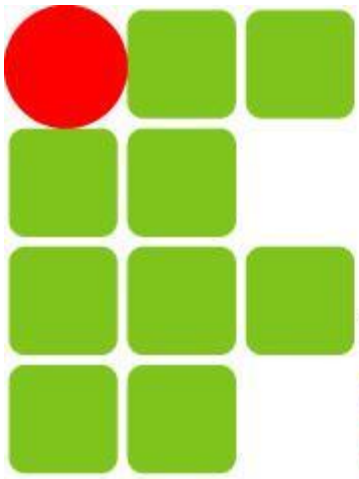

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cachoeiro de Itapemirim

Manipulação de Arquivos

Programação 1

Rafael Vargas Mesquita

http://www.ci.ifes.edu.br
ftp://ftp.ci.ifes.edu.br/informatica/rafael/

Tipo Arquivo

- Um arquivo é uma forma alternativa à entrada/saída de dados via teclado
- Para representar um arquivo utilizamos a palavra FILE (arquivo em inglês).
- As funções de manipulação de arquivos utilizam ponteiros, daí normalmente criamos variáveis do tipo FILE * e não FILE.

Funções de Manipulação de Arquivos

- *fopen*: função que abre ou cria o arquivo, e é a partir dela que toda a manipulação começa.
  - Sua assinatura é da seguinte forma:

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

- Onde:
  - nome_do_arquivo: nome do arquivo aberto, junto com seu caminho na estrutura de diretórios.
  - modo: forma ou permissão que o arquivo será aberto

Funções de Manipulação de Arquivos

- *fopen:* Esta função pode abrir um arquivo de muitas formas. O quadro ao lado ilustra essas possibilidades:

Modo	Significado
r	Abre um arquivo-texto para leitura
w	Cria um arquivo-texto para escrita
a	Anexa a um arquivo-texto
rb	Abre um arquivo binário para leitura
wb	Cria um arquivo binário para escrita
ab	Anexa a um arquivo binário
r+	Abre um arquivo-texto para leitura/escrita
w+	Cria um arquivo-texto para leitura/escrita
a+	Anexa ou cria um arquivo-texto para leitura/escrita
r+b	Abre um arquivo binário para leitura/escrita
w+b	Cria um arquivo binário para leitura/escrita
a+b	Anexa a um arquivo binário para leitura/escrita

Funções de Manipulação de Arquivos

- Exemplo (*fopen*):

```c
#include<stdio.h>

int main(){
    FILE *fp; // Declaração da estrutura

    fp = fopen ("exemplo.bin","r");

    if (!fp){
        printf ("Erro na abertura do arquivo.\n\n");
    }
    else{
        printf("Arquivo aberto com sucesso.\n\n");
    }
}
```

Funções de Manipulação de Arquivos

- *fclose*: função que fecha o arquivo. Após o uso dessa função o arquivo não será mais acessado, a não ser que seja novamente aberto.
  - Sua assinatura é da seguinte forma:

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

- Obs.: A função retorna 0 (zero) em caso de sucesso (se conseguir fechar o arquivo)

Funções de Manipulação de Arquivos

- Exemplo (*fclose*):

```
#include<stdio.h>

int main(){
    FILE *fp; // Declaração da estrutura
    /* o arquivo se chama exemplo.bin e está
    localizado no diretório corrente */
    fp = fopen ("exemplo.bin","wb");
    if (fclose(fp) == 0)
        printf ("Arquivo fechado com sucesso.\n\n");
    else
        printf("Erro no fechamento do arquivo.\n\n");
}
```

Funções de Manipulação de Arquivos

- *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.
  - Sua assinatura pode ser definida da seguinte forma:

```
int fprintf (FILE *, <mensagem>, parametros);
```

- 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.

Funções de Manipulação de Arquivos

- Exemplo (*fprintf*):

```c
#include<stdio.h>

int main(){
    FILE *fp; // Declaração da estrutura
    /* o arquivo se chama exemplo.txt e está
    localizado no diretório corrente */
    fp = fopen ("exemplo.txt","w");
    fprintf(fp, "Olá mundo.\n");
    fclose(fp);
}
```

Funções de 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.
  - Sua assinatura pode ser definida da seguinte forma:

```
int fscanf (FILE *fp, <tipos das variáveis>, &variáveis);
```

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

Funções de Manipulação de Arquivos

- Exemplo (*fscanf*):

```
#include<stdio.h>

int main(){
    FILE *fp; // Declaração da estrutura
    char str[50];
    /*o arquivo se chama exemplo.txt e está
    localizado no diretório corrente */
    fp = fopen ("exemplo.txt", "r");
    fscanf(fp, "%s", str);
    printf("A string lida eh: %s \n\n", str);
    fclose(fp);
}
```

- Obs.: Será mostrada a string: “Olá”. Isso ocorre porque o fscanf considera o espaço em branco um caracter de término de leitura.

