

CAP. IX - MANIPULAÇÃO DE ARQUIVOS

9.1 – Generalidades sobre arquivos

9.2 – Abertura e fechamento de arquivos

9.3 – Arquivos textos e arquivos binários

9.4 – Leitura e escrita em arquivos binários

9.5 – Procura direta em arquivos binários

9.1 - Generalidades sobre Arquivos

- Certos dados, numerosos ou não, precisam existir, **independentemente da execução** de qualquer programa e do computador **estar ligado**.
- **Exemplos:**
 - Catálogo dos livros de uma **biblioteca** (assunto, nome, autor, ...)
 - Relação dos clientes de um **banco** (nº da conta, nome, endereço, saldo, tipo da conta, ...)
- Tais dados devem ser armazenados **fora da memória principal**, num meio físico **não volátil**: disquetes, discos rígidos, fitas, laser, cartões, etc...
- **Nesta matéria:** manipulação de arquivos em **disquetes e discos rígidos**, usando **Linguagem C**.

• Operações com arquivos:

– Abertura, leitura, gravação, procura direta, fechamento.

- Todo **arquivo do programa**, durante a execução, possui um **ponteiro** apontando para o elemento (byte) **a ser lido** ou **a ser reescrito** nesse arquivo.
- Os bytes de um arquivo são **numerados a partir de zero**.

9.2 – Abertura e Fechamento de Arquivos

- Todo arquivo do diretório do usuário precisa ser **associado a uma variável** do tipo **FILE** do programa, para ser aberto e acessado por esse programa.
- **Comandos** para essa **associação e abertura** de arquivos:

```
FILE *fl;  
fl = fopen ( "nome_do_arquivo", "modo");
```

- Comando para **fechar** o arquivo aberto:

```
fclose (fl);
```

• Tabela de modos:

Modo	Significado
“r”	Abrir arquivo <i>texto</i> para leitura
“w”	Abrir arquivo <i>texto</i> para escrita
“a”	Abrir arquivo <i>texto</i> para aumentá-lo
“rb”	Abrir arquivo <i>binário</i> para leitura
“wb”	Abrir arquivo <i>binário</i> para escrita
“ab”	Abrir arquivo <i>binário</i> para aumentá-lo
“r+”	Abrir arquivo <i>texto</i> para leitura e escrita
“w+”	Abrir arquivo <i>texto</i> para escrita e leitura
“rb+”	Abrir arquivo <i>binário</i> para leitura e escrita
“wb+”	Abrir arquivo <i>binário</i> para escrita e leitura

- A abertura para **leitura** de um arquivo que **não existe** falhará; a função *fopen* retornará NULL.
- A abertura para **escrita** de um arquivo não existente **criará** esse arquivo; se ele já existe, será **apagado e reescrito**.
- A abertura para **aumentar** um arquivo não existente **criará** esse arquivo; se ele já existe, a escrita **começa em seu final**.

9.3 – Arquivos Textos e Arquivos Binários

- Sejam as seguintes **declarações em C** para a estrutura **EletoDoméstico**:

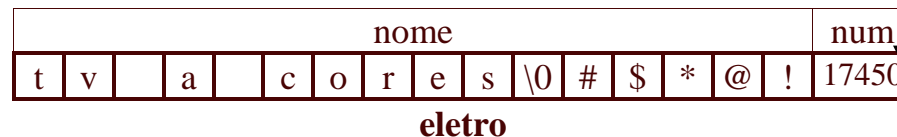
```
typedef char cadeia [16];
struct EletroDomestico {
    cadeia nome;
    int num;
};
typedef struct EletroDomestico EletroDomestico;
```

- As funções *fscanf* e *fprintf* são usadas para **ler e escrever** textos em arquivos abertos como **arquivos textos**.
- **Exemplo:** sejam as seguintes declarações:

```
FILE *fl;
fl = fopen ("arqelet", "w+");
EletroDomestico eletro;
```

Em binário de 2 bytes

– Supondo que o conteúdo da variável **eletro** seja:



– Então o comando

```
fprintf (fl, "%s%7d", eletro. nome, eletro. num);
```

– Escreverá no arquivo **arqelet** o seguinte:

t	v		a		c	o	r	e	s			1	7	4	5	0
---	---	--	---	--	---	---	---	---	---	--	--	---	---	---	---	---

– O formato **%s** escreve todos os caracteres até encontrar o ‘\0’, que não é escrito.

– O formato **%7d** converte o binário **17450** na cadeia **“17450”**, escrevendo-a precedida de 2 espaços em branco.

