

Arquivos em C

Material da Prof. Ana Eliza

Arquivos

- Definição

- Um arquivo é uma ***estrutura de dados linear*** (“lista”) que é mantida fora da memória principal, em um dispositivo de armazenamento (memória secundária).
- Um arquivo é uma seqüência de registros.

Arquivos

- Arquivos X Vetores (Comparação)

Vetores	Arquivos
Memória Principal	Memória Secundária
Acesso: Índice de elementos	Acesso: Apontador de Registros

Arquivos

- Tipos de Arquivos

- Arquivo Texto

- É uma seqüência de registros, onde cada registro é uma seqüência de caracteres ASCII (string) delimitada pelos caracteres CR / LF (*carriage return / line feed*).

Arquivos

- Tipos de Arquivos

- Arquivo Binário

- É uma seqüência de registros, onde cada registro é composto por campos e cada campo possui tipos e características específicas.
 - Um ou mais campos desses registros são chamados de campo-chave.
 - O campo-chave diferencia um registro dos demais.

Arquivos

- Sistema de Banco de Dados
 - Um sistema de banco de dados é composto por um ou mais arquivos e por um conjunto de programas de manipulação: *inclusão, exclusão, alteração, consultas, geração de relatórios.*

Arquivos

- Operações Básicas
 - Abrir o arquivo
 - Fechar o arquivo
 - Gravar um registro em um arquivo
 - Ler um registro de um arquivo

Arquivos

- Vantagens

- Armazenamento permanente de informações;
- Armazenamento de grandes quantidades de dados, estando limitado apenas pelo tamanho do meio físico utilizado para sua gravação.

Arquivos

- Comando de Abertura do Arquivo
 - Solicita ao sistema operacional que prepare um arquivo para ser utilizado pelo programa.
 - Associa um nome lógico (nome de variável do programa) ao nome físico do arquivo.
 - Posiciona o apontador de registro do arquivo.

Arquivos

- Comando de Abertura do Arquivo

Sintaxe:

FILE * **fopen** (nomeArquivo, modoAbertura);

Semântica

A função **fopen** “abre” o arquivo solicitado e retorna, para quem a chamou, o endereço do arquivo no dispositivo de armazenamento.

Arquivos

- Declaração de Ponteiros para Arquivos

Sintaxe:

```
FILE * nome_variável;
```

Exemplo:

```
FILE * arq;
```

Semântica:

Uma variável do tipo ponteiro para arquivo será utilizada para armazenar o endereço de um determinado arquivo no dispositivo de armazenamento.

