar a função, similar a scanf():

```
fscanf (<FILE*> , <"%codigoformato"> , <var1,var2,...> );
```

E, para se **escrever valores** em uma string/linha num arquivo, pode-se usar a função, similar a printf():

```
fprintf (<FILE *> , <"% codigoformato"> , <var1,var2,...> );
```

Ex: fscanf(meuarq,"%d",&m[i]);

fprintf(meuarq,"%f",d[3][6]);

Está-se lendo do meuarq
um valor inteiro para m[i]

Escreve-se em
meuarq o valor float
d[3][6]

Arquivo: operações para manipulação E/S

Observação:

se a linha prevista para o arquivo texto estiver terminado (sob o ponto de vista do programador!!), é necessário realizar a gravação de um fim de linha, ou seja do caractere \n.

Uma aplicação para exemplificar:
gravar uma matriz **d** de inteiros em um arquivo texto de nome **arq**.

```
for (i=0; i<MAX; i++)  
{  
    for (j=0; j<MAX; j++)  
        fprintf(arq,"%2d",d[i][j]);  
    fprintf(arq,"\n");  
}
```

Gravação do \n, fim de linha, quando há o término de uma linha da matriz **d**

Criação de uma matriz, gravação da matriz em um arquivo texto, apresentação da matriz

```
#include <stdio.h>
#include <stdlib.h>
#define MAXLIN 2
#define MAXCOL 3
void cria_arq (FILE *, int [] [MAXCOL], int, int);
void mostra_matriz (int [] [MAXCOL], int, int);
int main( )
{
    int m[MAXLIN][MAXCOL];
    FILE *arq;
    system("color 70");
    if (!(arq = fopen("primusarq.txt", "w")))// vincula a variavel arq ao arquivo
        primusarq
    {
        printf("\nErro na abertura!\n");
        system("pause");
    }
    else
    {
        cria_arq (arq, m, MAXLIN, MAXCOL);
        mostra_matriz (m, MAXLIN, MAXCOL);
        fclose(arq);
        system("pause");
        return 0;
    }
}
```

Criação de uma matriz, gravação da matriz em um arquivo texto, apresentação da matriz

```
void cria_arq (FILE * arq, int m[] [MAXCOL], int mlin, int mcol)
{
    int i,j;
    // Leitura dos valores de teclado para compor a matriz
    printf("Entre com os dados inteiros do intervalo [0,99] para a matriz m\n");
    for(i=0; i<mlin; i++)
        for(j=0; j<mcol; j++)
        {
            printf("Valor[%d] [%d]: ", i+ 1, j + 1);
            scanf("%2d", &m[i][j]);
        }
    // gravacao da matriz no arquivo primusarq
    for(i=0; i<mlin; i++)
    {
        for( j=0; j<mcol; j++)
            fprintf(arq,"%d ",m[i][j]); //gravacao de dados em arq
        fprintf(arq,"\n");
    }
}
```

Importante!!: gravar \n para separar as linhas da matriz no arquivo.

Criação de uma matriz, gravação da matriz em um arquivo texto, apresentação da matriz

```
void mostra_matriz (int m[ ][MAXCOL], int mlin, int mcol)
{
    int i,j;
    // Mostrar a matriz gravada
    printf("\n Matriz dada\n");
    for(i=0; i<mlin ; i++)
    {
        for(j=0; j<mcol; j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }
}
```

```
Entre com os dados inteiros do intervalo [0,99] para a matriz m
Valor[1] [1]: 1
Valor[1] [2]: 1
Valor[1] [3]: 2
Valor[2] [1]: 2
Valor[2] [2]: 3
Valor[2] [3]: 4

Matriz fornecida
1  1  2
2  3  4
Pressione qualquer tecla para continuar. . .
```

Leitura de uma matriz gravada em um arquivo texto e sua apresentação na tela

```
#include <stdio.h>
#include <stdlib.h>
#define MAXLIN 2
#define MAXCOL 3
void le_arq (FILE *, int [] [MAXCOL], int, int);
void mostra_matriz (int [] [MAXCOL], int, int);
int main( )
{
    int m[MAXLIN][MAXCOL];
    FILE *arq;
    system("color 70");
    if (!(arq = fopen("primusarq.txt","r"))) // vincula a variavel arq ao arquivo
        primusarq
        {
            printf("\nErro na abertura!\n");
            system("pause");
        }
    else
    {
        le_arq (arq, m, MAXLIN, MAXCOL);
        mostra_matriz (m, MAXLIN, MAXCOL);
        fclose(arq);
        system("pause");
        return 0;
    }
}
```

