

# CCP130

# Desenvolvimento de Algoritmos

Prof. Danilo H. Perico



# Arquivos

# Arquivos

- Utilizar um arquivo é uma maneira **excelente de trabalhar** com a **entrada** e a **saída** de dados para os programas
- Um **arquivo** é um mecanismo de abstração utilizado para armazenar informações
- Em 1952, **arquivo** era utilizado para se referir aos cartões perfurados.
- Em muitos sistemas operacionais, os arquivos são organizados como uma **sequência contínua de bytes**



# Arquivos

- Utilizar **arquivos** é uma forma de **garantir o armazenamento persistente** dos dados que são importantes no seu programa, pois nenhuma variável continua existindo depois que o programa termina:

- As **variáveis** são armazenadas na **memória principal** (memória **RAM**) durante a execução do programa
- Quando o programa encerra, as variáveis são desalocadas e perdem seus conteúdos



# C: Arquivos e Streams



- **Arquivo**
  - É um dispositivo real onde as informações realmente estarão gravadas.

# C: Arquivos e Streams



- **Streams**

- É uma abstração utilizada por linguagens de programação para representar um arquivo
- Isso permite trabalhar com dispositivos de fita, disco e terminais sem ter que se preocupar com os detalhes do dispositivo
- Agrega o conceito de **Buffer** :
  - uma operação de escrita será feita somente quando o Buffer estiver cheio, evitando muito acesso ao dispositivo.

# C: Arquivos e Streams



- **Streams**

- ***Buffer*** :

- Área de memória (RAM) usada para temporariamente armazenar dados quando eles estão sendo transferidos entre um computador e um outro dispositivo

# C: Streams



- Existem 2 tipos de Streams:
  - **Texto:**
    - Sequência de caracteres
  - **Binário:**
    - É uma sequência de bytes com uma correspondência de um para um com aqueles caracteres encontrados no dispositivo externo



# C: Arquivos



- Em C, um arquivo pode ser qualquer coisa:
  - Arquivo em disco (*.txt, por exemplo*)
  - Terminal
  - Impressora
- Para escrever nestes dispositivos, devemos associá-los a uma **Stream** efetuando uma operação de **abertura**
- Para desassociar um arquivo de uma **Stream**, basta efetuar uma operação de **fechamento**

# C: Arquivos




## FILE \* (Ponteiro para um arquivo)

- Um ponteiro para um arquivo é o meio comum que une o sistema C ANSI de Entrada/Saída a um *Buffer*.
- Define diversas informações sobre um arquivo:
  - Nome
  - Status
  - Posição atual
- Para ler e escrever em arquivos você precisará este recurso!

# C: Arquivos - Principais Funções

Função	Ação
fopen( )	Abre um arquivo
fclose( )	Fecha o arquivo
putc( ) e fputc( )	Escreve um caractere em um arquivo
getc( ) e fgetc( )	Lê um caractere de um arquivo
fprintf( )	Efetua a impressão formatada em um arquivo
fscanf( )	Efetua a leitura formatada em um arquivo
feof( )	Verifica o final do arquivo
fwrite( )	Escreve tipos maiores que 1 byte em arquivo
fread( )	Lê tipos maiores que 1 byte do arquivo

# C: Arquivos - Como abrir?



```
1  #include <stdio.h>
2
3  int main(void){
4      FILE* fopen(const char* nomeArq,
5                  const char* modo);
6  }
```

- Abre o arquivo *nomeArq* no modo *modo*

# C: Arquivos - Como abrir?



- Modos possíveis:

```
r  - open for reading
w  - open for writing (file need not exist)
a  - open for appending (file need not exist)
r+ - open for reading and writing, start at beginning
w+ - open for reading and writing (overwrite file)
a+ - open for reading and writing (append if file exists)
```