Arquivos

- Comando de Abertura do Arquivo

Exemplo:

```
FILE * arq;
```

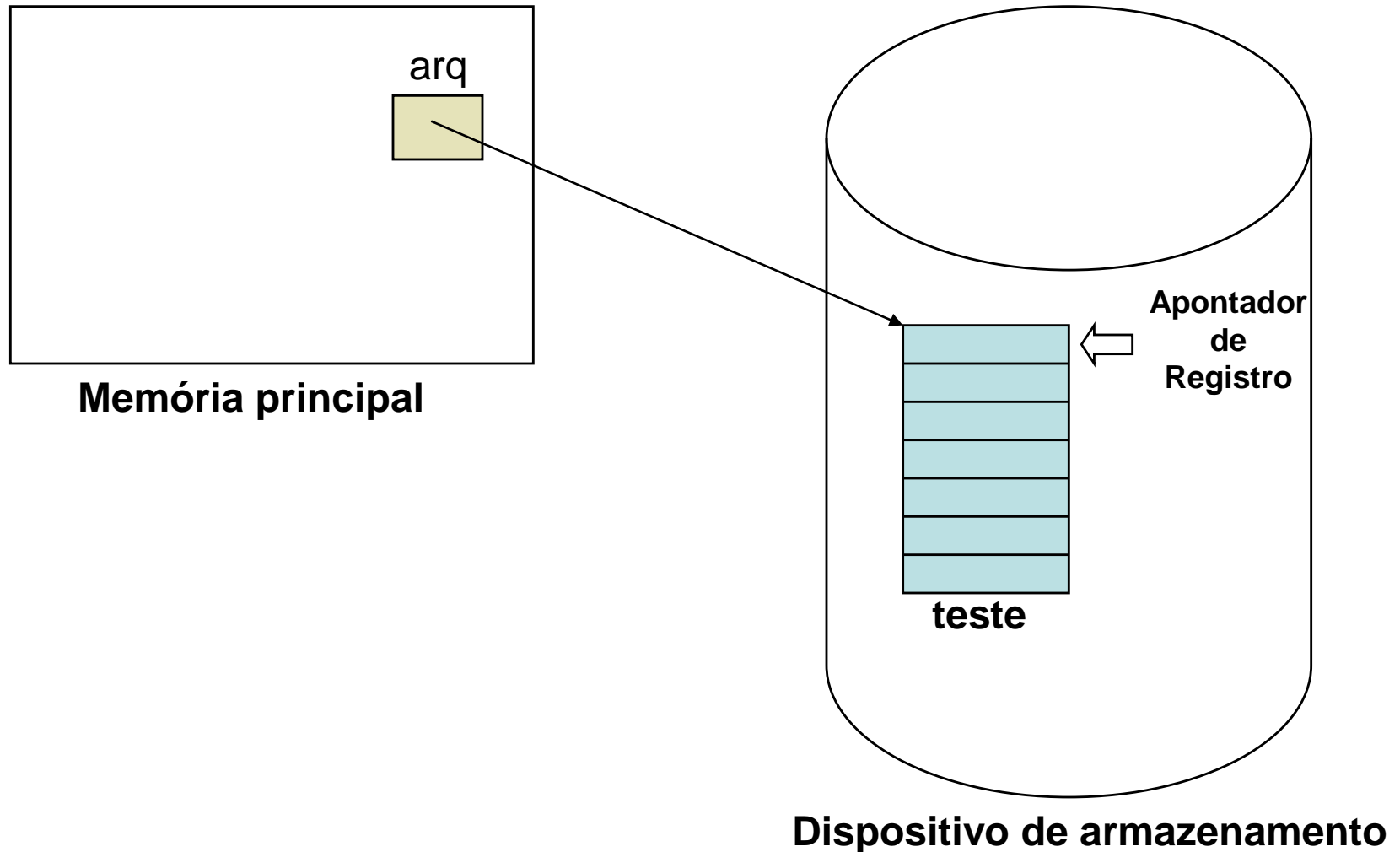
...

```
arq = fopen ("c:\teste", "rb");
```

Semântica

Abre o arquivo binário “teste” para leitura e coloca seu endereço na variável **arq**.

Arquivos



Arquivos

- Modos de Abertura de um Arquivo Binário

Modo	Significado	Ponteiro
rb	Abre um arquivo para leitura	Início
wb	Cria um arquivo para gravação	Início
ab	Abre um arquivo para anexar dados	Final
r+b	Abre um arq. p/ leitura/gravação	Início
w+b	Cria um arq. p/ leitura/gravação	Início
a+b	Cria ou abre um arq. p/ leitura/gravação	Final

OBS: O modo define o tipo das operações que são permitidas no arquivo.

Arquivos

- Modos de Abertura de um Arquivo Texto

Modo	Significado	Ponteiro
rt	Abre um arquivo para leitura	Início
wt	Cria um arquivo para gravação	Início
at	Abre um arquivo para anexar dados	Final
r+t	Abre um arq. p/ leitura/gravação	Início
w+t	Cria um arq. p/ leitura/gravação	Início
a+t	Cria ou abre um arq. p/ leitura/gravação	Final

Arquivos

- Comando de Abertura do Arquivo

Retorno:

- Sucesso: retorna o endereço do arquivo aberto.
- Falha: retorna o valor NULL.

Exemplo:

```
arq = fopen ("c:\teste.txt", "rt");  
if (arq == NULL)  
    printf ("Erro na abertura do arquivo \n");  
else  
    printf ("Arquivo aberto com sucesso \n");
```


Arquivos

- Fechamento do Arquivo

– Informa ao sistema operacional que o arquivo não será mais utilizado pelo programa.

Sintaxe:

```
int fclose (variável_ponteiro_arquivo);
```

Exemplo:

```
fclose (arq);
```

Arquivos

- Fechamento do Arquivo

Retorno:

- Sucesso: retorna o número inteiro zero.
- Falha: retorna um número inteiro diferente de zero.