Leitura de uma matriz gravada em um arquivo texto e sua apresentação na tela

```
void le_arq (FILE * arq, int m[] [MAXCOL], int mlin, int mcol)
{
    int i,j;
    // Leitura da matriz a partir do arquivo primusarq
    for(i=0; i<mlin; i++)
        for( j=0; j<mcol; j++)
            fscanf(arq,"%d",&m[i][j]); // leitura dos valores do arquivo para
            matriz m
}

void mostra_matriz (int m[ ] [MAXCOL], int mlin, int mcol)
{
    int i,j;
    // Mostrar a matriz gravada
    printf("\nMatriz lida do arquivo\n");
    for(i=0; i<mlin ; i++)
    {
        for(j=0; j<mcol; j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }
}
```

```
Matriz lida do arquivo
1 1 2
2 3 4
Pressione qualquer tecla para continuar. . .
```


Arquivo: operações para manipulação E/S

*As funções `fscanf` e `fprintf` são generalizações da `scanf` e da `printf`, respectivamente, ou seja, caso o parâmetro `<FILE *>` delas especificar alguns dos dispositivos de E/S padrões, então poderão ser usadas como a `printf` e a `scanf`.*

Os dispositivos padrões de E/S também são reconhecidos como arquivos. Eles são estabelecidos por constantes predefinidas, conforme tabela a seguir:

Ex: as chamadas a seguir:

```
printf("%2f",real);
```

```
fprintf(stdout,"%2f",real);
```

são equivalentes.

Arquivo: operações para manipulação E/S

Constante	Significado	Dispositivo
stdin	Dispositivo padrão de entrada	teclado
stdout	Dispositivo padrão de saída	vídeo
stderr	Dispositivo padrão de erro	vídeo
stdaux	Dispositivo padrão auxiliar	Porta serial
stdprn	Dispositivo padrão de impressão	Porta paralela

Arquivo: operações para manipulação

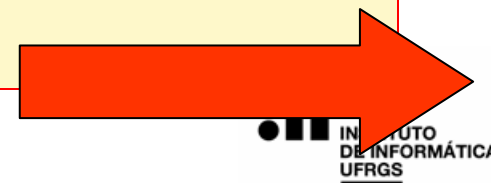
Os exemplos até agora manipularam com valores numéricos em arquivos textos. É relativamente simples usar a função `fscanf` para ler informações numéricas em um arquivo de dados.

Mas os dados no arquivo devem estar separados uns dos outros por pelo menos um espaço em branco.

Para os dados numéricos, o espaço em branco é um delimitador.

Mas, ao se tratar com dados alfanuméricos (strings, caracteres), é preciso empregar outra forma de leitura.

É o que será visto:



Arquivo: operações para manipulação

Para a leitura de dados como caracteres (alfanuméricos de um arquivo deve-se:

1. Ler toda a linha do arquivo como um string
2. Extrair substrings que correspondem aos dados, através de funções que decompõem a linha (string);
3. Converter cada substring para o tipo de dado correspondente.

Para isto se pode usar as funções, p.ex.:

1. `fgets(vetorcarac, tamvetor, ponteiro do arquivo),`
2. `strtok(vetorcarac, símbolo separador)`
3. `atoi(vetorcarac)` transforma vetorcaracteres em inteiro
ou a
`atof(vetorcarac)` que transforma vetorcarac em float

Arquivo: operações para manipulação

fgets(vetorcarac, tamvetor, ponteirodoarq);

permite ler toda uma linha de um arquivo como um string;

vetorcarac: onde o conteúdo da linha será armazenada

tamvetor: tamanho do vetor vetorcarac

ponteirodoarq: arquivo de onde será feita a leitura

Ex:

fgets(leitura, MAX, arq);

Vetor de tamanho MAX que reterá o string lido de arq

Arquivo: operações para manipulação

Os dados devem estar obrigatoriamente separados uns dos outros por delimitadores (caracteres) diferentes do branco, no arquivo.