– Agora, supondo que o arquivo **arqelet** seja aberto como **arquivo binário**:

```
fl = fopen ("arqelet", "wb+");
```

– E que o conteúdo da variável **eletro** seja o mesmo, o comando:

```
fwrite (&eletro, sizeof (EletroDomestico), 1, fl);
```

– Deixará no arquivo **arqelet** o seguinte conteúdo:

t	v		a		c	o	r	e	s	\0	#	\$	*	@	!	*	D
---	---	--	---	--	---	---	---	---	---	----	---	----	---	---	---	---	---

[68, 42] → [01000100, 00101010] → 17450

- **fwrite** (&var, tamanho, n, fl);

Escreve no arquivo associado a **fl**, a partir da posição de seu ponteiro, (**n*tamanho**) bytes copiados da memória, a partir do endereço da variável **var**.

- Tipicamente, **tamanho** é o número de bytes de uma estrutura ou elemento de vetor e **n** é o número de unidades dessa estrutura ou de elementos desse vetor a serem copiados.

- A região especificada da memória é copiada sem qualquer conversão (**fwrite** é mais rápido que **fprintf**).

- Neste capítulo só serão abordados **arquivos binários**.

9.4 – Leitura e Escrita em Arquivos Binários

9.4.1 – Formação de arquivos binários

- Neste tópico, será apresentado um programa completo, fragmentado conforme o assunto a ser tratado.

- Nos fragmentos a seguir, são apresentadas as declarações globais, a função **main** () e a função usada para a **formação** de um arquivo de dados sobre **eletrodomésticos**.

- Para essa formação, os dados são **digitados** e a seguir **inseridos** no arquivo.

- Include's e declarações globais:

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

```
typedef char cadeia [16];
```

```
struct EletroDomestico {
    cadeia nome; int num;
};
```

```
typedef struct EletroDomestico EletroDomestico;
```

- Protótipos das funções:

```
void FormarArquivo (FILE *);
void EscreverArquivo (FILE *);
long filesize (FILE *);
void OrdenarArquivo (FILE *);
void FazerProcuras (FILE *);
char Membro (cadeia, long, long, FILE *);
```

- Função *main*: **abre e forma** um arquivo de eletrodomésticos, **ordena** esse arquivo alfabeticamente, **escreve** no vídeo seu conteúdo e faz **procuras** de vários itens nesse arquivo.

```
void main () {
    FILE *fl; long fz;
    fl = fopen ("arquetros", "wb+");
    FormarArquivo (fl);
    fz = filesize (fl);
    printf ("\n\nTamanho do arquivo: %ld", fz);
    printf ("\n\nNumero de itens: %ld\n", fz / sizeof
        (EletroDomestico));
    printf ("\n\nArquivo desordenado:\n\n");
    EscreverArquivo (fl);
    OrdenarArquivo (fl);
    printf ("\n\nArquivo ordenado:\n\n");
    EscreverArquivo (fl);
    FazerProcuras (fl);
    fclose (fl);
}
```

- Função *FormarArquivo*: recebe dados do teclado sobre vários eletrodomésticos, **inserido-os** no arquivo.

```
void FormarArquivo (FILE *fl) {
    EletroDomestico eletro; char c;
    printf ("Inserir eletro-domestico? (s/n): ");
    c = getche ();
    while (c == 's' || c == 'S') {
        fflush (stdin);
        printf ("\nNome: ");
        gets (eletro. nome);
        printf ("Numero de unidades: ");
        scanf ("%d", &eletro. num);
        fwrite (&eletro, sizeof (eletro), 1, fl);
        printf ("\nInserir eletro-domestico? (s/n): ");
        c = getche ();
    }
}
```

- **fflush (stdin)**: esvazia *stdin* (buffer do teclado).
- **gets (eletro. nome)**: recebe do buffer do teclado uma cadeia de caracteres encerrada por *enter* e a armazena no campo **nome** da estrutura **eletro**.

9.4.2 – Leitura de arquivos binários

- Função *EscreverArquivo*: lê as informações sobre cada eletrodoméstico armazenado no arquivo, **escrevendo-as** no vídeo.

```
void EscreverArquivo (FILE *fl) {
    EletroDomestico eletro;
    rewind (fl);
    while (! feof (fl)) {
        fread (&eletro, sizeof (eletro), 1, fl);
        if (! feof (fl))
            printf ("Eletrodomestico: %-17s; Quantidade:
%5d;\n",
                eletro. nome, eletro. num);
    }
}
```

- **fread (&var, tamanho, n, fl)**;

Lê do arquivo associado a **fl**, a partir da posição de seu ponteiro, **n*tamanho** bytes, copiando-os na memória, a partir do endereço da variável **var**.