Exemplo:

```
status = fclose (arq);  
if (status == 0)  
    printf ("Arquivo fechado com sucesso \n");  
else  
    printf ("Erro no fechamento do arquivo \n");
```

Arquivos

- **Exemplo: Programa para criar um arquivo**

```
#include <stdio.h>
int main ( ) {
    FILE * arq;
    char nomeArq [ ] = "teste.txt";
    int status;
    arq = fopen ("teste.txt", "wt");
    if (arq == NULL)
        printf("Erro ao tentar criar o arquivo %s", nomeArq);
    else {
        printf("Arquivo %s criado com sucesso", nomeArq);
        status = fclose (arq);
        if (status == 0)
            printf("Arquivo %s fechado com sucesso", nomeArq);
        else
            printf("Erro ao tentar fechar o arquivo %s", nomeArq);
    }
    return 0;
}
```

Arquivos

- Gravando Dados em um Arquivo

- Grava no arquivo, na posição apontada pelo apontador de registro, as informações contidas na variável indicada.
- Avança o apontador de registro para a posição seguinte ao registro gravado.

Sintaxe:

```
int fwrite (&var, qtd_bytes, cont, arq);
```

Onde:

var: é a variável que armazena o conteúdo a ser gravado no arquivo;

qtd_bytes: é o total em bytes ocupado por um registro do arquivo;

cont: a quantidade de registros do tamanho especificado por qtd_bytes que será gravada no arquivo;

arq: é o endereço do arquivo onde as informações serão gravadas.

Arquivos

- Gravando Dados em um Arquivo

Exemplo:

```
fwrite (&num, sizeof(float), 1, arq);
```

Semântica

Grava o conteúdo da variável **num** que é o tipo **float** no arquivo **arq**.

Arquivos

- Gravando Dados em um Arquivo

Retorno: retorna um valor igual ao número de registros efetivamente gravados.

- Sucesso: retorna um valor igual ao parâmetro **cont**;

- Falha: o valor retornado será menor que **cont**.

Exemplo1:

```
status = fwrite (&num, sizeof (float), 1,arq);  
if (status == 1)  
    printf ("Registro gravado com sucesso \n");  
else  
    printf ("Erro na gravação do registro \n");
```

Arquivos

- Gravando Dados em um Arquivo

Exemplo2:

```
float vet [10];
```

```
...
```

```
status = fwrite (vet, sizeof (float), 10, arq);
```

```
if (status == 10)
```

```
    printf("Registros gravados com sucesso \n");
```

```
else {
```

```
    printf ("Erro na gravação dos registros \n");
```

```
    printf("Qtd. de registros gravados: %i \n", status);
```

```
}
```

Arquivos

- Descarregando o Buffer do Arquivo

Sintaxe:

```
int fflush (arq);
```

Semântica:

Grava o conteúdo armazenado temporariamente no buffer em um determinado arquivo.

Retorno:

- Sucesso: retorna zero;
- Falha: retorna EOF.

Arquivos

- Descarregando o Buffer do Arquivo

Exemplo:

```
status = fflush (arq);  
if (status == 0)  
    printf ("Buffer descarregado com sucesso \n");  
else  
    printf ("Erro ao descarregar o buffer \n");
```

Observações:

- Todos os buffers são automaticamente descarregados no encerramento normal do programa ou quando estão cheios;
- Fechar um arquivo descarrega seu buffer.

Arquivos

- Lendo Dados de um Arquivo

- Lê um registro do arquivo na posição indicada pelo apontador de registros e armazena as informações obtidas na variável indicada.
- Avança o apontador de registro para a posição seguinte ao registro lido.

Sintaxe:

```
int fread (&var, qtd_bytes, cont, arq);
```

Onde:

var: é a variável que vai receber o conteúdo lido do arquivo;

qtd_bytes: é o total em bytes ocupado por um registro do arquivo;

cont: a quantidade de registros do tamanho especificado por qtd_bytes que será lida do arquivo;

arq: é o endereço do arquivo de onde as informações serão lidas.