Por exemplo: a vírgula, #, /

Ex: algumas linhas de um arquivo
Exemplo1:

2,Maria da Silva,8.4,5.7,8.8\n

5,José Ferreira,9.0,5.8,7.5\n

14,Agnes Lindeman,8.5,7.0,6.0\n

.....

ou, Exemplo2:

20#Ricardo Leão#8.4,5.7,8.8\n

55#João Pedreira#9.0,5.8,7.5\n

67#Dagmar Scherer#8.5,7.0,6.0\n

Arquivo: operações para manipulação: auxiliares

strtok(vetorcaracteres, “símboloseparador”);

A função **strtok** retorna o string correspondente ao 1º. valor encontrado em **vetorcaracteres** e que é separado dos outros pelo separador dado como parâmetro em **símboloseparador**.

Este valor(*substring*) possui um significado no contexto do string (vetorcaracteres) que é determinado pelo programador.

A este substring especial dá-se o nome de **token**.

Ex:

strtok(vetorbuffer,“,”);

De vetorbuffer é retirado o 1º. Substring que é determinado por todos os caracteres até o delimitador indicado em “,”

Arquivo: operações para manipulação-auxiliares

Seja: **A1= strtok(vetorbuffer, ",");**

isto é, do vetorbuffer é extraído um token (substring) até encontrar-se a vírgula.

Se, mais valores devem ser extraídos da linha vetorbuffer, então as chamadas da função strtok são feitas conforme:

A2= strtok(NULL, ",");

A3= strtok(NULL, ",");

Observação: o nome do vetor é usado na função strtok apenas para extrair o 1º token.

Para os demais "tokens" deve ser usada a constante NULL como primeiro parâmetro da strtok, sempre que o vetor é o mesmo, isto exigido pela definição da função.


```

#include<stdio.h>
#include<stdlib.h>
#include <ctype.h>    // necessario para toupper()
#include<string.h>    // necessario para strtok()
#define MAX 80
char opcao_menu();
void listar_notas(int);
int main()
{
    char op;
    system("color 70");
    do
    {
        op = opcao_menu();
        if (op == 'L')
            listar_notas(MAX);
        printf("\n");
        system("pause");
    }
    while (op != 'F');
    return 0;
}

```

Programa que lê um arquivo texto e o lista. Arquivo armazena codigo(int), nome(string) 2 notas(float). Apresenta a média das notas dos alunos.

inf

Continua....



```
char opcao_menu ( )    //menu com as operacoes
```

```
{  
    system("CLS");  
    printf("(L)istar notas\n");  
    printf("(F)im \n");  
    printf(">");  
    return(toupper(getchar()));  
}
```

```
void listar_notas(int maximo )
```

```
{  
    int num, notas;  
    float n1,n2,media;  
    char *nome;  
    char buf[maximo];  
    FILE *arq;  
    arq = fopen("dados.txt","r"); // abrir arq com opcao so de leitura: r  
    if (arq == NULL)  
    {  
        printf("Erro ao abrir arquivo \n");  
        system("pause");  
    }  
    else  
    {  
        (...)  
    }
```

**Programa que lê um arquivo texto e o lista.
Arquivo armazena codigo(int), nome(string)
2 notas(float). Apresenta a média das notas
dos alunos.**

Continua....



```

(...)
printf("\n");
printf("NUM |      NOME      |  N1 |  N2  \n");
printf("-----+-----+-----+-----\n");
notas=0;
media=0;
fgets(buf,maximo,arq);
while (!feof(arq))    // enquanto nao for fim de arquivo.....
{
    num = atoi(strtok(buf,",")); //
    nome = strtok(NULL,"," );
    n1 =atof( strtok(NULL,","));
    n2 =atof( strtok(NULL,","));
    printf("%03d |  %20s | %",
    notas = notas +2;
    media = media + n1 + n2;
    fgets(buf,maximo,arq);
}
printf("-----+-----+-----+-----\n");
media = media/notas;
printf("Media das notas = %4.1f\n", media);
fclose(arq);
}
}

```

**Programa que lê um arquivo texto e o lista.
Arquivo armazena codigo(int), nome(string)
2 notas(float). Apresenta a média das notas
dos alunos.**

