


Introdução à Programação

Aula 09 – Arquivos

Edirlei Soares de Lima
<edirlei@iprj.uerj.br>



Funções de Entrada e Saída em Arquivos

- Até agora nós desenvolvemos somente programas que capturam dados de entrada via teclado e exibem seus dados de saída na tela.
- **Problema:**
 - Erros de digitação obrigam o usuário a começar desde o início
- O que fazer quando um programa precisa processar um volume de dados muito grande?

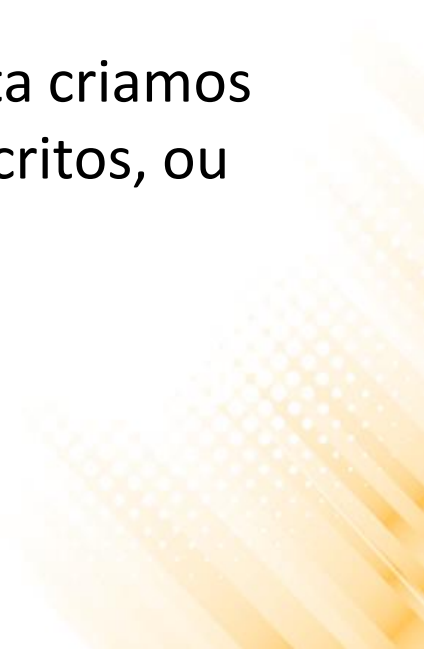
Funções de Entrada e Saída em Arquivos

- A estratégia adequada para se trabalhar com grande volume de dados é fazer uso de **arquivos de dados**.
- Para que programas possam ler e salvar dados em um arquivo, é necessário ter acesso a um **conjunto de funções** que permitam realizar essas operações:
 - Nesse curso, usaremos um conjunto de funções presentes na biblioteca **stdio.h**

Funções de Entrada e Saída em Arquivos

- Um arquivo pode ser visto de duas maneiras: **modo texto** e **modo binário**
 - Neste curso, vamos trabalhar apenas com arquivos no modo texto
- Para que possamos ler ou escrever em um arquivo é necessário que antes o **arquivo esteja aberto**.
- Cada arquivo é **identificado pelo seu nome**, que normalmente é o nome do arquivo mais a extensão: exemplo.txt, cap8.ppt, saida.txt

Funções de Entrada e Saída em Arquivos

- Para abrir um arquivo é necessário definir se o mesmo será utilizado para leitura ou escrita:
 - **Leitura:** para abrirmos um arquivo para leitura é necessário que o arquivo já exista.
 - **Escrita:** quando abrimos um arquivo para escrita criamos um novo arquivo onde dados de saída serão escritos, ou sobrescrevemos um arquivo existente.
- 

Funções de Entrada e Saída em Arquivos

- Para abrir um arquivo, a biblioteca **stdio.h** oferece a função **fopen**.
 - Esta função serve tanto para abrir um arquivo para leitura como para escrita.
 - A função recebe dois parâmetros, o **nome do arquivo** que se deseja abrir e o **modo de abertura**:
 - Leitura: "r" (read)
 - Escrita: "w" (write).
 - Esta função retorna um ponteiro para o tipo FILE.
 - Um ponteiro é um tipo de variável que serve para armazenar endereços de memória.

Declaração e Abertura de um Arquivo

- Declaração de uma variável do tipo ponteiro para **FILE**:

```
FILE *fp;
```

- Abrindo o arquivo para **leitura** de dados:

```
fp = fopen("entrada.txt", "r");
```

- Abrindo o arquivo para **escrita** de dados:

```
fp = fopen("saida.txt", "w");
```

Declaração e Abertura de um Arquivo

- Se a abertura do arquivo não for bem sucedida, a função retorna o valor definido pela constante simbólica **NULL**:

```
fp = fopen("entrada.txt", "r");
if (fp == NULL)
{
    printf("Erro na abertura do arquivo.\n");
    return 1; /* aborta programa */
}
```

- Após realizar as operações de entrada e saída no arquivo, este deve ser **fechado** com o uso da função **fclose**.

```
fclose(fp);
```


Leitura em Arquivo: fscanf

- A principal função para leitura de arquivos chama-se **fscanf**, e é similar ao scanf usado para captura de dados via teclado.

```
int a;  
float b;  
FILE* fp = fopen("entrada.txt", "r");  
. . .  
fscanf(fp, "%d %f", &a, &b);  
. . .  
fclose(fp);
```

- A função fscanf tem como valor de retorno o número de parâmetros que foram lidos com sucesso.

Exemplo de Leitura

- Vamos considerar a existência de um arquivo que armazena **notas** que os alunos obtiveram em uma disciplina.
 - A primeira linha contém um número inteiro que informa a **quantidade de notas** armazenadas a seguir, estando **uma nota por linha**.

