

# Algoritmos e Programação II

## Manipulação de Arquivos

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

# Bibliografia



**Título:** Como Programa C – Sexta Edição

**Editora:** Pearson

**Autor:** Paul Deitel

## Capítulo 11.

# Manipulação de Arquivo

- O armazenamento de dados em variáveis e arrays é temporário; todos os dados são perdidos quando um programa termina.
- Os arquivos são usados para conservação permanente de grandes quantidades de dados. Os computadores armazenam arquivos em dispositivos secundários de armazenamento, especialmente dispositivos de armazenamento em disco.

# Manipulação de Arquivo

- A linguagem C visualiza cada arquivo simplesmente como um fluxo sequencial de bytes.
- Cada arquivo termina ou com um marcador de final de arquivo (end-of-file marker),

# Manipulação de Arquivo

- Abrir um arquivo retorna um ponteiro para uma estrutura FILE (**definida em <stdio.h>**) que contém informações usadas para processar o arquivo.
- O padrão de entrada, o padrão de saída e o padrão de erros são manipulados por meio dos ponteiros de arquivos **stdin**, **stdout** e **stderr**.

# Gravando Arquivo Sequencial

```
#include <stdio.h>

main() {
    int i;
    FILE *txt;

    if ((txt = fopen("Arquivo.txt", "a+")) == NULL) {
        printf("Erro ao abrir arquivo");
    }
    else {
        i = 50;
        while(i < 100)
        {
            fprintf(txt, "%d\n", i);
            ++i;
        }
        fclose(txt);
    }
}
```

# Modos de Abertura

- Para criar um arquivo ou eliminar o conteúdo de um arquivos antes da gravação dos dados, abra o arquivo para gravação ("w").
- Para ler um arquivo existente, abra-o para leitura ("r").
- Para adicionar registros ao final de um arquivo existente, abra o archive append("a").
- Para abrir um arquivo de forma que ele possa ser gravado e lido, abra o arquivo com um dos três modos de atualização "r+", "w+" ou "a+".
  - O modo "r+" abre um arquivo para leitura e gravação.
  - O modo "w+" cria um arquivo para leitura e gravação. Se o arquivo já existir, o arquivo é aberto e o conteúdo atual é eliminado.
  - O modo "a+" abre um arquivo para leitura e gravação. Toda gravação é feita no final do arquivo. Se o arquivo não existir, é criado.

# Lendo Arquivo Sequencial (fscanf)

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

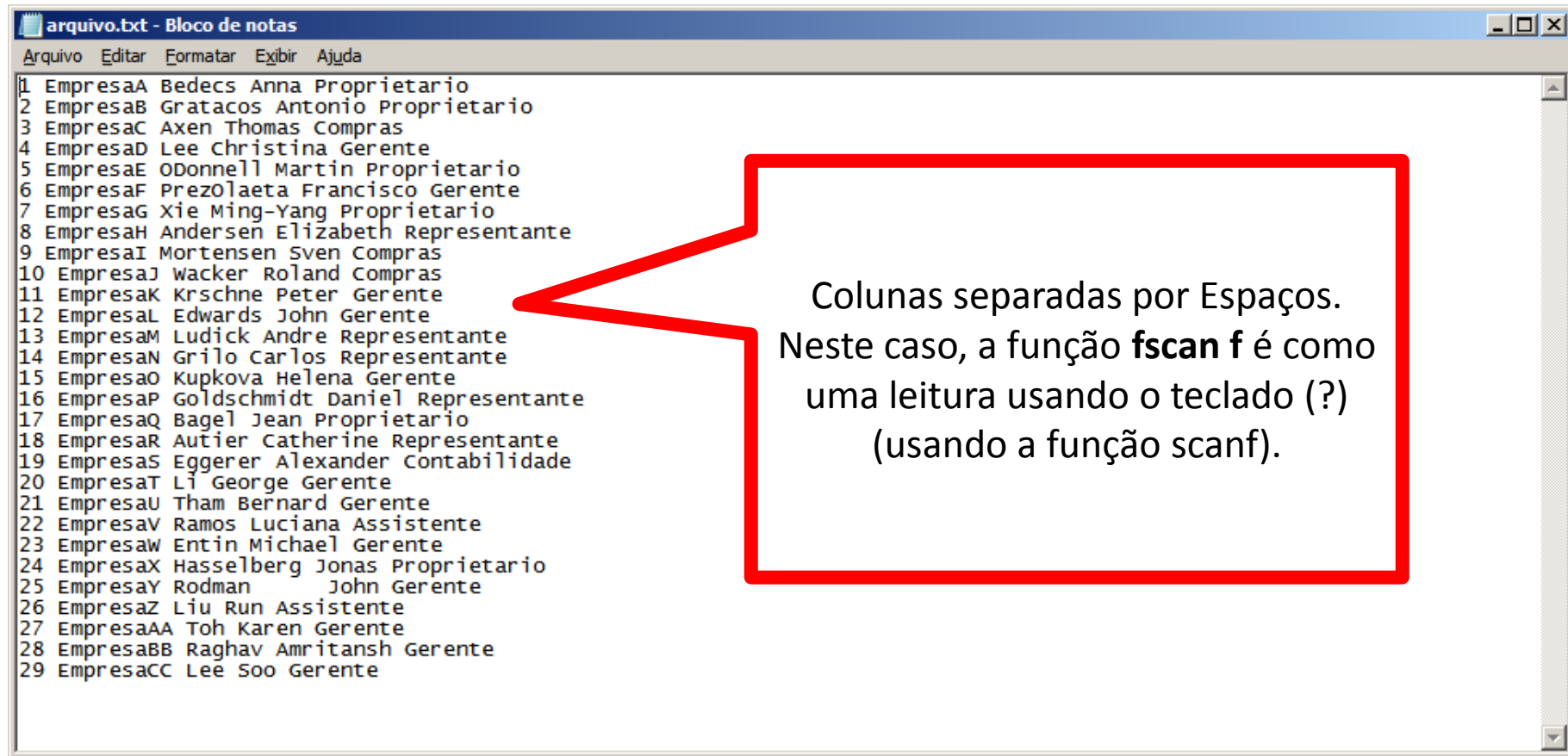
main() {
    int codigo;
    char empresa[10];
    char sobre[12];
    char nome[10];
    char funcao[10];

