

SSC0501 - Introdução à Ciência de Computação I

Bem Vindos!

Entrada e Saída de Dados em C

- Na linguagem C, **não existem comandos nativos** para Entrada e Saída de dados (E/S).
- Essas operações são realizadas por meio de **funções específicas**, organizadas em **bibliotecas padrão** (por exemplo, stdio.h).
- Em C, os dados podem ser manipulados através de dois tipos principais de **fluxos**:
 - **Fluxo de texto:**
 - **Fluxo binário:**

Fluxos (Streams)

- **Fluxo de Texto**

- É composto por uma **sequência de caracteres**, que podem ou não ser divididos em **linhas** terminadas por um caractere especial de **fim de linha** ('\n').
- Atenção: na última linha de um arquivo de texto não é obrigatório o caractere de fim de linha — isso pode causar diferenças no processamento entre sistemas ou ferramentas.
- Em sistemas como Windows, o fim de linha no arquivo é representado por dois caracteres ('\r\n'), mas, ao ler o arquivo em C, essa sequência é automaticamente convertida para '\n'.

Fluxos (Streams)

- **Fluxo Binário**

- É composto por uma **sequência de bytes** lidos **diretamente do dispositivo externo, sem qualquer tradução** ou interpretação.
- Existe uma correspondência **um para um** entre os bytes armazenados no dispositivo (arquivo, por exemplo) e os bytes manipulados na memória.
- Esse tipo de fluxo é ideal para trabalhar com arquivos que não são de texto, como **imagens, áudio**, ou **estruturas de dados** serializadas.

Arquivos

- Em C, os **arquivos** são tratados como **fluxos de dados (streams)** que estão armazenados em **dispositivos de armazenamento secundário** (como discos rígidos, SSDs, etc.).

Exemplo de fluxos:

Fluxos padrão (pré-definidos):

- **stdin**: Entrada padrão (normalmente teclado)
- **stdout**: Saída padrão (normalmente terminal)
- **stderr**: Saída padrão de erro

Fluxos associados a arquivos ou dispositivos de hardware

Fluxos associados a pipe para comunicação entre processos

Arquivos

- Em C, os **arquivos** são tratados como **fluxos de dados (streams)** que estão armazenados em **dispositivos de armazenamento secundário** (como discos rígidos, SSDs, etc.).

Exemplo de fluxos:

Fluxos padrão (pré-definidos):

- **stdin**: Entrada padrão (normalmente teclado)
- **stdout**: Saída padrão (normalmente terminal)
- **stderr**: Saída padrão de erro

Fluxos associados a arquivos ou dispositivos de hardware

Fluxos associados a pipe para comunicação entre processos

Principais operações com arquivos

- **Abertura e fechamento de arquivos**

- Antes de manipular um arquivo, ele deve ser aberto com a função `fopen()`.
- Após a manipulação, o arquivo deve ser fechado com `fclose()` para garantir a integridade dos dados.

- **Apagar um arquivo**

- Arquivos podem ser removidos do sistema com a função `remove()`.

- **Leitura e escrita de um caractere**

- Funções como `fgetc()` e `fputc()` permitem a leitura ou escrita de um caractere por vez.

Principais operações com arquivos

- **Indicação de fim de arquivo**

- A função `feof()` permite verificar se o fim do arquivo (EOF) foi alcançado durante a leitura.

- **Reposicionamento no arquivo**

- É possível posicionar o cursor em um ponto específico do arquivo usando funções como `fseek()` e `rewind()`.

Funções da biblioteca stdio.h

Função	Descrição
<i>fopen()</i>	Abre um arquivo
<i>fclose()</i>	Fecha um arquivo
<i>fseek()</i>	Posiciona o ponteiro do arquivo
<i>feof()</i>	Retorna VERDADE se chegou ao fim do arquivo
<i>fflush()</i>	Descarrega o buffer associado com o arquivo

Função	Descrição
<i>fprintf()</i>	Equivalente a printf(), usando stream
<i>fscanf()</i>	Equivalente a scanf(), usando stream
<i>fgets()</i>	Equivalente a gets(), usando stream
<i>fwrite()</i>	Escrita de dados no modo binário
<i>fread()</i>	Leitura de dados no modo binário

Abrindo e fechando um arquivo

```
1 #include <stdio.h>
2
3 int main ()
4 {
5     // Declara um ponteiro para arquivo (FILE)
6     FILE *arq;
7
8     // Abre/cria arquivo. Retorna NULL se ocorrer algum erro.
9     arq=fopen("dados.txt", "w+");
10
11     if (arq==NULL)
12     {
13         printf("Erro na abertura do arquivo");
14         return 0;
15     }
16
17     // Fecha um arquivo
18     fclose(arq);
19
20     return 0;
21 }
```

