



# Introdução

- As informações em disco são persistentes.
  - Atribuição de nomes aos elementos de informação (arquivos e diretórios (pastas)).
  - Os nomes dos arquivos são, em geral, compostos pelo nome em si seguido de uma extensão:
    - Arquivo.c
    - Arquivo.doc



# Introdução

- Um arquivo pode ser visto de duas maneiras:
  - Modo texto
  - Modo binário
- Podemos optar por salvar (e recuperar) informações em disco em um dos dois modos.



# Introdução

- Principais serviços oferecidos pela linguagem C:
  - Abertura de arquivos
  - Leitura do arquivo
  - Escrita no arquivo
  - Fechamento de arquivo



# Introdução

- Uma das informações mantidas pelo SO é um *cursor* que indica a posição de trabalho no arquivo.
- Biblioteca de entrada e saída (*stdio.h*).

The background of the slide is a blurred, high-angle photograph of a laptop keyboard and a computer mouse. The text is overlaid on this image.

# **Funções para abrir e fechar arquivos**

# Funções para abrir e fechar arquivos

- Função básica para abrir um arquivo:  
`FILE* fopen (char* nome_arquivo, char* modo);`
- Modos de abertura (caracteres interpretados):

r	<i>read</i>	Indica modo para leitura.
w	<i>write</i>	Indica modo para escrita.
a	<i>append</i>	Indica modo para escrita ao final do existente.
t	<i>text</i>	Indica modo texto.
b	<i>binary</i>	Indica modo binário.

# Funções para abrir e fechar arquivos

- Testando abertura de um arquivo:

```
...  
FILE* fp;  
fp = fopen("entrada.txt", "rt");  
if(fp == NULL) {  
    printf("Erro na abertura do arquivo!\n");  
    exit(1);  
}  
...
```

Fonte: CELES;CERQUEIRA;RANGEL, 2004

# Funções para abrir e fechar arquivos

- Função para fechamento de um arquivo:  
`int fclose (FILE* fp);`
  - Retorna zero se fechado com sucesso ou EOF caso contrário.





**Arquivos em modo texto**



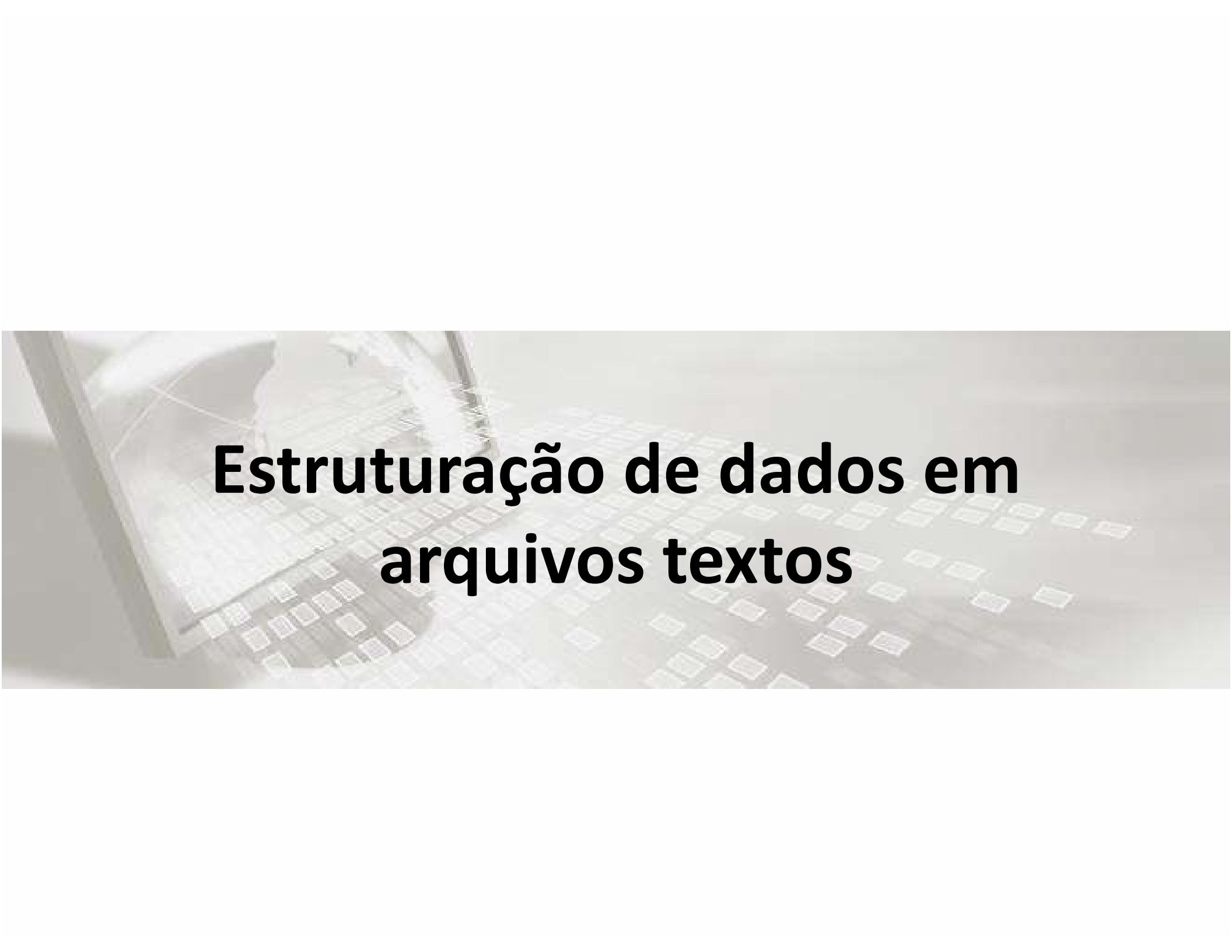
# Arquivos em modo texto

- Funções para ler dados:
  - `int fscanf (FILE* fp, char* formato, ...);`
  - `int fgetc (FILE* fp);`
  - `char* fgets (char* s, int n, FILE* fp);`
- A informação lida é sempre a informação apontada pelo cursor do arquivo.



