Programação de Computadores I

Arquivos

Arquivo

- Um arquivo é uma coleção de dados organizados e armazenados em um dispositivo, como o disco rígido (HD), SSD ou pen drive.
- No computador, tudo o que não está na memória RAM (que apaga quando o PC desliga) precisa ser salvo como um arquivo para persistir. Para o sistema operacional e para a linguagem C, um arquivo é visto apenas como uma sequência de bytes (0s e 1s) guardada sob um nome específico.

Tipos de Arquivos

- Eles são divididos em duas grandes categorias:
  - Arquivos de Texto: Contêm caracteres legíveis por humanos (letras, números e símbolos). Exemplos: .txt, .c, .html. Cada byte representa um caractere de acordo com uma tabela (como ASCII ou UTF-8).
  - Arquivos Binários: Contêm dados em formato de máquina que não fazem sentido se abertos em um bloco de notas. Exemplos: .exe, .png, .mp3, .pdf. Eles exigem programas específicos para interpretar seus bytes corretamente.

Como o Sistema Operacional Identifica um Arquivo?

- Nome: A identificação principal (ex: dados).
- Extensão: As letras após o ponto (ex: .txt) que indicam o formato e qual programa deve abri-lo.
- Caminho (Path): A localização exata dele nas pastas do sistema (ex: C:\Projetos\programa.c)

Passos básicos para acessar um arquivo

1- Declarar um ponteiro para FILE:

```
FILE *arq;
```

2- Abrir o arquivo com fopen():

A função fopen recebe o caminho/nome do arquivo e o modo de abertura. Sempre verifique se o ponteiro é diferente de NULL para garantir que o arquivo foi aberto corretamente:

```
arq=fopen(“exemplo.txt”,”r”); // “r” para leitura (read)
if (arq==NULL){
    printf(“Erro ao abrir o arquivo!\n”);
}
```

Passos básicos para acessar um arquivo

3 -Realizar as operações de leitura ou escrita:

- Leitura de linha inteira: Use fgets().
- Leitura formatada: Use fscanf().
- Leitura caractere por caractere: Use fgetc().
- Escrita de texto: Use fprintf() ou fputc()

Passos básicos para acessar um arquivo

4 - Fechar o arquivo com fclose():

- Sempre feche o arquivo após terminar o uso para liberar os recursos do sistema:

```
fclose(arq);
```

Modos de Abertura de Arquivo

- Os principais modos de abertura de arquivos usando fopen :
  - "r": Abre para leitura (o arquivo deve existir).
  - "w": Abre para escrita (cria um arquivo novo ou apaga o conteúdo do arquivo existente).
  - "a": Abre para adição (append, escreve no final do arquivo sem apagar o que já está lá).
  - "r+", "w+", "a+": Modos que permitem leitura e escrita simultâneas

Instruções de Escrita

- fprintf() — Escrita Formatada
  - É a função mais versátil. Funciona exatamente como o printf, mas o primeiro argumento é o ponteiro do arquivo.
  - Sintaxe:

    fprintf(ponteiro_do_arquivo, "texto com formatadores", variaveis);
  - Exemplo:

    fprintf(arq, "Nome: %s Telefone: %ld\n", nome, telefone);

Instruções de Escrita

- fputs() — Escrita de Strings
  - Escreve uma string inteira de uma vez. Não aceita formatadores (como %d ou %s).
  - Sintaxe:

    fputs(string, ponteiro_do_arquivo);
  - Exemplo:

    fputs(“Agenda Telefônica.\n”,arq);

Instruções de Leitura

- fgets() — Leitura de Linha por Linha
  - A forma mais segura e recomendada para ler textos. Ela lê uma linha inteira até encontrar uma quebra de linha (\n) ou atingir o limite de tamanho definido.
  - Sintaxe:

    fgets(variavel_string, tamanho_limite, ponteiro_do_arquivo);
  - Exemplo:

    fgets(linha,100,arq); // Lê até 99 caracteres ou até fim da linha

Instruções de Leitura

- fscanf() — Leitura Formatada
  - Funciona como o scanf, ideal para arquivos que possuem dados organizados em colunas ou padrões fixos (como números separados por espaço).
  - Sintaxe:

    ```
    fscanf(ponteiro_do_arquivo, "formatadores", &variaveis);
    ```

  - Exemplo:

    ```
    fscanf(arq, "%s %ld", &nome, &telefone);
    ```