

# Algoritmos de Programação 2

Professor: José Eurípedes Ferreira de Jesus Filho  
*jeferreirajf@gmail.com*

Universidade Federal de Jataí – UFJ

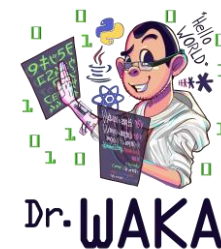
# Introdução

- Vimos alguns **comandos** para trabalharmos com **arquivos de texto**.
- Arquivos de texto geralmente ocupam muito espaço na memória e não são seguros.
- Para facilitar a leitura/escrita em um arquivo de texto é **necessário estabelecer padrões**.



# Arquivos binários

- Arquivos binários.
  - ✓ São os arquivos .bin .
  - ✓ Guardam os dados em formato binário (0 ou 1).
  - ✓ Não são lidos facilmente.
  - ✓ Maior segurança.



# Arquivos binários

- São **fáceis** de serem escritos.
- É possível escrever registros inteiros de uma só vez em um arquivo binário.
- É necessário **controlar a posição** em que se está no arquivo.
- Podemos pensar em um arquivo binário como se **fosse um vetor**.



# Operando com arquivos binários

- Para conseguir operar com arquivos binários é necessário o seguinte:

1. **Criar uma variável do tipo FILE.**

```
FILE *file;
```

2. **Associar o arquivo com a variável.**

```
file = fopen(ENDEREÇO_ARQUIVO, MODOS_ABERTURA);
```

3. **Executar as devidas operações.**

```
fwrite(ENDEREÇO_DADO, TAMANHO_DADO, QUANTIDADE, ARQUIVO);  
fread(ENDEREÇO_VAR, TAMANHO_DADO, QUANTIDADE, ARQUIVO);
```

4. **Fechar o arquivo.**

```
fclose(file);
```



# Operando com arquivos binários

- A função `fopen( )` vincula o arquivo desejado com uma variável do tipo `FILE`.
- A função retorna ou uma variável `FILE` ou `NULL`, caso não consiga abrir o arquivo desejado.

```
FILE *file;  
file = fopen("arquivo.txt", r);
```



# Operando com arquivos binários

- É necessário informar o modo de operação que se deseja abrir o arquivo.
  - ✓ rb para leitura.
  - ✓ wb para escrita em sobreposição.
  - ✓ ab para adição de informação.
  - ✓ rb+ abre para leitura e escrita.
  - ✓ wb+ abre para leitura e escrita.
  - ✓ ab+ abre para leitura e adição de informação.



# Operando com arquivos binários

- Escrevendo um registro em um arquivo:

```
struct s_dado{  
    int n1, n2, n3;  
} dado;
```

```
FILE *file;  
file = fopen("arquivo.txt", "wb");  
fwrite(&dado, sizeof(dado), 1, file);  
fclose(file);
```





# Operando com arquivos binários

- Lendo um registro de um arquivo:

```
struct s_dado{  
    int n1, n2, n3;  
} dado;
```

```
FILE *file;  
file = fopen("arquivo.txt", "rb");  
fread(&dado, sizeof(dado), 1, file);  
fclose(file);
```



# Operando com arquivos binários

- Quando se tem **mais de um dado gravado em um arquivo**, podemos posicionar o cursor para pegarmos exatamente o dado que queremos dentro do arquivo. Para isso, usamos `fseek()`.

`fseek(ARQUIVO, DESLOCAMENTO, INICIO)`

- Para o início, temos as seguintes opções:
  - ✓ `SEEK_SET`: Começa do início do arquivo.
  - ✓ `SEEK_END`: Começa do final do arquivo.
  - ✓ `SEEK_CUR`: Inicia da posição atual.



# Operando com arquivos binários

- Lendo um registro na última posição de um arquivo:

```
struct s_dado{  
    int n1, n2, n3;  
} dado;
```

```
FILE *file;  
file = fopen("arquivo.txt", "rb");  
fseek(file, -1*sizeof(struct s_dado), SEEK_END);  
fread(&dado, sizeof(dado), 1, file);  
fclose(file);
```



# Operando com arquivos binários

- A função `fclose()` desvincula a variável do tipo `FILE` com o arquivo.

```
FILE *file;  
file = fopen("arquivo.txt", "rb");  
fseek(file, -1*sizeof(struct threeNum), SEEK_END);  
fread(&dado, sizeof(dado), 1, file);  
fclose(file);
```

- É **MUITO IMPORTANTE** sempre fechar o arquivo após operar com ele.



# Informações adicionais

- Geralmente, os sistemas operacionais permitem infinitos acessos a um único arquivo desde que esses acessos sejam para leitura.
- Somente **UM ÚNICO** acesso é permitido caso o arquivo seja aberto para escrita.
- É uma boa prática de programação e de design de software sempre abrir uma conexão com um arquivo utilizando o modo correto.



# Resumo

- **ARQUIVOS**

- ✓ Sequência de bytes na memória não volátil.
- ✓ O acesso aos bytes de um arquivo é feito de maneira sequencial.

- **TIPOS DE ARQUIVOS**

- ✓ Texto.
- ✓ Binário.

- **OPERAÇÕES COM ARQUIVOS BINÁRIOS**

- ✓ Necessário uma variável do tipo FILE para a comunicação entre o programa e o arquivo.
- ✓ Função `fopen()` para fazer a associação.
- ✓ Funções `fwrite()`, `fseek()`, `fread()`, etc, para fazer operações nos arquivos.
- ✓ Função `fclose()` para desfazer a associação.

