

# MATA54 - Estruturas de Dados e Algoritmos II

## Operações de Acesso a Arquivos

Flávio Assis

IC - Instituto de Computação

v. 1.0

Salvador, agosto de 2021

# Tipos Básicos de Arquivos

## 1. arquivos de texto:

- ▶ elementos básicos são **caracteres**
- ▶ estruturados como uma sequência de linhas, com marcador de fim de linha

## 2. arquivos binários:

- ▶ elementos básicos são **bytes**
- ▶ formas específicas de armazenar dados no arquivo e de acessá-los definem diferentes tipos de organização

## 3. Exemplo de diferença entre estes tipos de arquivos: armazenamento do número 250

- ▶ Em um arquivo texto: seriam armazenados três **caracteres**: '2', '5', '0' (cada um, por exemplo, ocupando um *byte*)
- ▶ Em um arquivo binário: seria armazenada uma representação binária do inteiro 250: por exemplo: 1111 1010 (em um computador hipotético que armazena inteiros em um *byte*)

# Acesso a Arquivos Binários

- ▶ Linguagens de programação oferecem abstrações para acesso a arquivos
- ▶ Tipicamente o conteúdo dos arquivos é apresentado como uma sequência de *bytes*

Endereço ( <i>byte offset</i> )	Byte
0	byte
1	byte
2	byte
3	byte
4	byte
...	

# Operações Básicas sobre Arquivos

## Abrir um arquivo:

- ▶ Antes de ser acessado um arquivo tem que ser aberto

- ▶ Exemplo em C:

```
FILE *f;
```

```
f = fopen("dados", "r")
```

- ▶ Indica-se o nome do arquivo (primeiro parâmetro) e o **modo** de abertura
- ▶ O modo de abertura indica o tipo de acesso que se quer fazer ao arquivo (leitura, escrita, etc.)

# Operações Básicas sobre Arquivos

## Abrir um arquivo (cont):

- ▶ O sistema operacional:
  - ▶ verifica se o usuário pode acessar de acordo com o modo solicitado (controle de acesso e concorrência)
  - ▶ registra informações sobre o acesso e aloca estruturas de dados internas para gerenciamento do acesso
  - ▶ associa uma variável para acesso ao arquivo
  - ▶ mantém um **apontador** para a próxima posição a ser acessada do arquivo

# Operações Básicas sobre Arquivos

## Fechar um arquivo:

- ▶ Após o uso de um arquivo, este deve ser fechado
- ▶ Exemplo em C:  
`fclose(f)`
- ▶ A operação tem como parâmetro a variável alocada para acesso ao arquivo
- ▶ Esta operação libera o acesso ao arquivo e os recursos do SO alocados para gerenciamento dos acessos a ele.

# Operações Básicas sobre Arquivos

## Ler dados do arquivo:

### ► Exemplo em C:

```
fread (<end. memória>, <núm. bytes>, <quant>, <arquivo>)
```

<end. memória>: endereço onde serão armazenados os dados lidos

<núm. bytes>: número de bytes a serem lidos

<quant>: quantas vezes a quantidade de bytes deve ser lida

<arquivo>: variável de acesso ao arquivo

Os dados serão lidos a partir da posição do apontador do arquivo.

Registro r;

```
fread (&r, sizeof (Registro), 1, f);
```

# Operações Básicas sobre Arquivos

## Escrever dados no arquivo:

### ► Exemplo em C:

```
fwrite (<end. memória>, <núm. bytes>, <quant>, <arquivo>)
```

<end. memória>: endereço onde estão armazenados os dados a serem escritos no arquivo

<núm. bytes>: número de bytes a serem escritos

<quant>: quantas vezes a quantidade de bytes deve ser escrita

<arquivo>: variável de acesso ao arquivo

Os dados serão escritos a partir da posição onde está o apontador do arquivo.

```
Registro r;
```

```
...
```

```
fwrite (&r, sizeof (Registro), 1, f);
```



# Operações Básicas sobre Arquivos

## Gerenciamento do ponteiro do arquivo:

- ▶ Define uma nova posição para o apontador do arquivo
- ▶ Exemplo em C:

```
fseek(<arquivo>, <deslocamento>, <referência>);
```

<arquivo>: variável de acesso ao arquivo

<deslocamento>: deslocamento a partir da uma referência

<referência>: ponto a ser usado como referência:

- ▶ SEEK\_SET: começo do arquivo
- ▶ SEEK\_CUR: posição atual
- ▶ SEEK\_END: fim do arquivo

```
Registro r;
```

```
...
```

```
fseek(f, 3*sizeof(Registro), SEEK_SET);
```

# Operações Básicas sobre Arquivos

- ▶ Ver exemplo de código em C no [moodle](#)