Aula 14 - Programação de Computadores I

Engenharia Elétrica

Prof. Marcos Matsuo

marcos.matsuo@ifsc.edu.br

Operações com arquivos

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Arquivos em C

- Nesta aula, veremos como programas em C podem criar, abrir e fechar arquivos armazenados na memória secundária (por exemplo, HD) de um computador.
- Em C, um arquivo representa uma sequência de bytes.
- A linguagem C possui funções (pertencentes a biblioteca stdio) que permitem a leitura e escrita de arquivos armazenados no computador.

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Criando arquivos de texto

- **Objetivo**: criar um arquivo de texto para ser salvo no computador.
- Principais funções:
  - fopen: utilizada para abrir/criar um arquivo.
  - fclose: utilizada para fechar um arquivo.
  - fprintf: escreve uma string formatada no arquivo de texto
- FILE é um objeto que contém todas as informações necessárias para controlar o arquivo.

Programa em C

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

int main()
{
    FILE *fp;
    char nome[] = "Fulano";
    int idade   = 25;

    fp = fopen("arquivo_teste.txt","w");

    fprintf(fp, "Arquivo de texto\n");
    fprintf(fp, "Nome: %s\n", nome);
    fprintf(fp, "Idade: %d\n", idade);

    fclose(fp);
    return 0;
}
```

Criando arquivos de texto

- A função fflush é utilizada para limpar o buffer do dispositivo padrão de entrada (stdin – teclado).

```c
int main()
{
    FILE *fp;
    char titulo[31];
    int regnum, k = 1;;
    float preco;
    fp = fopen("livros.txt", "w");

    while(1)
    {
        printf("---Registro do livro %d---",k);
        printf("\nDigite o titulo: ");

        fflush(stdin);
        scanf("%30[^\n]s", titulo);

        if (strlen(titulo)<=1) break;

        printf("\nDigite o registro: ");
        scanf("%d",&regnum);
        printf("\nDigite o preco: ");
        scanf("%f",&preco);

        fprintf(fp, "%s\t%d\t%f\n",titulo, regnum, preco);
        k++;
    }
    fclose(fp);
    return 0;
}
```

Criando arquivos de texto

- Complementando o arquivo anterior, abrindo com append ("a").

```c
int main()
{
    FILE *fp;
    char titulo[31];
    int regnum, k = 1;;
    float preco;
    fp = fopen("livros.txt", "a");

    while(1)
    {
        printf("---Registro do livro %d---",k);
        printf("\nDigite o titulo: ");

        fflush(stdin);
        scanf("%30[^\n]s", titulo);

        if (strlen(titulo)<=1) break;

        printf("\nDigite o registro: ");
        scanf("%d",&regnum);
        printf("\nDigite o preco: ");
        scanf("%f",&preco);

        fprintf(fp, "%s\t%d\t%f\n",titulo, regnum, preco);
        k++;
    }
    fclose(fp);
    return 0;
}
```

Criando arquivos de texto

- Criando um arquivo e escrevendo caractere por caractere.
- Nova função:
  - fputc → escreve um caractere em um arquivo

```c
#include <stdio.h>

int main()
{
    FILE *fp;
    fp = fopen("teste_escrita.txt","w");
    char c;

    printf("Digite o que deseja gravar no arquivo e pressione enter:\n");

    do {
        fflush(stdin);
        c = getc(stdin);
        fputc(c,fp);

    } while (c != '\n');

    fclose(fp);

    return 0;
}
```

Criando arquivos de texto

- Criando um arquivo e escrevendo linha por linha.
- Nova função:
  - fputs → escreve uma *string* em um arquivo

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

int main()
{
    FILE *fp;
    fp = fopen("teste_escrita2.txt","w");
    setlocale(LC_ALL,"portuguese");

    fputs("Programação C\n",fp);
    fputs("IFSC\n",fp);
    fputs("Engenharia Elétrica",fp);

    fclose(fp);
    return 0;
}
```

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Abrindo arquivos de texto

- Lendo dados formatados.

