

# Programação II

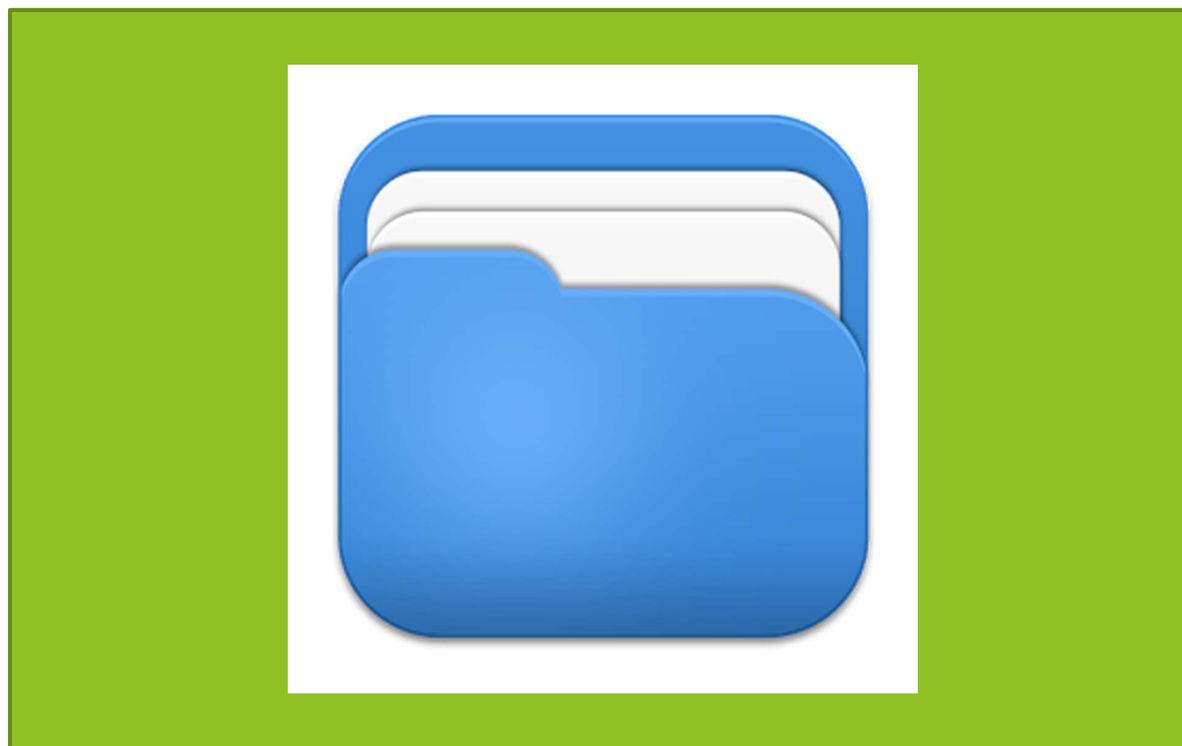
Jordana S. Salamon

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

DEPARTAMENTO DE INFORMÁTICA

CENTRO TECNOLÓGICO

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO



Arquivos

# Definição

- ▶ Um arquivo é uma coleção de bytes armazenados em um dispositivo de armazenamento.
- ▶ Dois tipos de arquivos na linguagem C:
  - ▶ **Arquivos binários**
    - ▶ Armazena dados como estão organizados na memória do computador (arquivos executáveis, arquivos compactados)
  - ▶ **Arquivos de Texto**
    - ▶ Armazena dados como sequências de caracteres
- ▶ A linguagem C não possui funções que automaticamente leiam todas as informações de um arquivo, mas é possível manipula-los no código.
- ▶ A biblioteca necessário para manipular um arquivo é a `<stdio.h>`

```
001110100101101001
010010101101000100
110001100011000111
000111001010110100
110001110001100011
101100110101001011
110100100101010011
101010100000111101
010100010100010111
000101010111010110
```

Isso é um  
arquivo contendo  
informação  
  
traduzida como  
caracteres

# Manipulação de Arquivos

- ▶ Declarando uma variável do tipo arquivo:

```
FILE *fp;
```

Ponteiro para Arquivo

- ▶ Função para abrir um arquivo:

```
FILE *fopen(char *nome_do_arquivo, char *modo)
```

r - w

- ▶ Função para fechar um arquivo:

```
int fclose(FILE *fp)
```

Ao fechar o arquivo as informações são gravadas.



# Manipulação de Arquivos

**FILE \*fopen (char \*nomearq, char \*modo);**

Parâmetro	Efeito
r	Abre um arquivo-texto para leitura
w	Cria um arquivo-texto para escrita
a	Abre um arquivo-texto para gravar ao fim dele
r+	Abre um arquivo-texto para leitura/escrita
w+	Cria um arquivo-texto para leitura/escrita
a+	Abre ou cria (se não existir) um arquivo-texto para ler dele ou gravar ao fim dele
rb	Abre um arquivo binário para leitura
wb	Cria um arquivo binário para escrita
ab	Abre um arquivo binário para gravar ao fim dele
r+b	Abre um arquivo binário para leitura/escrita
w+b	Cria um arquivo binário para leitura/escrita
a+b	Abre ou cria um arquivo binário para gravar ao fim dele