Funções de Manipulação de Arquivos

- *feof*: EOF ("End of file") indica o fim de um arquivo. Às vezes, é necessário verificar se um arquivo chegou ao fim. Para isto podemos usar a função feof().
  - Sua assinatura pode ser definida da seguinte forma:

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

- Obs.: A função feof retorna não-zero se o arquivo chegou ao EOF, caso contrário retorna zero.

Funções de Manipulação de Arquivos

- Exemplo (*feof*):

```c
#include<stdio.h>

int main(){
    FILE *fp; // Declaração da estrutura
    char str[50];
    int cont;
    fp = fopen ("exemplo.txt", "r");
    cont = 0;
    while(!feof(fp)){
        fscanf(fp, "%c", &str[cont]);
        cont++;
    }
    str[cont-1] = '\0';
    printf("A string lida eh: %s \n\n", str);
    fclose(fp);
}
```

Funções de Manipulação de Arquivos

- *fread*: Podemos escrever e ler blocos de dados. A função fread faz a leitura de blocos de dados.
  - Sua assinatura pode ser definida da seguinte forma:

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

- Onde:
  - buffer: é a região de memória na qual serão armazenados os dados lidos.
  - numero_de_bytes: é o tamanho da unidade a ser lida.
  - count: indica quantas unidades devem ser lidas. Isto significa que o número total de bytes lidos é: numero_de_bytes*count

- Obs.: 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.

Funções de Manipulação de Arquivos

- *fwrite*: A função fwrite() funciona como a sua companheira fread(), porém escrevendo no arquivo.
  - Sua assinatura pode ser definida da seguinte forma:

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

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

Funções de Manipulação de Arquivos

- Exemplo (*fread* e *fwrite*):

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

int main(){
    FILE *pf;
    float pi = 3.1415, pilido;
    pf = fopen("arquivo.bin", "wb");
    if(fwrite(&pi, sizeof(float), 1,pf) != 1)
        printf("Erro na escrita do arquivo");
    fclose(pf);
    pf = fopen("arquivo.bin", "rb");
    if(fread(&pilido, sizeof(float), 1,pf) != 1)
        printf("Erro na leitura do arquivo");
    else
        printf("\nO valor de PI eh: %f \n\n", pilido);
    fclose(pf);
    return(0);
}
```

Funções de 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.
  - Sua assinatura pode ser definida da seguinte forma:

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

- Obs.: O parâmetro origem determina a partir de onde os numbytes de movimentação serão contados.

Funções de Manipulação de Arquivos

- *fseek*
  - O parâmetro origem é definido de acordo com a seguinte tabela:

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

Funções de Manipulação de Arquivos

- Exemplo (*fseek*):

```
#include<stdio.h>

int main(){
    FILE *fp;
    char str[50];
    int cont = 0, encontrou_espaco = 0;
    fp = fopen ("exemplo.txt", "r"); // Supor no arquivo: "Ola mundo"
    while((!feof(fp))&&(!encontrou_espaco)){
        fscanf(fp, "%c", &str[cont]);
        //Encontra espaço depois do Ola
        if (str[cont] == ' ')
            // verdadeiro
            encontrou_espaco = 1;
        cont++;
    }
    // Avança um char ( m )
    fseek(fp, sizeof(char), SEEK_CUR);
    // Lê: "undo"
    fscanf(fp, "%s", str);
    printf("A string lida eh: %s \n\n", str);
    fclose(fp);
}
```

Funções de Manipulação de Arquivos

- *rewind*: retorna a posição corrente do arquivo para o início.
  - Sua assinatura pode ser definida da seguinte forma:

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

- Obs.: Equivale ao *fseek* com:
  - origem = SEEK_SET
  - num_bytes = 0.

Funções de Manipulação de Arquivos

- Exemplo (*rewind*):

```c
#include<stdio.h>

int main(){
    FILE *fp;
    char str[50];
    int cont = 0, encontrou_espaco = 0;
    // Supor no arquivo: "Ola mundo"
    fp = fopen ("exemplo.txt", "r");
    while((!feof(fp)) && (!encontrou_espaco)){
        fscanf(fp, "%c", &str[cont]);
        // Encontra espaço depois Ola
        if (str[cont] == ' ')
            // verdadeiro
            encontrou_espaco = 1;
        cont++;
    }
    rewind(fp);
    fscanf(fp, "%s", str); // Lê: "Ola"
    printf("A string lida é: %s ", str);
    fclose(fp);
}
```

Funções de Manipulação de Arquivos

- *remove*: Apaga um arquivo especificado.
  - Sua assinatura pode ser definida da seguinte forma:

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

Funções de Manipulação de Arquivos

- Exemplo (*remove*):

```
#include<stdio.h>

int main(){
    remove("exemplo.txt" );
}
```

Bibliografia

- SANTOS, Henrique José. Curso de Linguagem C da UFMG, apostila.
- FORBELLONE, André Luiz. Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados. São Paulo: MAKRON, 1993.