Arquivo gerado através do programa do slide 7.

Arquivo "livros.txt"

```
Livro1      102     70.5
Livro2      321     63.25
```

```c
#include <stdio.h>

int main()
{
    FILE *fp;
    char titulo[31];
    int regnum;
    float preco;
    fp = fopen("livros.txt","r");

    while(fscanf(fp, "%30s\t%d\t%f\n", titulo, &regnum, &preco) != EOF)
    {
        printf("Titulo: %s   Registro: %d   Preco:%f\n", titulo, regnum, preco);
    }
    fclose(fp);

    return 0;
}
```

Abrindo arquivos de texto

- Objetivo: ler o conteúdo de um arquivo salvo no computador (caractere por caractere).
- Usando a macro EOF na condição da estrutura de repetição while.
- Exemplo:
  - Criar um arquivo com o conteúdo mostrado no quadro abaixo, salvar com o nome “teste.txt” na mesma pasta do projeto do CodeBlocks.

Programacao de computadores I
Curso de Eng. Eletrica

```c
#include <stdio.h>

int main()
{
    FILE *fp;
    char ch;
    fp = fopen("teste.txt","r");

    while((ch = fgetc(fp)) != EOF)
    {
        printf("%c",ch);
    }

    fclose(fp);

    return 0;
}
```

Abrindo arquivos de texto

- Objetivo: ler o conteúdo de um arquivo salvo no computador (caractere por caractere).
- Usando a função `feof()` na condição do while
- Exemplo:
  - Criar um arquivo com o conteúdo mostrado no quadro abaixo, salvar com o nome “teste.txt” na mesma pasta do projeto do CodeBlocks.

Programacao de computadores I
Curso de Eng. Eletrica

```c
#include <stdio.h>

int main()
{
    FILE *fp;
    char ch;
    fp = fopen("teste.txt","r");

    while(!feof(fp))
    {
        ch = fgetc(fp);
        printf("%c",ch);
    }

    fclose(fp);

    return 0;
}
```

Abrindo arquivos de texto

- Função exit() para tratar erro na abertura do arquivo.

```c
#include <stdio.h>

int main()
{
    FILE *fp;
    char ch;
    fp = fopen("abc.txt","r");

    if (fp == NULL)
    {
        printf("Nao foi possivel abrir o arquivo");
        exit(1);
    }

    while(!feof(fp))
    {
        ch = fgetc(fp);
        printf("%c",ch);
    }
    fclose(fp);

    system("pause");
    return 0;
}
```

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Exemplo 1 – Arquivo de texto de acesso sequencial

Escrita de arquivo de texto

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

int main()
{
    int conta;
    char nome[30];
    float saldo;
    FILE *cptr;

    if ((cptr = fopen("clientes.txt","w")) == NULL) {
        printf("Arquivo nao pode ser criado.");

    } else {

        printf("Digite a conta, o nome e o saldo\n");
        printf("(digite EOF para encerrar a entrada de dados).\n"); // Ctrl+Z
        printf("? ");
        scanf("%d%s%f",&conta, nome, &saldo);

        while (!feof(stdin))
        {
            fprintf(cptr,"%d %s %.2f\n", conta, nome, saldo);
            printf("? ");
            scanf("%d%s%f", &conta, nome, &saldo);
        }
        fclose(cptr);
    }

    system("pause");
    return 0;
}
```

Exemplo 1 – Arquivo de texto de acesso sequencial

Leitura de arquivo de texto

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

int main()
{
    int conta;
    char nome[30];
    float saldo;
    FILE *cptr;

    if ((cptr = fopen("clientes.txt","r")) == NULL) {
        printf("Arquivo nao pode ser aberto.\n");
    } else {

        printf("%-10s%-13s%s\n", "Conta", "Nome", "Saldo");
        fscanf(cptr,"%d%s%f",&conta, nome, &saldo);

        while (!feof(cptr)) {
            printf("%-10d%-13s%7.2f\n", conta, nome, saldo);
            fscanf(cptr,"%d%s%f", &conta, nome, &saldo);
        }
        fclose(cptr);
    }

    system("pause");
    return 0;
}
```

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercícios
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Exercício 1

- Escreva um programa em C para armazenar informações dos alunos de uma classe dentro de um arquivo texto.
- Para cada estudante as seguintes informações devem ser armazenadas: número de matrícula, nome, sobrenome, nota final na disciplina de Programação de Computadores I.
- Organize o programa em funções.
- Peça para o usuário fornecer inicialmente o número de estudantes que serão cadastrados.
- O nome do arquivo a ser gravado deve ser fornecido pelo usuário.