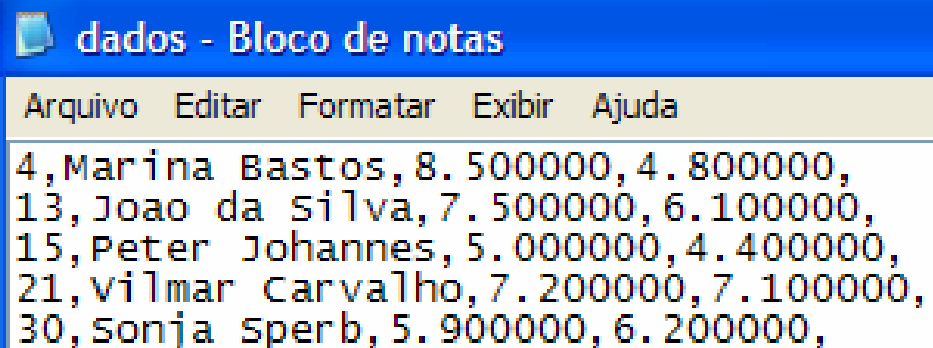
inf

Continua....



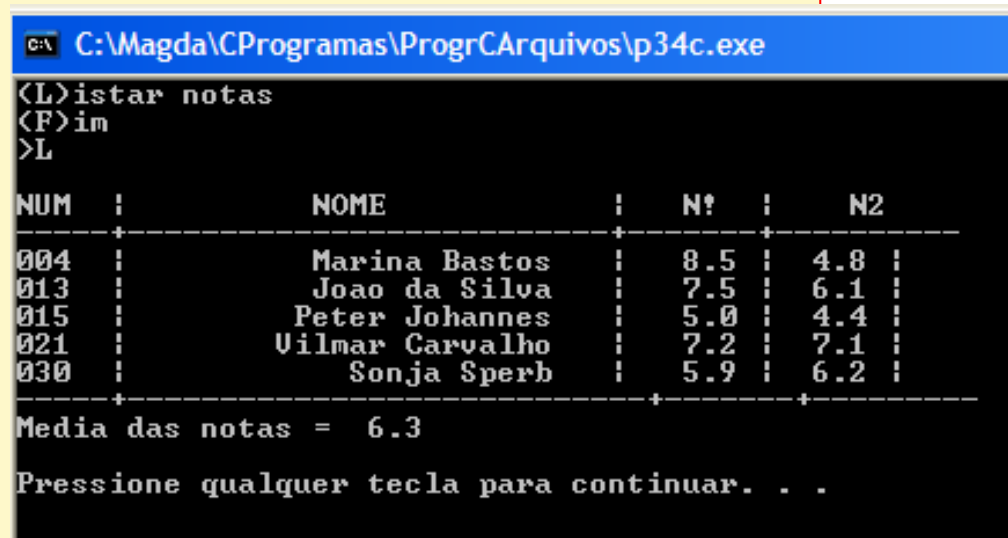
Programa que lê um arquivo texto e o lista. Arquivo armazena código(int), nome(string) 2 notas(float). Apresenta a média das notas dos alunos.

O programa ao executar lê o arquivo dados.txt:



```
4,Marina Bastos,8.500000,4.800000,
13,Joao da Silva,7.500000,6.100000,
15,Peter Johannes,5.000000,4.400000,
21,Vilmar Carvalho,7.200000,7.100000,
30,Sonja Sperb,5.900000,6.200000,
```

produzindo a saída:



```
C:\Magda\CProgramas\ProgrCARquivos\p34c.exe
<L>istar notas
<F>im
>L

NUM | NOME | N1 | N2
---+---+---+---
004 | Marina Bastos | 8.5 | 4.8
013 | Joao da Silva | 7.5 | 6.1
015 | Peter Johannes | 5.0 | 4.4
021 | Vilmar Carvalho | 7.2 | 7.1
030 | Sonja Sperb | 5.9 | 6.2
---+---+---+---
Media das notas = 6.3
Pressione qualquer tecla para continuar. . .
```

Programa gerador do arquivo dados.txt

```
/*#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX 80
int main ( )
{
    FILE *arq;
    char nome[MAX];
    int num, op, res;
    float n1,n2,media;
    system("color 70");
    if (!(arq = fopen("dados.txt","w")))
    {
        printf("\nErro abertura\n");
        system("pause");
    }
    else
    {
        (...)
```