[abrindo_arquivo.c](#)

Modos de abertura

- **r**: abre o arquivo para **leitura**
(falha se o arquivo não existir)
- **w**: cria ou abre o arquivo para **escrita**
(apaga o conteúdo existente, se houver)
- **a**: abre ou cria o arquivo para **escrita no final (anexar)**
(não apaga o conteúdo existente)
- **r+**: abre o arquivo para **leitura e escrita**
(falha se o arquivo não existir)

Modos de abertura

- **w+**: cria ou abre o arquivo para **leitura e escrita** (apaga o conteúdo existente)
- **a+**: abre ou cria o arquivo para **leitura e escrita no final** (anexar)
(as gravações sempre vão para o fim do arquivo)

Funções para manipulação de arquivo

- `fprintf(arq, "string", variáveis)` – escreve no arquivo (semelhante ao `printf`)
- `fscanf(arq, "string", variáveis)` – le do arquivo (semelhante ao `scanf`)
- `fgets(*dados, tamanho dos dados, arq)` – le do arquivo (semelhante ao `gets`)
- `fwrite(*dados, tamanho dos dados, número de itens, arq)` – escreve dados no arquivo.
- `fread(*dados, tamanho dos dados, número de itens, arq)` – le dados do arquivo.

Funções para manipulação de arquivo

- `fseek(arq, posição a ser buscada, posição de referência)` - busca posição em arquivo aberto
 - posição a ser buscada: em bytes
 - posição de referência:
 - `SEEK_SET` para início do arquivo
 - `SEEK_CUR` para posição atual
 - `SEEK_END` para fim do arquivo
- `feof(arq)` - retorna zero se ponteiro não aponta para o fim do arquivo
- `fflush(arq)` – descarrega o buffer no arquivo

Exemplo: escrita de texto no arquivo

```
1 #include <stdio.h>
2
3 int main ()
4 {
5     FILE *arq;
6     int i, n;
7
8     arq=fopen("test.txt", "w+");
9
10    if(arq==NULL)
11        printf("\nErro abrindo arquivo.\n");
12
13    for(i=0; i<5; i++)
14    {
15        scanf("%d", &n);
16        fprintf(arq, "%d\n", n);
17    }
18
19    fclose(arq);
20
21    return 0;
22 }
```

[escrita_arquivo.c](#)

Exemplo: leitura de texto no arquivo

```
1#include <stdio.h>
2
3int main ()
4{
5    FILE *arq;
6    int i, n;
7
8    arq=fopen("test.txt", "r+");
9    if(arq == NULL)
10        printf("\nErro abrindo arquivo.\n");
11
12    for(i=0; i<5; i++)
13    {
14        fscanf(arq, "%d", &n);
15        printf("%d\n", n);
16    }
17
18    fclose(arq);
19
20    return 0;
21}
```

[leitura_arquivo.c](#)

Prática!

Código Matricula: R4SM

<https://runcodes.icmc.usp.br/offerings/view/83>

[run.codes]



Menu Professor ▾

matheus.m.santos@icmc.usp.br ▾

Hora do Servidor: 14/03/2025 18:46:06

Home > SSC0501

SSC0501 - Introdução à Ciência de Computação I

Professores/Monitores

Professores: Matheus Machado dos Santos
Turma: 2025101
Universidade: USP
Ativa até: 21/07/2025



Código de Matrícula

R4SM



Novo Exercício



Enviar E-mail



Ver Notas



Exportar Tabela de Notas

Exercícios

No.	Exercício	Status	Casos Corretos	Nota	Entregas	Participantes	Prazo de Entrega	Ações
1	Hello World	Finalizado	1/1	10.00	78	43/46	14/03/2025 21:00:00	Ver Detalhes Remover Exercício
2	1.01 Maior número	Não Entregue	0/6	0	0	0/46	19/03/2025 23:59:59	Ver Detalhes Remover Exercício
3	1.02 Par ou ímpar	Não Entregue	0/7	0	0	0/46	19/03/2025 23:59:59	Ver Detalhes Remover Exercício
4	1.03 Positivo, negativo ou zero	Não Entregue	0/7	0	0	0/46	19/03/2025 23:59:59	Ver Detalhes Remover Exercício
5	1.04 Maior de três números	Não Entregue	0/7	0	0	0/46	19/03/2025 23:59:59	Ver Detalhes Remover Exercício

SSC0501 - Introdução à Ciência de Computação I

Obrigado pela atenção!!