Exercício 2

- Escreva um programa para ler as informações dos alunos no arquivo de texto criado pelo programa do Exercício 1.
- As informações dos alunos dever ser armazenadas em vetor de estruturas.
- Organize o programa em funções.
- O nome do arquivo a ser lido deve ser fornecido pelo usuário.
- Após a leitura do arquivo, o programa deve apresentar um menu com as opções:

  <1> Média da turma

  <2> Quantos alunos ficaram com nota maior do que 6

  <3> Identicar qual aluno tirou a maior nota

  <0> Sair do programa

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Escrita de arquivos binários

- Podemos usar as funções `fwrite` para escrever blocos de dados → número de bytes definido pelo programador.
- Escrita em modo binário "wb".

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

int main()
{
    FILE *fp;

    float num = 12.23;
    int i = 101;
    char ch = 'a';

    if ((fp=fopen("teste.bin", "wb")) == NULL){
        printf("O arquivo nao pode ser criado.\n");
        exit(1);
    }

    fwrite(&num, sizeof(float), 1, fp);
    fwrite(&i, sizeof(int), 1, fp);
    fwrite(&ch, sizeof(char), 1, fp);

    fclose(fp);

    return 0;
}
```

Leitura de arquivos binários

- Podemos usar as funções `fread` para ler blocos de dados → número de bytes definido pelo programador.
- Leitura em modo binário "rb".

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

int main()
{
    FILE *fp;

    float num;
    int i;
    char ch;

    if ((fp=fopen("teste.bin", "rb")) == NULL){
        printf("O arquivo nao pode ser aberto.\n");
        exit(1);
    }

    fread(&num, sizeof(float), 1, fp);
    fread(&i, sizeof(int), 1, fp);
    fread(&ch, sizeof(char), 1, fp);

    printf("%f %d %c", num, i, ch);

    fclose(fp);

    return 0;
}
```

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Exemplo 2 – Arquivo de acesso sequencial

Escrita de arquivos binários

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

struct DadosCliente {
   char nome[50];
   char telefone[15];
};

int main()
{
    setlocale(LC_ALL, "Portuguese");

    struct DadosCliente cliente;
    int num_clientes, k;

    FILE *cptr;

    if ((cptr = fopen("clientes.dat", "wb")) == NULL) {
        printf("Arquivo nao pode ser criado.");

    } else {

        printf("Quantos clientes serão cadastrados? ");
        scanf("%d", &num_clientes);

        for (k=1; k<num_clientes; k++) {

            printf("Cliente %d:\n", k);

            printf("Digite o nome: ");
            fflush(stdin);
            scanf("%49[^\n]s", cliente.nome);
```

*continua no próximo slide.

Exemplo 2 – Arquivo de acesso sequencial

Escrita de arquivos binários

```
            printf("Telefone: ");
            fflush(stdin);
            scanf("%14[^\n]s", cliente.telefone);

            fwrite(&cliente, sizeof(struct DadosCliente), 1, cptr);
        }

        fclose(cptr);
    }

    system("pause");
    return 0;
```

Exemplo 2 – Arquivo de acesso sequencial

Leitura de arquivos binários

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

struct DadosCliente {
    char nome[50];
    char telefone[15];
};

int main()
{
    setlocale(LC_ALL, "Portuguese");

    struct DadosCliente cliente;
    FILE *cptr;

    if ((cptr = fopen("clientes.dat","rb")) == NULL) {
        printf("Arquivo nao pode ser aberto.\n");
    } else {

        fread(&cliente, sizeof(struct DadosCliente), 1, cptr);

        while (!feof(cptr))
        {
            printf("Nome: %s\n", cliente.nome);
            printf("Telefone: %s\n\n", cliente.telefone);
            fread(&cliente, sizeof(struct DadosCliente), 1, cptr);
        }

        fclose(cptr);
    }

    system("pause");
    return 0;
}
```

