

Manipulação de Arquivos

Regina Célia Coelho

rccoelho@unifesp.br

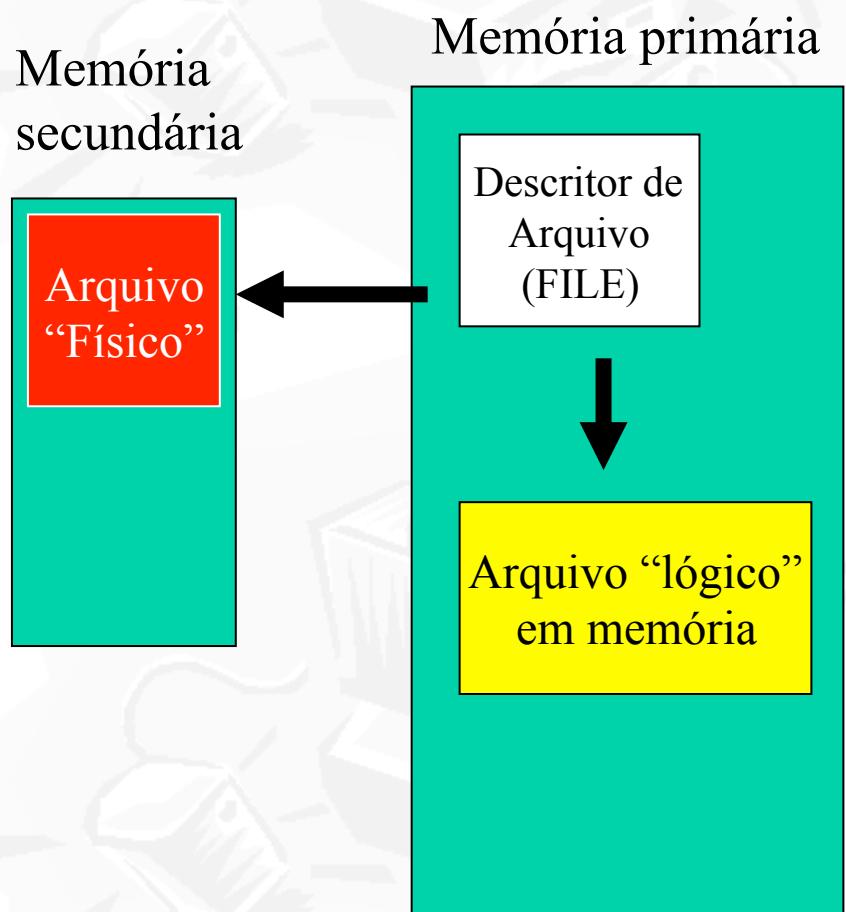
Definição

- ◆ Podem armazenar grande quantidade de informação
- ◆ Dados são persistentes (gravados em disco)
- ◆ Acesso aos dados pode ser não sequêncial (acesso direto a registros em um banco de dados)
- ◆ Acesso a informação pode ser concorrente (mais de um programa ao mesmo tempo)

Tipos de Arquivos

- ◆ Arquivo texto - armazena caracteres que podem ser mostrados diretamente na tela ou modificados por um editor de textos simples.
- ◆ Arquivo binário - sequência de bytes sujeita as convenções dos programas que o gerou (exemplos: arquivos executáveis, arquivos compactados).
 - Formado por uma sequência de bytes sem correspondência com um tipo de dado.
 - Cabe ao programador fazer esta correspondência quando lê e escreve estes arquivos.

Descriptor de Arquivos



- ◆ Em C: Estrutura de dados denominada **FILE**
- ◆ Armazena informações sobre o arquivo em memória secundária (arquivo físico).
- ◆ O arquivo lógico é armazenado em variáveis de memória definidas pelo programador.
- ◆ Criando um descriptor:
FILE *parq;

Abrindo Arquivo

- ◆ Abrir um arquivo: função `fopen(. .)`
 - Associar um descritor a um arquivo físico:

```
FILE *parq;  
parq = fopen("teste.txt", "r");  
if (parq == NULL) {  
    printf("\n\nNão foi possível abrir o arquivo para  
escrita.\n");  
    exit(1);  
}  
else  
    printf("Arquivo aberto para leitura.\n");
```

- Em caso de erro, a função retorna `NULL`

Abrindo Arquivo

◆ Modos de abertura

r	Abre um arquivo texto para leitura
w	Cria um arquivo texto para escrita
a	Adiciona texto ao fim de um arquivo texto
rb	Abre um arquivo binário para leitura
wb	Abre um arquivo binário para escrita
ab	Anexa a um arquivo binário
[r+, w+, a+] + b	O + inclui escrita ou leitura nos modos e o + b define como binário