    FILE *txt;

    if((txt = fopen("c:\\arquivo.txt", "r")) == NULL)
    {
        printf("Erro ao abrir arquivo");
    }
    else
    {
        while (!feof(txt)) {
            fscanf(txt, "%d %s %s %s %s ", &codigo, empresa, sobre, nome, funcao);
            printf("%d \t %-10s %-12s %-10s %-10s \n", codigo, empresa, sobre, nome, funcao);
        }
        fclose(txt);
    }
    system("pause");
}
```



# Layout do Arquivo



```
arquivo.txt - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
1 EmpresaA Bedecs Anna Proprietario
2 EmpresaB Gratacos Antonio Proprietario
3 EmpresaC Axen Thomas Compras
4 EmpresaD Lee Christina Gerente
5 EmpresaE ODonnell Martin Proprietario
6 EmpresaF Prezolaeta Francisco Gerente
7 EmpresaG Xie Ming-Yang Proprietario
8 EmpresaH Andersen Elizabeth Representante
9 EmpresaI Mortensen Sven Compras
10 EmpresaJ Wacker Roland Compras
11 EmpresaK Krschne Peter Gerente
12 EmpresaL Edwards John Gerente
13 EmpresaM Ludick Andre Representante
14 EmpresaN Grilo Carlos Representante
15 EmpresaO Kupkova Helena Gerente
16 EmpresaP Goldschmidt Daniel Representante
17 EmpresaQ Bagel Jean Proprietario
18 EmpresaR Autier Catherine Representante
19 EmpresaS Eggerer Alexander Contabilidade
20 EmpresaT Li George Gerente
21 EmpresaU Tham Bernard Gerente
22 EmpresaV Ramos Luciana Assistente
23 EmpresaW Entin Michael Gerente
24 EmpresaX Hasselberg Jonas Proprietario
25 EmpresaY Rodman John Gerente
26 EmpresaZ Liu Run Assistente
27 EmpresaAA Toh Karen Gerente
28 EmpresaBB Raghav Amritansh Gerente
29 EmpresaCC Lee Soo Gerente
```

Colunas separadas por Espaços.  
Neste caso, a função **fscanf** é como  
uma leitura usando o teclado (?)  
(usando a função scanf).

# Lendo Arquivo Sequencial (fgetc)

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

main()
{
    char caractere;

    FILE *txt;

    if((txt = fopen("c:\\arquivo.txt", "r")) == NULL)
    {
        printf("Erro ao abrir arquivo");
    }
    else
    {
        while (!feof(txt)) {
            caractere = fgetc(txt);
            printf("%c", caractere);
        }

        fclose(txt);
    }
    system("pause");
}
```



Lê um caractere por vez.

# Lendo Arquivo Sequencial (fgetc)

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

main()
{
    char caractere;

    FILE *txt;

    if((txt = fopen("c:\\arquivo.txt","r")) == NULL)
    {
        printf("Erro ao abrir arquivo");
    }
    else
    {
        while (!feof(txt)) {
            caractere = fgetc(txt);
            printf("%c", caractere);
        }
        fclose(txt);
    }
    system("pause");
}
```

Lê um caractere por vez.

```
if (caractere == ' ')
    printf(" | ");
else
    printf("%c", caractere);
```

# Lendo Arquivo Sequencial (fgets)

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

main()
{
    char linha[1024];

    FILE *txt;

    if((txt = fopen("c:\\arquivo.txt","r")) == NULL)
    {
        printf("Erro ao abrir arquivo");
    }
    else
    {
        while (!feof(txt)) {
            fgets(linha, 1024, txt);
            printf("%s", linha);
        }
        fclose(txt);
    }
    system("pause");
}
```

Lê uma string até o **\n** ou **1023** caracteres;

# Lendo Arquivo Sequencial (fgets)

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

main() {
    char linha[1024];
    char * ultima;
    FILE *txt;
    if((txt = fopen("c:\\arquivo.txt","r")) == NULL)
    {
        printf("Erro ao abrir arquivo");
    }
    else {
        while (!feof(txt)) {
            fgets(linha, 1024, txt);
            ultima = strtok (linha, " ");
            while (ultima != NULL)
            {
                printf ("%s ",ultima);
                ultima = strtok (NULL, " ");
            }
            fclose(txt);}
        system("pause"); }
```

**strtok** quebra a string no delimitador.