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Função fseek

- Até agora vimos como acessar um arquivo de forma sequencial.
- Porém a linguagem C permite realizar operações de escrita e leitura “pulando” uma determinada quantidade de bytes. Tal operação é realizada através da função fseek().

```
int fseek(FILE *fp, long numbytes, int origem);
```

Fseek define a nova posição de "fp", a qual é calculada adicionando o valor de offset (segundo argumento da função) ao endereço de referência.

Argumentos de entradas:

fp: ponteiro para o arquivo.

numbytes: é o total de bytes a ser pulado a partir do endereço de origem.

Origem: endereço de origem a partir do qual os bytes serão pulados.

- 0 = SEEK_SET -> início do arquivo;
- 1 = SEEK_CUR -> posição atual;
- 2 = SEEK_END -> fim do arquivo.

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Exemplo 3 – Arquivo binário de acesso aleatório

- Veja arquivo PGR22003 - Aula 14 – Exemplo 3.pdf (disponível no Moodle).

- Arquivos em C
- Criando arquivos de texto
- Abrindo arquivos de texto
- Exemplo 1 – Arquivo de texto de acesso sequencial
- Exercício
- Leitura e escrita de arquivos binários
- Exemplo 2 – Arquivo binário de acesso sequencial
- Função fseek
- Exemplo 3 – Arquivo binário de acesso aleatório
- Exercício

Exercício

- Escrever um programa para registro e consulta dos dados dos alunos de uma turma.
- Para cada aluno devem ser armazenados o número do registro, nome e nota.
- O programa deve ter um menu com duas opções: escrita ou consulta.
- Dentro de escrita, apresentar as opções de novo arquivo ou complementar arquivo existente.
- O nome do arquivo deve ser fornecido pelo usuário via teclado.
- Dentro de consulta, devem ser disponibilizadas as opções: consulta de registro, média geral, menor nota e maior nota.
- Os dados devem ser escritos e lidos no modo binário.
- Organize o programa em funções.

Aula 14 - Programação de Computadores I

Engenharia Elétrica

Prof. Marcos Matsuo

marcos.matsuo@ifsc.edu.br

Funções comumente usadas

Material extra

- fopen → abre um arquivo (tanto para leitura quanto para escrita)
- fclose → fecha um arquivo
- fprintf → generaliza o ideia do printf do console para arquivos em geral
- fscanf → generaliza a ideia do scanf do console para arquivos em geral
- feof → retorna verdadeiro se o fim do arquivo for atingido
- fputc → escreve um caractere em um arquivo
- fgetc → lê um caractere de um arquivo
- fputs → escreve uma *string* em um arquivo
- fgets → lê uma *string* de um arquivo
- rewind → coloca o indicador de posição no início do arquivo
- fflush → limpa o *buffer*
- fseek → posiciona o arquivo em um byte específico
- fread → lê um bloco de dados
- Fwrite → escreve um bloco de dados

Especificações da função fopen

Material extra

- r → abre um arquivo texto para leitura
- w → cria um arquivo texto para escrita
- a → anexa a um arquivo texto
- rb → abre um arquivo binário para leitura
- wb → cria um arquivo binário para escrita
- ab → anexa a um arquivo binário
- r+ → abre um arquivo texto para escrita/leitura
- w+ → cria um arquivo texto para escrita/leitura
- a+ → anexa ou cria um arquivo texto para escrita/leitura
- r+b → abre um arquivo binário para escrita/leitura
- w+b → cria um arquivo binário para escrita/leitura
- a+b → anexa ou cria um arquivo binário para escrita/leitura