Arquivos

Quando for manipular um arquivo deve fazer a associação de uma variável ponteiro a um arquivo físico (FILE). A associação será feita pela função fopen().
Modo de abertura do arquivo
r Abre arquivo já existente para leitura
w Abre (cria se necessário) para gravação. Caso já exista ele sobrescreve.
a Abre (cria se necessário) arquivo, mas, vai até o final dele e depois começa a gravar – acrescenta.
rt Leitura de arquivo texto
rb Leitura de arquivo binário
- + Abre e grava no arquivo (cuidado r+ abre e grava, mas no começo!)

```
FILE *arq;
--------------------------------------------------------------------------
arq = fopen("nome_arq", "modo" );

Arq = fopen( var_nome, const_modo );
Ou
Arq = fopen( "C:\\LOCAL\\ARQUIVO.TEXT", w);

--------------------------------------------------------------------------
fclose(arq);
```

Os arquivos podem ser de modo:
Texto – é composto por sequência de caracteres (tabela ASCII)
Binário – é formado por sequencias de bytes.
Texto: pode ser lido por qualquer um e editado em qualquer programa
Binário: é seguro – para gravar dados – menos espaço.

Fopen() – Função que encontra o arquivo.
Fclose()- Função que atualiza no disco a informação que está no buffer (memória) e depois libera a memória.

Arquivos em modo binário
Gravar
fwrite (o endereço onde está o conteúdo a ser gravado,
tamanho em bytes a ser gravado,
o número de elementos que vai gravar
o ponteiro que indica para onde vai copiar

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

struct  entrada
{
   char nome[40];
   float valor;
} devedores[4];

main()
{
   int i;
    FILE *fp;
   for (i=0; i<4; i++)
   {
      fflush(stdin); printf("Digite o nome: "); gets(devedores[i].nome);
      printf("Valor da Divida: R$"); scanf("%f", &devedores[i].valor);
   }

                fp = fopen("devedores.txt","w");  //Etapa de abertura e gravação do arquivo
                if (fp == NULL)
                {
                  printf("Erro de abertura do arquivo");
                  exit(1);
                }
                fwrite(&devedores,   sizeof(devedores),   1,   fp);
//se for um devedor só – não houver uma matriz de estrutura não coloca o &devedores

                fclose(fp);
                _sleep(3000);
 }


F= 3.1415
Fwrite (&f, sizeof(f), 1, fp)
Float m[10][2];
Fwrite (m, sizeof(m), 1, fp)


Arquivos em modo binário
Leitura
fread(o endereço onde será armazenado o conteúdo lido,
        tamanho em bytes a ser avançado o cursor,
        o número de elementos que vai ler,
        o ponteiro que indica o nome do arquivo que tem o conteúdo


struct  entrada
{
   char nome[40];
```

```
        float valor;
} devedores[4];

main()
{
   int i;

   FILE *fp;


   printf("Segunga etapa - relacao de devedores ");
   fp = fopen("devedores.txt","r");

   fread(&devedores, sizeof(devedores),1,fp);

   for (i=0; i<4; i++)
   {
      printf("\n%s  - R$ %.2f",devedores[i].nome,devedores[i].valor);
   }
        getch();
        fclose(fp);
}
```

Pesquisa em arquivo binário

SEEK_CUR – posição corrente
SEEK_SET – início do arquivo
SEEK_END – final do arquivo

fseek(nome do arquivo posicionado, quantos bytes irá avançar, qual a relação de avanço)

```
fseek( fp, 3*sizeof(devedores), SEEK_SET);
fread(&devedores, sizeof(devedores), 1, fp);
```

Arquivos em modo texto
Leitura
fscanf
fgetc
fgets

Gravação
fprintf() –grava um sequência de dados – retorna
o total de bytes gravados

```
{       FILE *fp;
                char texto[20];
                fp = fopen("alo.txt","r");
                fscanf (fp,"%s",texto);
                printf ("%s",texto);
                fclose(fp);
```

```
                    _sleep(4000);
}

{
FILE *fp;
char texto[ ] = {“* Programação *"};
fp = fopen("alo.txt","w");
fprintf(fp,"%s",texto);
fclose(fp);
_sleep(4000);
}


int main()
{
    FILE *arquivo;
    int t,c=0,r ;
    printf ("Digite qual a tabuada...:");  scanf ("%d",&t);
    arquivo = fopen("tabuada.xls","w");
    printf (arquivo,"<table>");

    do
    {
        r=c*t;
        fprintf (arquivo,"<tr><td>%d x %d = %d</td></tr>",c,t,r);
        c++;
     } while (c<=10);

     fprintf (arquivo,"</table>");
     fclose(arquivo);
     printf ("Arquivo gravado com sucesso");
     _sleep(4000);
   }
```