

## Arquivos

Introdução à Ciência da Computação I

Prof. Denis F. Wolf

## Entrada e Saída de Dados

- Em C não existem comandos de Entrada e Saída, sendo estas tarefas executadas por funções especialmente criadas para esta finalidade e armazenadas em bibliotecas específicas.
- Dados podem ser manipulados em dois diferentes tipos de fluxos:
  - fluxo de texto
  - fluxo binário.

## Fluxos (streams)

- **Fluxo de texto:** composto por uma seqüência de caracteres, que pode ou não ser dividida em linhas terminadas por um caractere de final de linha.
  - Um detalhe que deve ser considerado é que na última linha não é obrigatório o caractere de fim de linha.
- **Fluxo binário:** composto por uma seqüência de bytes lidos, sem tradução, diretamente do dispositivo externo.
  - Não ocorre nenhuma tradução e existe uma correspondência um para um entre os dados do dispositivo e os que estão no fluxo.

## Arquivos

- Os arquivos são entendidos como *streams* cujos dados estão guardados em um dispositivo de armazenamento secundário
- Operações comuns em arquivos são:
  - abertura e fechamento de arquivos;
  - apagar um arquivo;
  - leitura e escrita de um caractere;
  - indicação de que o fim do arquivo foi atingido;
  - posicionar o arquivo em um ponto determinado
- Importante:
  - Ao final das operações necessárias o programa deve fechar o arquivo.
  - Quando um programa é encerrado todos os arquivos associados são fechados automaticamente e os conteúdos dos buffers são descarregados para o dispositivo externo.

## Funções da biblioteca *stdio.h*

Função	Descrição
<i>fopen()</i>	Abre um arquivo
<i>fclose()</i>	Fecha um arquivo
<i>fseek()</i>	Posiciona o ponteiro do arquivo
<i>feof()</i>	Retorna VERDADE se chegou ao fim do arquivo
<i>fflush()</i>	Descarrega o buffer associado com o arquivo
<i>fprintf()</i>	Equivalente a <i>printf()</i> , usando stream
<i>fscanf()</i>	Equivalente a <i>scanf()</i> , usando stream
<i>fgets()</i>	Equivalente a <i>gets()</i> , usando stream

## Abrindo e fechando um arquivo

```
FILE *arq;
```

Declara um ponteiro para arquivo (FILE)

```
arq = fopen("nome_do_arq", modo);
```

Abre/cria arquivo. Retorna NULL se ocorrer algum erro.

```
fclose(arq);
```

Fecha um arquivo


## Abrindo e fechando um arquivo

```
void main ()
{
    FILE *arq;
    arq=fopen("dados.txt", "w+")
    if (arq==NULL)
    {
        printf("Erro na abertura do      arquivo");
        exit(0);
    }
    fclose(arq);
}
```

## Modos de abertura

- r: abre arquivo para leitura
- w: cria arquivo para escrita
- a: abre arquivo para escrita (anexando)
- r+: abre arquivo para leitura e escrita
- w+: cria/sobrescreve arquivo para leitura e escrita, apagando os dados existentes anteriormente
- a+: cria/abre arquivo para leitura e escrita (anexando)

## Funções para manipulação de arquivo

- fprintf(arq, "string", variáveis) – escreve no arquivo (semelhante ao printf)
  - fscanf(arq, "string", variáveis) – le do arquivo (semelhante ao scanf)
  - fgets(\*dados, tamanho dos dados, arq) – le do arquivo (semelhante ao gets)
- 

## Funções para manipulação de arquivo

- fseek(arq, posição a ser buscada, posição de referencia) – busca posição em arquivo aberto  
posição a ser buscada: em bytes  
posição de referência: SEEK\_SET para início do arquivo  
SEEK\_CUR para posição atual  
SEEK\_END para fim do arquivo
- feof(arq) – retorna zero se ponteiro não aponta para o fim do arquivo
- fflush(arq) – descarrega o buffer no arquivo

## Exemplo: escrita de texto no arquivo

```
#include <stdio.h>

void main ()
{
    FILE *arq;
    int i, n;

    if((arq=fopen("test.txt", "w+"))==NULL)
        printf("\nErro abrindo arquivo.\n");

    for(i=0; i<5; i++) {
        scanf("%d", &n);
        fprintf(arq, "%d\n", n);
    }

    fclose(arq);
}
```

## Exemplo: leitura de texto no arquivo

```
#include <stdio.h>

void main ()
{
    FILE *arq;
    int i, n;

    if((arq=fopen("test.txt", "r+"))==NULL)
        printf("\nErro abrindo arquivo.\n");

    for(i=0; i<5; i++) {
        fscanf(arq, "%d", &n);
        printf("%d\n", n);
    }

    fclose(arq);
    system("PAUSE");
}
```

## Exercícios

- 1) Criar um programa que lê uma frase e escreve em um arquivo de texto. Após a escrita no arquivo, o programa deve ler o conteúdo do arquivo e exibi-lo na tela.
- 2) Criar um programa que lê uma palavra digitada, abre um arquivo de texto já existente e verifica se a palavra está contida no arquivo.