Arquivos

- Lendo Dados de um Arquivo

Exemplo:

```
fread (&num, sizeof(float), 1, arq);
```

Semântica

Lê um registro, do tipo **float**, no arquivo **arq** e armazena o conteúdo lido na variável **num** que também é o tipo **float**.

Arquivos

- Lendo Dados de um Arquivo

Retorno: retorna um valor igual ao número de registros efetivamente lidos.

- Sucesso: retorna um valor igual ao parâmetro **cont**;
- Falha: o valor retornado será menor que **cont**.

Exemplo1:

```
status = fread (&num, sizeof (float), 1,arq);  
if (status == 1)  
    printf ("Registro lido com sucesso \n");  
else  
    printf ("Erro na leitura do registro \n");
```

Arquivos

- Lendo Dados de um Arquivo

Exemplo2:

```
float vet [10];  
  
...  
status = fread (&vet, sizeof (float),10,arq);  
if (status == 10)  
    printf("Registros lidos com sucesso \n");  
else {  
    printf ("Erro na leitura dos registros \n");  
    printf("Qtd. de registros lidos: %i \n",status);  
}
```

Arquivos

- Função de Teste de Final de Arquivo

Sintaxe:

```
int feof (arq);
```

Semântica:

Verifica se o apontador de registro está posicionado no final do arquivo.

Retorno:

- Fim de arquivo: retorna um valor diferente de zero.
- Caso contrário, retorna zero.

Arquivos

- Função de Teste de Final de Arquivo

Exemplo:

```
while (1) {
    status = fread (&num, sizeof (float), 1,arq);
    if (status == 0) {
        if (feof(arq) != 0)
            printf("Fim de arquivo");
        else
            printf("Erro de leitura");
        break;
    }
    else {
        // processamento dos dados lidos
        ....
    }
}
```

Arquivos

- Movendo o Ponteiro de Registro

Semântica:

Posiciona o cursor (apontador de registro) em um endereço específico, tornando possível leituras e escritas aleatórias.

Sintaxe:

```
int fseek (arq, qtd_bytes, posicao);
```

Onde:

arq: endereço do arquivo que será percorrido.

qtd_bytes: quantidade de bytes que será percorrida a partir de **posicao** para encontrar o registro desejado.

posicao: ponto a partir do qual a busca será executada.

Arquivos

- Movendo o Ponteiro de Registro

Exemplo:

fseek (arq, 0, 0); ⇐ primeiro registro

A variável posicao poderá assumir três valores:

SEEK_SET ou 0 – posição inicial do arquivo;

SEEK_CUR ou 1 – posição atual do arquivo;

SEEK_END ou 2 – posição final do arquivo.

Exemplo: Arquivo de registros do tipo pessoa

fseek(arq,sizeof(pessoa),SEEK_END); ⇐ penúltimo registro

Arquivos

- Movendo o Ponteiro de Registro

Retorno:

- Sucesso: retorna zero.
- Falha: retorna valor diferente de zero.

Exemplo:

```
status = fseek(arq, -sizeof(float), SEEK_CUR);
```

```
if (status != 0)
```

```
    printf (“Falha ao tentar posicionar ponteiro \n”);
```

Arquivos

- Removendo fisicamente um arquivo

Semântica:

Apaga um arquivo do meio de físico.

Sintaxe:

```
int remove (nome_arq);
```

Exemplo:

```
remove ("c:\temp\teste.dat");
```

Arquivos

- Removendo fisicamente um arquivo

Retorno:

- Sucesso: retorna zero.
- Falha: retorna valor diferente de zero.

Exemplo:

```
status = remove("c:\temp\teste.dat");
```

```
if (status != 0)
```

```
    printf ("Falha ao tentar remover o arquivo \n");
```

Arquivos

- Renomeando um arquivo

Semântica:

Altera o nome físico de um arquivo.

Sintaxe:

```
int rename (nomeAntigo,nomeNovo);
```

Exemplo:

```
rename ("c:\temp\teste.dat", "c:\temp\novo.dat");
```

Arquivos

- Renomeando um arquivo

Retorno:

- Sucesso: retorna zero.
- Falha: retorna valor diferente de zero.

Exemplo:

```
status = rename (“teste.dat”, “novo.dat”);
```

```
if (status != 0)
```

```
    printf (“Falha ao tentar renomear o arquivo \n”);
```