- Arquivo **notas.txt**:

6
7.5
8.4
9.1
4.0
5.7
4.3

- **Exemplo 1:** O programa deve ler as notas do arquivo e escrever na tela a média dos alunos

```
#include <stdio.h>

int main(void)
{
    int i, n;
    float nota, soma = 0.0;
    FILE *fp ;
    /* abertura do arquivo para leitura */
    fp = fopen ("notas.txt", "r");
    /* teste para verificar se houve algum erro */
    if (fp == NULL)
    {
        printf("Erro na abertura do arquivo.\n");
        return 1; /* aborta programa */
    }
    ...
}
```

(continuação)

```
/* leitura da quantidade de notas no arquivo */
fscanf(fp , "%d", &n);
/* laço para leitura de cada nota */
for (i=0; i<n; i++)
{
    fscanf(fp, "%f ", &nota);
    soma = soma + nota ;
}

/* fechamento do arquivo */
fclose(fp);

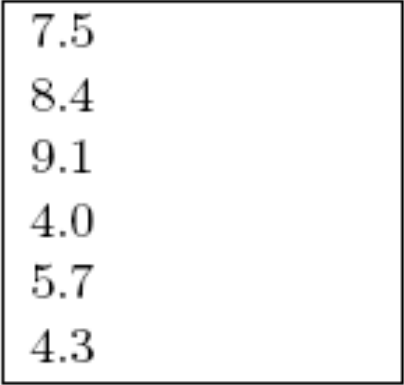
/* cálculo da média e impressão na tela */
printf("Media = %.2f \n", soma / n);
return 0;
}
```

Exemplo de Leitura

- Quando escrevemos programas para processar dados de entrada de um arquivo, procuramos manter o **formato** dos dados presentes no arquivo o **mais flexível possível**.
- O formato do arquivo de notas do **exemplo anterior não é muito flexível** pois se formos adicionar uma nota no arquivo teremos também que atualizar a primeira linha do arquivo, que representa a quantidade de notas.
- Para tornar o formato mais flexível, podemos pensar em **eliminar a informação da primeira linha**, isto é, podemos listar apenas as notas, e o programa deve ser capaz de exibir a média de todas as notas.

```
#include <stdio.h>

int main (void)
{
    int n;
    float nota, soma = 0.0;
    FILE *fp ;
    fp = fopen ("notas.txt", "r");
    n = 0;
    /* laço para leitura de cada nota */
    while (fscanf(fp, "%f", &nota) == 1)
    {
        soma = soma + nota;
        n++;
    }
    fclose(fp);
    if (n > 0)
    {
        printf("Media = %.2f \n", soma / n);
    }
    return 0;
}
```



7.5
8.4
9.1
4.0
5.7
4.3

O **while** processa as notas do arquivo enquanto o final do arquivo não é alcançado.

Escrita em Arquivo: fprintf

- A principal função para escrita de dados em arquivos chama-se **fprintf**, e é similar ao printf usado para escrita de dados na tela.

```
int a ;
float b ;
FILE* fp = fopen("saida.txt", "w");
. . .
fprintf(fp, "Valores : %d %f \n", a, b);
. . .
fclose(fp);
```

Exemplo de Escrita

- Vamos considerar uma disciplina onde cada aluno faz 3 provas, obtendo três notas que são consideradas para o cálculo da nota final.
- Considere que as notas obtidas para cada aluno estão armazenadas no arquivo “entrada.txt”

7.5	8.5	7.8
8.4	9.2	6.8
9.1	10.0	9.5
4.0	5.2	4.6
5.7	3.4	4.3
4.3	6.0	5.8

Exemplo de Escrita

- O objetivo é desenvolver um programa que **processe as notas do arquivo** “entrada.txt”.
- Para cada aluno, o programa deve **calcular a sua nota final** e verificar se o aluno foi **aprovado ou reprovado**.
- O programa deve **gerar um arquivo** de saída com o nome “saida.txt”. Neste arquivo de saída, para cada linha do arquivo de entrada, deve-se escrever a linha correspondente com a **nota final do aluno e sua situação**:
 - A se o aluno foi aprovado ($\text{media} \geq 5.0$)
 - R se o aluno foi reprovado.

```
#include <stdio.h>

int main (void)
{
    float p1, p2, p3, media;
    FILE *ent, *sai;
    ent = fopen("entrada.txt", "r");
    sai = fopen("saida.txt", "w");
    while (fscanf(ent, "%f %f %f", &p1, &p2, &p3) == 3)
    {
        media = (p1 + p2 + p3) / 3;
        fprintf(sai, "%.1f ", media);
        if (media >= 5.0)
            fprintf(sai, "A\n");
        else
            fprintf(sai, "R\n");
    }
    fclose(ent);
    fclose(sai);
    return 0;
}
```

Exercícios

Lista de Exercícios 07 - Arquivos

<http://www.inf.puc-rio.br/~elima/intro-prog/>

