

**MC102 - Algoritmos e programação de computadores**

# **Aula 26: Arquivos binários**

# Motivação

- Variáveis `int` ou `float` têm tamanho fixo (por exemplo, 4 bytes) na memória
- Representação em texto precisa de um número variável de dígitos (10, 5.673, 100.340).
- Armazenando em arquivos análogo ao utilizado em memória permite
  - reduzir o tamanho do arquivo
  - permitir busca não sequencial

## fread e fwrite

```
size_t fread(void *ptr, size_t size,  
             size_t nmemb, FILE *stream);
```

```
size_t fwrite(const void *ptr, size_t size,  
             size_t nmemb, FILE *stream);
```

- Permitem a leitura e escrita de blocos de dados.
- Veja os códigos `le_vetor.c` e `cp.c`

# Acesso não seqüencial

```
int fseek(FILE *stream, long offset, int whence);
```

- Altera a posição de leitura/escrita no arquivo
- O deslocamento pode ser relativo ao
  - início do arquivo (SEEK\_SET)
  - ponto atual (SEEK\_CUR)
  - final do arquivo (SEEK\_END)
- Veja o código: `fseek.c`

# Registros

- Um arquivo pode armazenar registros (como um banco de dados)
- O acesso a cada registro pode ser direto (comando `fseek`)
- Veja e complete o código: `registro.c`

# Exercício

## Merge

- Escreva um programa que leia dois arquivos de inteiros ordenados e escreva um arquivo cuja saída é um único arquivo ordenado.
  - Vale a pena colocar o conteúdo dos arquivos de entrada em dois vetores?
  - Escreva duas versões deste programa, uma para arquivos texto e outra para arquivos binários.

# Exercício

## Ordenação

- Escreva um programa que lê uma série de linhas de um arquivo texto e escreve um arquivo contendo estas linhas ordenadas.
  - Como você faria para armazenar estas linhas em memória?
  - Qual algoritmo de ordenação você acha mais adequado?