- **rewind (fl)**: posiciona o ponteiro no primeiro byte do arquivo associado a **fl**, ou seja no byte de número **zero**.

- **feof (fl):** retorna 1 se a marca de fim do arquivo associado a **fl** tiver sido encontrada; caso contrário, retorna zero.

Para encontrar essa marca, não basta o ponteiro atingir a posição após o último byte do arquivo; é necessário uma leitura dessa marca.

9.5 – Procura Direta em Arquivos Binários

- Função **fseek (fl, desl, ref):** coloca o ponteiro do arquivo associado a **fl** numa posição dada pelo valor do argumento **ref** e, em seguida, desloca-o para outra posição distante **desl** bytes da primeira.
 - **desl:** deslocamento a partir da posição relacionada com o valor do argumento **ref**;
 - **ref:** referência para fazer o deslocamento; seus valores podem ser 0, 1, ou 2, correspondendo respectivamente ao início do arquivo, à posição atual do ponteiro ou ao final do arquivo.

- **Exemplos:**

`fseek (fl, i, 0):` posiciona o ponteiro no byte de número **i**.

`fseek (fl, i, 1):` desloca o ponteiro **i** posições à frente (**i** pode ser negativo).

`fseek (fl, -i, 2):` posiciona o ponteiro **i** bytes antes do fim do arquivo; caso **i** seja **zero**, o ponteiro vai para o fim.

- Função **ftell (fl):** retorna o número do byte apontado pelo ponteiro do arquivo associado a **fl**.
- As funções *fseek* e *ftell* pertencem ao *stdio.h*.
- Função *filesize*: retorna o número de bytes de um arquivo.

```
long filesize (FILE *fl) {  
    fseek (fl, 0, 2);  
    return ftell (fl);  
}
```

- Função **OrdenarArquivo**: ordena um arquivo de eletrodoméstico em ordem alfabética, pelo **bubble-sort**.

```

void OrdenarArquivo (FILE *fl) {
    EletroDomestico eletro1, eletro2, eletroaux;
    char trocou; long i, p;
    p = filesize (fl) / sizeof (eletro1) - 2;
    do {
        trocou = 0; fseek (fl, 0, 0);
        if (p >= 0) fread (&eletro1, sizeof (eletro1), 1, fl);
        for (i = 0; i <= p; i++) {
            fseek (fl, (i+1) * sizeof (eletro2), 0);
            fread (&eletro2, sizeof (eletro2), 1, fl);
            if (strcmp (eletro1. nome, eletro2. nome) > 0)
            {
                trocou = 1; eletroaux = eletro1;
                eletro1 = eletro2; eletro2 = eletroaux;
                fseek (fl, i * sizeof(eletro1), 0);
                fwrite (&eletro1, sizeof (eletro1), 1, fl);
                fwrite (&eletro2, sizeof (eletro2), 1, fl);
            }
            eletro1 = eletro2;
        }
        p--;
    } while (trocou && p >= 0);
}

```

- Função **FazerProcuras**: procurar no arquivo vários nomes de eletrodomésticos lidos do teclado:

```

void FazerProcuras (FILE *fl) {
    char c; cadeia nome;
    printf ("Procurar eletro-domestico? (s/n): ");
    c = getche ();
    while (c == 's' || c == 'S') {
        fflush (stdin);
        printf ("\nNome: ");
        gets (nome);
        if (Membro (nome, 0, filesize (fl) /
                    sizeof (EletroDomestico) - 1, fl))
            printf ("%s estah no arquivo\n\n", nome);
        else
            printf ("%s nao estah no arquivo\n\n", nome);
        printf ("\nProcurar eletro-domestico? (s/n): ");
        c = getche ();
    }
}

```

- Função *Membro*: verifica se um eletrodoméstico pertence ao arquivo (usa procura binária).

```
char Membro (cadeia nome, long i, long f, FILE *fl) {  
    long med; EletroDomestico eletro; char m;  
    m = 0;  
    if (i <= f) {  
        med = (i + f) / 2;  
        fseek (fl, med * sizeof (eletro), 0);  
        fread (&eletro, sizeof (eletro), 1, fl);  
        if (strcmp (nome, eletro. nome) == 0)  
            m = 1;  
        else if (strcmp (nome, eletro. nome) < 0)  
            m = Membro (nome, i, med-1, fl);  
        else  
            m = Membro (nome, med+1, f, fl);  
    }  
    return m;  
}
```