



Descrição: MANIPULAÇÃO DE ARQUIVOS UTILIZANDO A LINGUAGEM C.

Software utilizado: Dev-C++ 4.9.9.2 (Licenciado pela GPL)

1º Parte: Inclusão em um arquivo .txt

```
// Módulo de Inclusão em arquivo
#include<stdio.h>
#include<conio.h>
int main(){
    int id; // Identificação do produto
    char product[50]; // variável para produto com nome de até 50 char's
    float price; // variável para guardar o preço do produto
    FILE *fileptr; // ponteiro do tipo FILE para o arquivo

    printf("INCLUDE NEW PRODUCT\n"); // Informa que é o para incluir produtos
    fileptr = fopen("DBproducts.txt", "a"); // Abre o arquivo txt

    if(fileptr == NULL){ //testar se o arquivo pode ser aberto
        printf("\nO arquivo nao pode ser aberto!");
    }
    else{
        printf("\nID: "); scanf("%d",&id); // Lê ID do produto

        printf("PRODUCT: "); fflush(stdin); gets(product); // Lê nome do produto

        printf("PRICE: ");scanf("%f",&price); // Lê preço do produto
    }

    fprintf(fileptr,"%d,%s,%.2f#\n",id, product, price); // Escreve no Arquivo

    fprintf(stdout,"Last ID included:\n%d %s %f\n\n", id, product, price); // Imprime na Stream de
saída padrão

    fclose(fileptr); //Fecha o Arquivo txt
    getch(); // só para não fechar o programa
    return 0; // Modo correto no DEV pois o DEV-C++ a função main retorna int por default
}
```

Análise do código:

FILE *fileptr;

Declaração de um ponteiro do tipo FILE. Este ponteiro vai representar o arquivo DBproducts.txt no programa.

fileptr = fopen("DBproducts.txt", "a");

fopen recebe 2 parâmetros do tipo string fopen("filename.txt","modo") e retorna um ponteiro de arquivo. Caso o arquivo não exista será criado.

No modo "a" anexa ao arquivo criado. Existem outros modos.

fprintf(fileptr,"%d,%s,%.2f#\n",id, product, price);

A função fprintf escreve no arquivo apontado pelo ponteiro fileptr. Não esqueça de usar um separador, isto será útil quando estivermos na parte de leitura de arquivo, **guarde isto!**. E o \n para ir para próxima linha no arquivo. *Não pode ser usado espaço para separar valores no arquivo.*

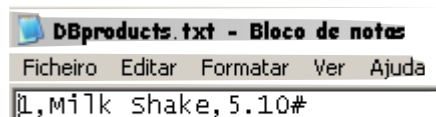
fprintf(stdout,"Last ID included:\n%d %s %f\n\n", id, product, price);

Imprime na tela o equivalente a printf("%d, %s, %f\n",id, product,price); observe que a stream padrão agora é stdout. A stream padrão de saída é o vídeo. A de entrada o teclado(stdin) no scanf.

fclose(fileptr);

Fecha o arquivo para liberar os recursos que ele consumiu do SO.

Exemplo como ficará no .txt criado:



2º Parte: Visualização formatada de dados do arquivo .txt criado.

```
// Módulo de Leitura do Arquivo txt
#include<stdio.h>
#include<string.h>
#include<conio.h>
#define max 50
int main(){
    int id;
    char *product,*price;
    FILE *fileptr;
    char temp[max]; // String temporária que armazenará a linha na função fgets

    fileptr = fopen("DBproducts.txt","r"); // abre o arquivo no modo leitura

    if(fileptr == NULL){ //testar se o arquivo pode ser aberto
        printf("\nO arquivo nao pode ser aberto!\n");
    }
    else{
        fgets(temp,max,fileptr); // pega primeira linha do txt

        printf(" ID |   PRODUCTS   | PRICE\n");

        while(!feof(fileptr)){ //Enquanto arquivo não chegue ao fim
            id = atoi(strtok(temp,",")); // pega até a virgula e tranf. para int
            product = strtok(NULL,"," ); // pega até a virgula e coloque em produto
            price = strtok(NULL,"#"); // pega até a virgula
            printf("%04d | %20s| %s", id, product , price);
            printf("\n");
            fgets(temp,max,fileptr); // pega próxima linha
        }
        fclose(fileptr); // fecha o arquivo
        getch();// só para não fechar o programa
        return 0;
    }
}
```

Análise do código:

fopen("DBproducts.txt","r");
abrir o arquivo no modo leitura("r" de read), selecionar o arquivo criado anteriormente, DBproducts.txt.

fileptr == NULL
Caso Arquivo não exista a função fopen retorna NULL. Logo o arquivo não poderá ser aberto.

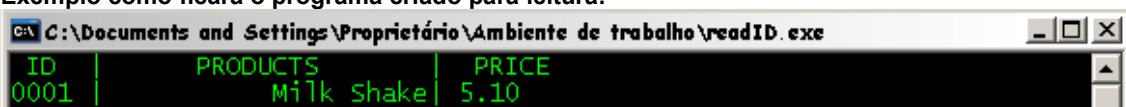
fgets(temp,max,fileptr);
Armazena a primeira linha do arquivo em um vetor temporário (temp), para que torne possível a manipulação. Não esqueça do tamanho Máximo do vetor.

!feof(fileptr)
Função que verifica se o arquivo, apontado pelo ponteiro fileptr, chegou ao fim. Neste programa esta função encontrasse em um laço que tem o seguinte objetivo: Enquanto o arquivo não chegar ao fim faça.

atoi(strtok(temp,","));
Função atoi: Transforma a string, retornada pela função strtok, em inteiro.
Função strtok: manipula a string armazenada no vetor temp. Recebe como parâmetros o vetor e uma string que será a delimitadora do dados. Para isto falei da importância de utilizar o delimitado ",", no fprintf(fileptr,"%d,%s,##\n",id, product, price) da 1º parte. A partir da extração da primeira sub-string utilizamos a constante NULL.

fclose(fileptr);
Fecha o arquivo para liberar os recursos que ele consumiu do SO.

Exemplo como ficará o programa criado para leitura:



```
C:\Documents and Settings\Proprietário\Ambiente de trabalho\readID.exe
ID |   PRODUCTS   | PRICE
0001 | Milk Shake | 5.10
```

P.S.: Arquivos .c em <http://www.geocities.com/holokaos2/monitoriac>