# Arquivos em modo texto

- Funções para escrever dados:
  - `int fprintf (FILE* fp, char* formato, ...);`
  - `int fputc (int c, FILE* fp);`

The background of the slide features a blurred image of a laptop screen. On the screen, there is a grid of data, possibly a spreadsheet or a database table, with various colored cells. A hand is visible, pointing at the screen, suggesting an interaction with the data. The overall tone is professional and technical.

# **Estruturação de dados em arquivos textos**

# Estruturação de dados em arquivos textos

- Acesso caractere a caractere (Exemplo I - parte I):

```
#include <stdio.h>

int main(void) {
    int c;
    int nlinhas = 0;
    FILE *fp;
    /* abre arquivo para leitura */
    fp = fopen("entrada.txt", "rt");
    if(fp == NULL) {
        printf("Nao foi possivel abrir arquivo.\n");
        return 1;
    }
}
```

# Estruturação de dados em arquivos textos

- Acesso caractere a caractere (Exemplo I - parte II):

```
/* lê caractere a caractere */  
while((c = fgetc(fp)) != EOF){  
    if(c == '\n')  
        nlinhas++;  
}  
/* fecha arquivo */  
fclose(fp);  
/* exibe resultado na tela */  
printf("Numero de linhas = %d\n", nlinhas);  
return 0;  
}
```

# Estruturação de dados em arquivos textos

- Acesso caractere a caractere (Exemplo II - parte I):

```
#include <stdio.h>
#include <ctype.h> /* função toupper */

int main(void) {
    int c;
    char entrada[121]; /* armazena nome do arquivo de entrada */
    char saida[121]; /* armazena nome do arquivo de saída */
    FILE* e; /* ponteiro do arquivo de entrada */
    FILE* s; /* ponteiro do arquivo de saída */

    /* pede ao usuário os nomes dos arquivos */
    printf("Digite o nome do arquivo de entrada: ");
    scanf("%120s", entrada);
    printf("Digite o nome do arquivo de saída: ");
    scanf("%120s", saida);
```

# Estruturação de dados em arquivos textos

- Acesso caractere a caractere (Exemplo II - parte II):

```
/*abre arquivos para leitura e para escrita */  
e = fopen(entrada, "rt");  
if(e == NULL){  
    printf("Nao foi possivel abrir arquivo de entrada.\n");  
    return 1;  
}  
s = fopen(saida, "wt");  
if(s == NULL){  
    printf("Nao foi possivel abrir arquivo de saida.\n");  
    fclose(e);  
    return 1;  
}
```



# Estruturação de dados em arquivos textos

- Acesso caractere a caractere (Exemplo II - parte III):

```
/* lê da entrada e escreve na saída */
while((c = fgetc(e)) != EOF)
    fputc(toupper(c), s);

/* fecha arquivos */
fclose(e);
fclose(s);

return 0;
}
```

# Estruturação de dados em arquivos textos

- Acesso linha a linha (parte I):

```
/* Procura ocorrência de subcadeia no arquivo */
#include <stdio.h>
#include <string.h> /* função strstr */

int main(void) {
    int n = 0; /* número da linha corrente */
    int achou = 0; /* indica se achou subcadeia */
    char entrada[121]; /* armazena nome do arquivo de entrada */
    char subcadeia[121]; /* armazena subcadeia */
    char linha[121]; /* armazena cada linha do arquivo */
    FILE* fp; /* ponteiro do arquivo de entrada */
```

# Estruturação de dados em arquivos textos

- Acesso linha a linha (parte II):

```
/* pede ao usuário o nome do arquivo e a subcadeia */
printf("Digite o nome do arquivo de entrada: ");
scanf("%120s", entrada);
printf("Digite a subcadeia: ");
scanf("%120s", subcadeia);

/* abre arquivo para leitura */
fp = fopen(entrada, "rt");
if(fp == NULL) {
    printf("Nao foi possivel abrir arquivo de entrada.\n");
    return 1;
}
```

# Estruturação de dados em arquivos textos

- Acesso linha a linha (parte III):

```
/* lê linha a linha */  
while(fgets(linha, 121, fp) != NULL) {  
    n++;  
    if(strstr(linha, subcadeia) != NULL) {  
        achou = 1;  
        break;  
    }  
}  
  
/* fecha arquivo */  
fclose(fp);
```

# Estruturação de dados em arquivos textos

- Acesso linha a linha (parte IV):

```
/* exibe saída */  
if(achou)  
    printf("Achou na linha %d.\n", n);  
else  
    printf("Nao achou.");  
  
return 0;  
}
```



# Referências

- CELES, W.; CERQUEIRA, R.; RANGEL, J. L. **Introdução a Estruturas de Dados: com técnicas de programação em C**. Rio de Janeiro: Ed. Campus-Elsevier, 2004.