Manipulação de Arquivos

Devemos iniciar nossa explanação pelo conceito de arquivo:

Arquivo é uma unidade lógica utilizada para armazenar dados em disco ou em qualquer outro dispositivo externo de armazenamento. Pode-se abrir, fechar, ler, escrever ou apagar um arquivo.

A linguagem C manipula tanto arquivos quanto dispositivos de I/O, se utilizando do conceito de “ponteiro para arquivo”. É disponibilizada uma série de funções para trabalhar com este conceito, cujos protótipos estão reunidos em **stdio.h**.

Manipulação de Arquivos

A definição do “ponteiro para arquivo” também está no arquivo **stdio.h**.

Podemos declarar um ponteiro de arquivo da seguinte maneira:

```
FILE *p;
```

Os arquivos podem ser classificados em:

- binários;
- texto.

Manipulação de Arquivos

- fopen()

Esta é a função de abertura de arquivos. Seu protótipo é:

FILE *fopen (char *nome_do_arquivo,char *modo);

Modo	Significado
"r"	Abre um arquivo texto para leitura. O arquivo deve existir antes de ser aberto.
"w"	Abrir um arquivo texto para gravação. Se o arquivo não existir, ele será criado. Se já existir, o conteúdo anterior será destruído.
"a"	Abrir um arquivo texto para gravação. Os dados serão adicionados no fim do arquivo ("append"), se ele já existir, ou um novo arquivo será criado, no caso de arquivo não existente anteriormente.

Modo	Significado
"rb"	Abre um arquivo binário para leitura. Igual ao modo "r" anterior, só que o arquivo é binário.
"wb"	Cria um arquivo binário para escrita, como no modo "w" anterior, só que o arquivo é binário.
"ab"	Acrescenta dados binários no fim do arquivo, como no modo "a" anterior, só que o arquivo é binário.
"r+"	Abre um arquivo texto para leitura e gravação. O arquivo deve existir e pode ser modificado.
"w+"	Cria um arquivo texto para leitura e gravação. Se o arquivo existir, o conteúdo anterior será destruído. Se não existir, será criado.
"a+"	Abre um arquivo texto para gravação e leitura. Os dados serão adicionados no fim do arquivo se ele já existir, ou um novo arquivo será criado, no caso de arquivo não existente anteriormente.
"r+b"	Abre um arquivo binário para leitura e escrita. O mesmo que "r+" acima, só que o arquivo é binário.
"w+b"	Cria um arquivo binário para leitura e escrita. O mesmo que "w+" acima, só que o arquivo é binário.
"a+b"	Acrescenta dados ou cria uma arquivo binário para leitura e escrita. O mesmo que "a+" acima, só que o arquivo é binário

Manipulação de Arquivos

Poderíamos então, para abrir um arquivo binário para escrita, escrever:

```
FILE *fp;
fp=fopen ("exemplo.bin","wb");
if (!fp)
        printf ("Erro na abertura do arquivo.");
```

A condição **!fp** testa se o arquivo foi aberto com sucesso porque no caso de um erro a função **fopen()** retorna um ponteiro nulo (**NULL**).

Manipulação de Arquivos

Uma vez aberto um arquivo, vamos poder ler ou escrever nele utilizando as funções que serão apresentadas a seguir.

Toda vez que estamos trabalhando com arquivos, há uma espécie de posição atual no arquivo. Esta é a posição de onde será lido ou escrito o próximo dado. Normalmente, num acesso seqüencial a um arquivo, não temos que mexer nesta posição pois quando lemos um dado a posição no arquivo é automaticamente atualizada. Num acesso randômico teremos que mexer nesta posição.

UNIVASF

Manipulação de Arquivos

- fclose

Quando acabamos de usar um arquivo que abrimos, devemos fechá-lo. Para tanto usa-se a função **fclose()**, cujo protótipo é:

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

O ponteiro **fp** passado à função **fclose()** determina o arquivo a ser fechado. A função retorna zero no caso de sucesso.

Manipulação de Arquivos

Fechar um arquivo faz com que qualquer dado que tenha permanecido no "buffer" associado ao fluxo de saída seja gravado.

A função exit() fecha todos os arquivos que um programa tiver aberto.

Manipulação de Arquivos

- putc

A função putc é a primeira função de escrita em arquivo que veremos. Seu protótipo é:

```
int putc (int ch, FILE *fp);
```

A referida função escreve um caractere no arquivo apontado por fp.

OBS.: **fputc()** é equivalente a **putc()**

```
#include <stdio.h>
int main()
{
      FILE *fp;
      char string[100];
      int i;
      fp = fopen("arquivo.txt","w");
      if(!fp)
      {
         printf( "Erro na abertura do arquivo");
         exit(1);
      }
      printf("Entre com a string a ser gravada no arquivo:");
      gets(string);
      for(i=0; string[i]; i++)
         putc(string[i], fp);
      fclose(fp);
      return 0;
}
```

Manipulação de Arquivos

Exercício:

Com o que vimos até o momento sobre manipulação de arquivos. Construa um função em C que possua a capacidade de escrever um inteiro em um arquivo binário. Escreva um programa que se utiliza adequadamente da função que você projetou.