**FILE \*arq1 , \*arq2;**

**arq1 = fopen ("somente\_escrita.txt", "w");**

**arq2 = fopen ("leio\_e\_escrevo.meu", "r");**

# Definição de arquivos no programa

- ▶ a função `fopen()`
  - ▶ avaliação de seu valor de retorno:
  - ▶ erros na abertura de um arquivo
    - ▶ exemplo: passar para `fopen()` o código “r”, mas o arquivo não existe
  - ▶ retorno de `fopen()` é `NULL` indicando a falha na abertura do arquivo

```
FILE *arq;  
arq = fopen ("alunos.txt", "r");  
if (arq == NULL) {  
    printf ("Erro na abertura de arquivo! Programa terminado...");  
    exit (1);  
}
```

`exit()` é uma função da biblioteca `stdlib.h` da linguagem C, que retorna o controle ao SO, passando um código de retorno e terminando o programa.

# Definição de arquivos no programa

- ▶ Para o nome do arquivo, deve-se indicar o caminho (estrutura de pastas) até o mesmo, que pode ser:
  - ▶ Absoluto (completo): O caminho de pastas do arquivo desde a raiz até a pasta onde se encontra o arquivo.
  - ▶ Relativo: em relação à pasta onde se encontra o executável do programa. Exemplo: para arquivos na mesma pasta, usa-se somente “arquivo.txt”.



# Fechamento de arquivos

- ▶ retorna zero se a operação terminou sem falhas
- ▶ caso contrário, retorna um valor diferente de zero
- ▶ recebe como parâmetro um ponteiro para FILE
- ▶ quando `fclose()` é usada, os dados pendentes na “buffer” do arquivo são escritos e o arquivo físico é dissociado de FILE\*



# Exemplo

```
#include <stdio.h>

int main() {
    FILE *f1, *f2;

    f1 = fopen("entrada.txt", "r");
    f2 = fopen("saida.bin", "w");

    fclose(f1);
    fclose(f2);
}
```



# Leitura e Escrita

```
#include <stdio.h>
```

```
int main() {  
    FILE *f1, *f2;  
    int x;
```

```
    f1 = fopen("entrada.txt", "r");  
    f2 = fopen("saida.bin", "w");
```

**Leitura** `fscanf(f1, "%d", &x);`

**Escrita** `fprintf(f2, "%d", x);`

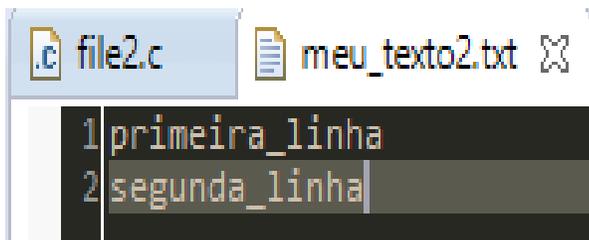
```
    fclose(f1);  
    fclose(f2);
```

```
}
```

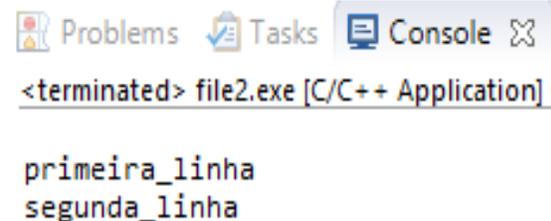


# Uso de fgets

```
8 #include <stdio.h>
9 #include <stdlib.h>
10 #define NOME_ARQUIVO "meu_texto2.txt"
11 #define TAM_STR 50
12
13 int main(void) {
14     char str1[50];
15     char str2[40];
16     FILE *arq;
17
18     if (!(arq = fopen(NOME_ARQUIVO, "r"))) {
19         printf("\nErro na abertura de arquivo! Abortando o programa ...\n");
20         exit(1);
21     }
22
23     fgets(str1, 50, arq);
24     fgets(str2, 40, arq);
25     printf("\n%s%s", str1, str2);
26     return 0;
27 }
28
```



```
1 primeira_linha
2 segunda_linha
```



Problems Tasks Console

<terminated> file2.exe [C/C++ Application]

```
primeira_linha
segunda_linha
```

# Verificar se o arquivo foi aberto corretamente

```
f1 = fopen("entrada.txt", "r");  
if (f1==NULL) {  
    printf("Erro ao abrir o arquivo!\n");  
    exit(1); //Finaliza o programa  
}
```

**<stdlib.h>**

# Ler o arquivo até o final

```
f1 = fopen("entrada.txt", "r");  
if (f1==NULL) {  
    printf("Erro ao abrir o arquivo!\n");  
    exit(1); //Finaliza o programa  
}  
  
while (!feof(f1)) {  
    fscanf(f1, "%d", &x);  
    fprintf(f2, "%d ", x);  
}
```



# Exercícios

- 1. Escreva um programa que leia do usuário os nomes de 2 arquivos texto. Em seguida, crie um terceiro arquivo texto com o conteúdo dos dois primeiros juntos (o conteúdo do primeiro seguido do conteúdo do segundo).
- 2. Faça um programa que leia um arquivo texto contendo uma lista de compras. Cada linha do arquivo possui nome, quantidade e valor unitário do produto. O programa então exibe o total da compra.

That's all Folks!



nemo