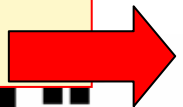
Programa gerador do arquivo dados.txt

```
(...)  
do  
{  
    printf("\nNumero: ");  
    scanf("%d", &num);  
    printf("\nNome: ");  
    fflush(stdin); // limpa o buffer de entrada do teclado  
    fgets(nome, sizeof(nome), stdin);  
    if (nome[strlen(nome) - 1] == '\n')  
        (nome[strlen(nome) - 1] = '\0');  
    fflush(stdin);  
    printf("\nNota 1: ");  
    scanf("%f", &n1);  
    printf("\nNota 2: ");  
    scanf("%f", &n2);  
    // gravacao dos dados em dados.txt, na chamada de fprintf a seguir  
    // observar a gravacao da virgula entre os valores - como delimitador  
    fprintf (arq, "%d,%s,%f,%f,\n", num, nome, n1, n2);  
    printf("\nseguir 1, parar 0\n");  
    scanf("%d",&op );  
}  
while (op); // enquanto usuario quiser continuar  
fclose(arq);  
return 0;  
}
```

Programa adiciona novos alunos ao arquivo dados.txt

```
#include<stdio.h>
#include<stdlib.h>
#include <ctype.h>    // necessario para toupper()
#include<string.h>    // necessario para strtok()
#define MAX 80
char opcao_menu() ;
void listar_notas(int);
void novo_aluno() ;
int main( )
{
    char op;
    do
    {
        op=opcao_menu();
        if (op == 'L')
            listar_notas(MAX);
        if (op == 'N')
            novo_aluno();
        printf("\n");
        system("pause");
    }
    while (op != 'F');
    return 0;
}
```

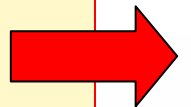
Continua...



Programa adiciona novos alunos ao arquivo dados.txt

```
char opcao_menu( )    //menu com as operações
{
    system("CLS");
    printf("(L)istar notas\n");
    printf("(N)ovo aluno\n");
    printf("(F)im \n");
    printf(">");
    return(toupper(getchar()));
}
void listar_notas(int maxim)
{
    int num, notas;
    float n1,n2,media;
    char *nome;
    char buf[maxim];
    FILE *arq;
    arq = fopen("dados.txt","r");
    if (arq == NULL)
    {
        printf("Erro ao abrir arquivo \n");
        system("pause");
    }
    else
    {
        (...)
    }
}
```

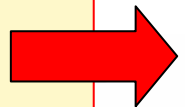
Continua...



Programa adiciona novos alunos ao arquivo dados.txt

```
(...)  
notas=0;  
media=0;  
printf("\n");  
printf("NUM |      NOME      | N1 |  N2  \n");  
printf("-----+-----+-----+-----\n");  
fgets(buf,sizeof(buf),arq);  
while (!feof(arq))  
{  
    num = atoi(strtok(buf,""));  
    nome = strtok(NULL,"");  
    n1 = atof( strtok(NULL,""));  
    n2 = atof( strtok(NULL,""));  
    printf("%03d | %20s | %4.1f | %4.1f \n",num, nome,n1,n2);  
    notas = notas +2;  
    media = media + n1 + n2;  
    fgets(buf,sizeof(buf),arq);  
}  
printf("-----+-----+-----+-----\n");  
media = media / notas;  
printf("Media das notas = %4.1f\n", media);  
fclose(arq);  
}  
}
```

Continua...



Programa adiciona novos alunos ao arquivo dados.txt

```
void novo_aluno()
{
    int num;
    float n1,n2;
    char nome[20];
    FILE *arq;
    arq = fopen("dados.txt","a");
    if (arq ==NULL)
    {
        printf("Erro ao abrir arquivo!\n");
        system("pause");
    }
    else
    {
        printf("\n");
        printf("Digite os dados do novo aluno: \n");
        printf("Numero...");
        scanf("%d",&num);
        printf("Nome....");
        fflush(stdin);
        fgets(nome, sizeof(nome), stdin);
        if (nome[strlen(nome) - 1] == '\n')
            nome[strlen(nome) - 1] = '\0';
        printf(" Nota 1:...");
        scanf("%f",&n1);
        printf(" Nota 2:...");
        scanf("%f",&n2);
        fprintf(arq,"%d,%s,%.1f,%.1f,\n",num,nome,n1,n2);
        fclose(arq);
    }
}
```

```
<L>istar notas
<N>ovo aluno
<F>im
>L
```

NUM	NOME	N1	N2
003	ROSA	1.0	1.0
005	PEDRO	2.0	2.0
009	Maria	8.0	5.0
012	Mariana	3.0	10.0

Media das notas = 4.0

Pressione qualquer tecla para continuar. . . -