

Manipulação de Arquivos

Podemos escrever e ler blocos de dados em um determinado arquivo. Para tanto, a linguagem C, disponibiliza as funções **fread()** e **fwrite()**.

- fread

O protótipo de **fread()** é:

```
unsigned int fread (void *buffer,  
int numero_de_bytes, int count, FILE *fp);
```

A função retorna o número de unidades efetivamente lidas. Este número pode ser menor que *count* quando o fim do arquivo for encontrado ou ocorrer algum erro.

Manipulação de Arquivos

- fwrite

A função `fwrite()` funciona como `fread()`, porém escrevendo no arquivo. Seu protótipo é:

```
unsigned int fwrite(void *buffer,  
int numero_de_bytes, int count, FILE *fp);
```

A função retorna o número de itens escritos. Este valor será igual a `count`, a menos que ocorra algum erro.

O exemplo a seguir ilustra o uso de `fwrite()` e `fread()` para gravar e posteriormente ler uma variável `float` em um arquivo binário.

```
#include <stdio.h>
int main() {
    FILE *pf;
    float pi = 3.1415, pilido;
    if(!(pf = fopen("arquivo.bin", "wb"))) exit(1);
    if(fwrite(&pi, sizeof(float), 1,pf) != 1) {
        printf("Erro na escrita do arquivo!");
        exit(2); }
    fclose(pf);
    if(!(pf = fopen("arquivo.bin", "rb"))) exit(1);
    if(fread(&pilido, sizeof(float), 1,pf) != 1) {
        printf("Erro na leitura do arquivo!");
        exit(3); }
    printf("\nO valor de PI, lido do arquivo e': %.4f", pilido);
    fclose(pf); return(0);
```

Manipulação de Arquivos

Exercício/Exemplo:

Construa funções na linguagem C que manipulem um determinado arquivo que contém registros, efetuando operações de escrita e leitura respectivamente. Um registro contém código, nome e salário de um funcionário específico. Uma função deve escrever e a outra ler um determinado registro no arquivo especificado.

```

#include <stdio.h>
typedef struct {
    long int codigo;
    char nome[20];
    float salario;
} t_reg;
...
void escrever_reg (FILE *fp, t_reg *r)
{
    fwrite(r,sizeof(t_reg),1,fp);
}
t_reg ler_reg (FILE *fp)
{
    t_reg temp;
    fread(&temp,sizeof(t_reg),1,fp);
    return temp;
}

```

Manipulação de Arquivos

- fseek

Para se fazer procuras e acessos randômicos em arquivos usa-se a função `fseek()`. Esta move a posição corrente de leitura ou escrita no arquivo de um valor especificado, a partir de um ponto especificado. Seu protótipo é:

```
int fseek (FILE *fp, long int numbytes, int origem);
```

Nome	Valor	Significado
SEEK_SET	0	Início do arquivo
SEEK_CUR	1	Ponto corrente no arquivo
SEEK_END	2	Fim do arquivo

Manipulação de Arquivos

Exercício:

Com base no exercício do slide 397, construa mais uma função, capaz de inserir um determinado registro em uma determinada posição no arquivo.

Observação: A função deve receber o nome do arquivo a ser manipulado e o número de registros contido no mesmo entre os seus parâmetros.

Manipulação de Arquivos

- **rewind**

A função **rewind()** de protótipo

```
void rewind (FILE *fp);
```

retorna a posição corrente do arquivo para o início.

- **remove**

Protótipo:

```
int remove (char *nome_do_arquivo);
```

Apaga um arquivo especificado, retornando zero caso seja bem sucedida; caso contrário, retorna um não nulo.

Manipulação de Arquivos

A linguagem C permite ao programador ler e escrever em arquivos da maneira padrão com a qual já vínhamos lendo e escrevendo na tela.

- **fprintf**

A função **fprintf()** funciona como a função **printf()**. A diferença é que a saída de **fprintf()** é um arquivo e não a tela do computador. Seu Protótipo é:

```
int fprintf (FILE *fp,char *str,...);
```

Como já poderíamos esperar, a única diferença do protótipo de **fprintf()** para o de **printf()** é a especificação do arquivo destino através do ponteiro de arquivo.

Manipulação de Arquivos

- fscanf

A função **fscanf()** funciona como a função **scanf()**. A diferença é que **fscanf()** lê de um arquivo e não do teclado do computador. O seu protótipo é:

```
int fscanf (FILE *fp, char *str,...);
```

Como já poderíamos esperar, a única diferença do protótipo de **fscanf()** para o de **scanf()** é a especificação do arquivo origem através do ponteiro de arquivo.

```
/* Exemplo das funções apresentadas */  
#include <stdio.h>  
#include <string.h>  
main() {  
    FILE *p;  
    char str[80],c;  
    fprintf(stdout,"Entre com um nome para o arquivo: ");  
    fgets(str,80,stdin);  
    if (str[strlen(str)-1]=='\n')  
        str[strlen(str)-1]='\0';  
    if (!(p = fopen(str,"w+")))  
        exit(1);  
    fprintf(p,"\nEste e um arquivo chamado: %s\n", str);  
    rewind(p);  
    while (!feof(p)) {  
        fscanf(p,"%c",&c);  
        fprintf(stdout,"%c",c); }  
    fclose(p);  
    return(0);
```

Manipulação de Arquivos

Exercício:

Construa uma função em C que possua o seguinte protótipo:

```
void print (char *arq, float *m, int i, int j);
```

onde *m* é o endereço inicial dos elementos de uma determinada matriz, *i* e *j* são, respectivamente, o número de linhas e de colunas da referida matriz, e *arq* conterà o nome de uma arquivo texto que deve ser criado e neste deve ser escrita a matriz mencionada com layout adequado.