◆ Exemplo: `parq = fopen ("texto.txt", "w+b");`

Fechando Arquivo

- ◆ Fechando um arquivo: função `fclose(FILE *)`
 - Quando um programa não precisa mais utilizar um arquivo por um período de tempo longo, o arquivo pode ser fechado
 - Fechar um arquivo protege os seus dados, garante que atualizações feitas serão salvas e libera o arquivo para outros usuários ou programas poderem utilizá-lo.

`fclose (parq) ;`

Lendo Dados

- ◆ Quando o arquivo é aberto, a posição corrente (mantida internamente pelo sistema) é o início do arquivo.
- ◆ A cada operação executada sobre o arquivo, essa posição é atualizada.
- ◆ A função **rewind(FILE *)** recoloca o indicador de posição de arquivo no início do arquivo. Ex:
`rewind(parq);`

Principais Funções para Leitura de Arquivos

Nome	Resumo
fscanf	Semelhante à função scanf, mas faz a leitura de informações de um arquivo texto e não da entrada do teclado.
fgetc	Semelhante à função getc, retornando o caractere lido do arquivo texto passado como parâmetro.
fgets	Semelhante à função gets, que retorna uma cadeia de caracteres lida do arquivo texto passado como parâmetro até encontrar o \n ou até que o valor máximo de caracteres seja alcançado.

Lendo Dados

```
FILE *f = fopen ("teste.txt", "r");
while (fscanf(f, "%c", &c) != EOF)
    printf ("%c", c);
fclose (f);
```

- ◆ **fopen** retorna um descritor de arquivos (*stream pointer*)
- ◆ **fscanf** é semelhante à função **scanf**
- ◆ **fclose** fecha o arquivo

Lendo Dados

```
int fgetc (FILE* fp);
```

- ◆ Recebe o ponteiro para o arquivo como parâmetro.
Retorna o caractere lido com sucesso.

```
/Lê o próximo caractere do arquivo  
letra = fgetc (fp);
```

```
char* fgets (char* s, int n, FILE* fp);
```

- ◆ Recebe uma cadeia de caracteres para armazenar o conteúdo lido, a quantidade máxima de caracteres a serem lidos e o ponteiro para o arquivo como parâmetros.
- ◆ Retorna a cadeia de caracteres lidos com sucesso ou NULL em caso de erro.

```
//Lê uma cadeia de 256 caracteres (incluindo  
o \0)  
linha = fgets (linha, 256, fp);
```

Escrevendo dados em Arquivo

- ◆ Principais funções para escrita em modo texto são:

Nome	Resumo
fprintf	Semelhante à função printf, mas faz a escrita das informações em um arquivo texto e não na saída padrão (monitor).
fputc	Semelhante à função putc, mas também faz a escrita do caractere no arquivo texto e não na saída padrão (monitor).

Escrevendo dados em Arquivo

```
FILE *fr = fopen ("teste.txt", "r");
FILE *fw = fopen ("saída.txt", "w");
while (fscanf(fr, "%c", &c) != EOF)
    fprintf(fw,"%c", c);
fclose(fr);
fclose(fw);
```

Escrevendo dados em Arquivo

```
int fputc (int c, FILE* fp);
```

- ◆ Recebe o caractere a ser escrito e o ponteiro para o arquivo como parâmetros.
- ◆ Retorna o próprio caractere escrito com sucesso ou o código EOF em caso de erro na gravação.

```
//Escreve um caractere no arquivo  
letra = fputc (letra, fp);
```

Exemplo

```
#include<stdio.h>
int main() {
    FILE *fr, *fw;
    char c;
    fr = fopen ("teste.txt", "r");
    fw = fopen ("saída.txt", "w");
    if (fr == NULL) {
        printf ("\n\nNão foi possível abrir o arquivo teste.txt
para leitura\n");
        return 1;
    }
    if (fw == NULL) {
        printf ("\n\nNão foi possível abrir o arquivo saída.txt
para escrita\n");
        return 2;
    }
    while (fscanf(fr, "%c", &c) != EOF)
        fprintf(fw, "%c", c);
    fclose(fr);
    fclose(fw);
    return 0;
}
```

Exercício

- Faça um programa que leia um arquivo texto qualquer e retorne a sua quantidade de linhas.