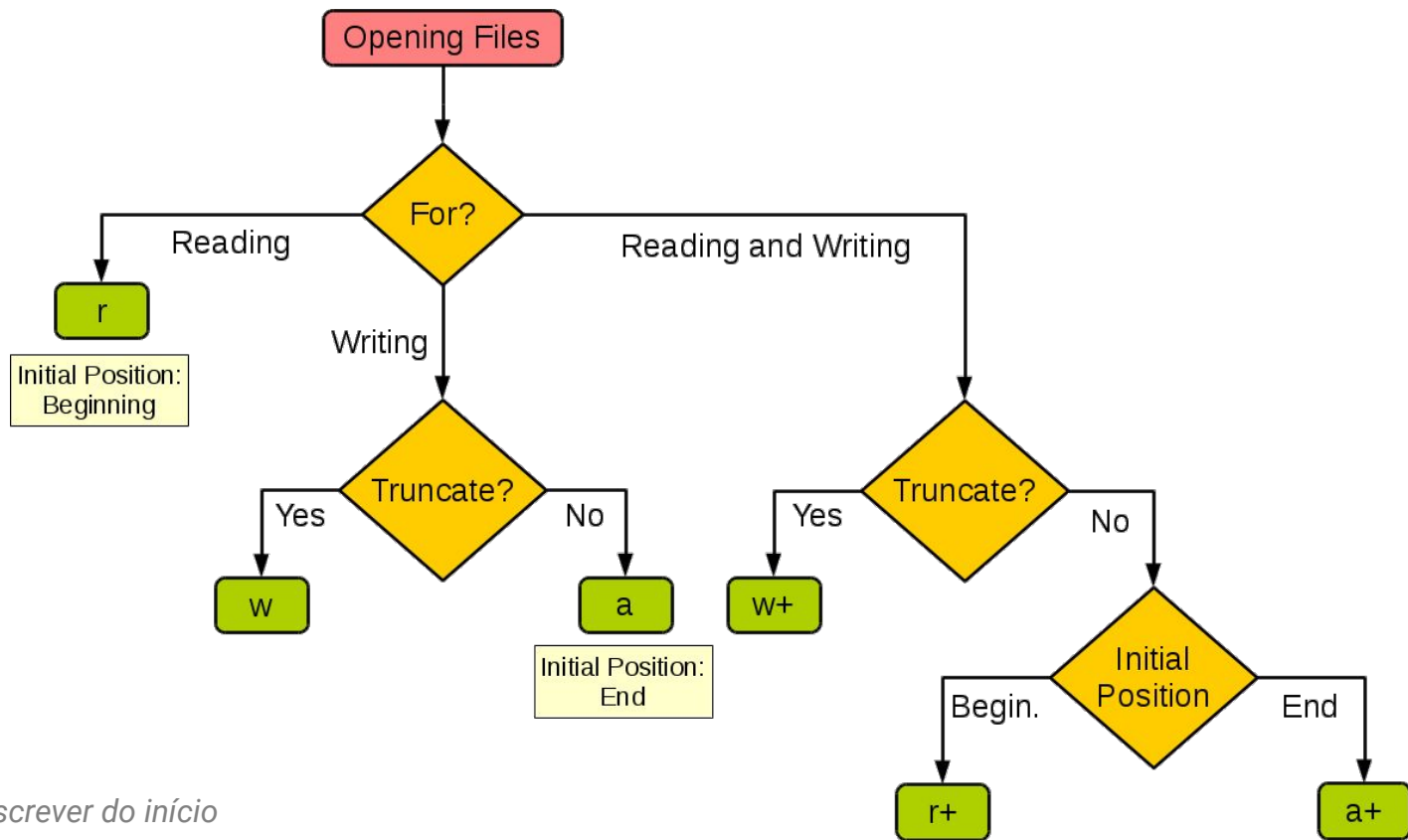
# C: Arquivos - Como abrir?

- Modos possíveis:

```
r - open for reading  
w - open for writing (file need not exist)  
a - open for appending (file need not exist)  
r+ - open for reading and writing, start at beginning  
w+ - open for reading and writing (overwrite file)  
a+ - open for reading and writing (append if file exists)
```


Estes modos apagam os arquivos caso os mesmos já estejam criados

# C: Arquivos - Como abrir?



*\*truncar: sobrescrever do início*

# C: Arquivos -Exemplo



```
1  #include <stdio.h>
2
3  ✓int main(void){
4      FILE* arquivo;
5      arquivo = fopen("teste.txt", "w");
6  }
```

- Cria e abre o arquivo **teste.txt** no modo **w:**
  - Para escrita de texto



# C: Arquivos -Exemplo

- Podemos verificar se a abertura do arquivo foi realizada de maneira correta!

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void){
5      FILE* arquivo;
6      arquivo = fopen("teste.txt", "w");
7      if(arquivo == NULL){
8          printf("Pan! Nao consegui abrir o arquivo!");
9          exit(1);
10     }
11 }
```

Sai do programa, se != 0 significa um erro - precisa do stdlib.h

## C: Arquivos - Escrita



- Exemplo: ***fopen, fprintf, fclose***
- Escrever em um arquivo texto a conta e o nome de vários cliente, conforme entrada do usuário

## C: Arquivos - Leitura



- Exemplo completo: ***fopen, feof, fscanf, fclose***
- Ler de um arquivo texto a conta e o nome de todos os clientes

# Exercícios

# Exercícios

1. Faça um programa que cria e grava números pares e ímpares em arquivos separados. Números de 0 a 999. Números pares no arquivo *pares.txt* e os ímpares no arquivo *impares.txt*