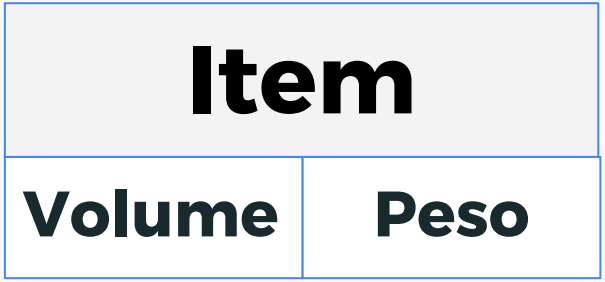
Introdução a Programação

Revisão

Structs & Unions

Estrutura

```
struct Item {
    float volume;
    unsigned int peso;
};
```



União

```
union Item {
    float volume;
    unsigned int peso;
};
```

Manipulação de Arquivos

Motivação

- Armazenamento permanente de dados;
- Grande quantidade dados pode ser armazenada;
- Acesso concorrente.

Tipos de Arquivos

- Arquivos de texto
  - Exemplos: documentos de texto, código fonte C, páginas XHTML.

- Arquivos binários
  - Exemplos: Executáveis, documentos do Word, arquivos compactados.

Streams

- Estrutura do tipo FILE;

- Biblioteca <stdio.h>.

```
FILE *stream;
```

Abrindo um Arquivo

- Função **fopen()** possui dois parâmetros:
  - nome do arquivo e,
  - modo de abertura.

```
FILE *fopen(const char *nome, const char *modo)
```

Modo de Abertura

- “r” - Abre um arquivo existente apenas para leitura em modo de texto.
- “w” - Cria um arquivo apenas para escrita em modo de texto. Se o arquivo já existir, seu conteúdo será destruído.
- “a” - Abre um arquivo existente em modo de texto para acréscimo, i.e., com escrita permitida apenas ao final do arquivo. Se o arquivo com o nome especificado não existir, um arquivo com esse nome será criado.

Modo de Abertura

- "r+" Abre um arquivo existente para leitura e escrita em modo de texto.
- "w+" Cria um arquivo para leitura e escrita em modo de texto. Se o arquivo já existir, seu conteúdo será destruído.
- "a+" Abre um arquivo existente ou cria um arquivo em modo de texto para leitura e acréscimo. Podem-se ler dados em qualquer parte do arquivo, mas eles podem ser escritos apenas ao final do arquivo.

Fechando um Arquivo

- Função **fclose()** possui um parâmetro:
  - ponteiro associado ao arquivo aberto por **fopen()**.

- Retorna 0, caso seja encerrado com sucesso.

```
int fclose(FILE *stream)
```

Exemplo

```c
#include <stdio.h>
int main(void)
{
  FILE *stream;
  /* Tenta abrir o arquivo para leitura */
  stream = fopen("Inexistente.txt", "r");
  /* Verifica se a abertura foi bem sucedida */
  if (stream == NULL) { /* Abertura falhou */
    printf( "\nO arquivo nao pode ser aberto. "
    "\nO programa sera' encerrado.\n" );
    return 1;
  } else { /* Abertura foi OK */
    printf("\nArquivo aberto com sucesso\n");
    /* Processa o arquivo ...*/
  }
  fclose(stream); /* Fecha o arquivo antes de encerrar */
  return 0;
}
```

Ocorrências de Erros em Processamento de Arquivos

- Função **feof()** retorna um valor diferente de zero quando há uma tentativa de leitura além do final do arquivo.

```
int feof(FILE *stream)
```

Ocorrências de Erros em Processamento de Arquivos

- Função **ferror()** retorna um valor diferente de zero após ocorrer algum erro de processamento associado ao stream recebido como parâmetro ou zero, em caso contrário.

```
int ferror(FILE *stream)
```

Ocorrências de Erros em Processamento de Arquivos

- A constante simbólica EOF (End Of File), definida no cabeçalho <stdio.h>, pode ter dois significados:
  - tentativa de leitura além do final do arquivo e,
  - ocorrência de erro de leitura.

Principais Funções de Manipulação de Arquivos

- **fflush()** - Descarrega o buffer de saída.

```
int fflush(FILE *stream)
```

- **fscanf()** - Leitura de entrada formatada (semelhante ao scanf()).

```
int fscanf(FILE *stream, const char *formato, ...)
```

- **fprintf()** - Escrita de saída formatada (semelhante ao printf()).

```
int fprintf(FILE *stream, const char *formato, ...)
```

Principais Funções de Manipulação de Arquivos

- **remove()** - exclui o arquivo cujo nome (string) é recebido como parâmetro

```
int remove(const char *nomeDoArquivo)
```

- **rename()** - altera o nome de um arquivo

```
int rename(const char *nomeAtual, const char *nomeNovo)
```

Principais Funções de Manipulação de Arquivos

- **fgets()** - Obtém uma string do arquivo.

```
char *fgets(char *ar, int n, FILE *stream)
```

- **fgetc()** - Obtém um caracter do arquivo.

```
int fgetc(FILE *stream)
```

- **fputs()** - Insere uma string no arquivo.

```
int fputs(const char *s, FILE *stream)
```

Principais Funções de Manipulação de Arquivos

- **fputc() -** Insere um caracter no arquivo.

```
int fputc(int byte, FILE *stream)
```

- **fread()** - Lê um bloco de dados do arquivo.

```
size_t fread( void *ar, size_t tamanho, size_t n, FILE *stream )
```

- **fwrite()** - Escreve um bloco de dados no arquivo.

```
size_t fwrite( const void *ar, size_t tamanho, size_t n, FILE *stream )
```

Principais Funções de Manipulação de Arquivos

- **fseek()** - Reposiciona o ponteiro.

```
int fseek(FILE *stream, int distancia, int deOnde)
```

- **rewind()** - Reposiciona o ponteiro para o início do arquivo.

```
void rewind(FILE *stream)
```

- **ftell()** - Retorna a posição do ponteiro.

```
int ftell(FILE *arquivo)
```

Exercícios

1. Faça um programa que leia o nome e sobrenome de 10 alunos e armazene em um arquivo, de tal forma que o arquivo tenha um aluno por linha.

2. Faça um programa que leia um vetor de inteiros A de tamanho 20 e guarde seus valores em um arquivo, um por linha. Em seguida, reabra o arquivo e leia os elementos para o vetor B, verificando se os valores foram gravados corretamente.