

Manipulação de Arquivos

- fgets

Para se ler uma string num arquivo podemos usar **fgets()** cujo protótipo é:

```
char *fgets (char *str, int tamanho, FILE *fp);
```

A função lê a string até que um caracter de nova linha seja lido ou *tamanho-1* caracteres tenham sido lidos. Se o caracter de nova linha ('\n') for lido, ele fará parte da string, o que não acontecia com `gets`. A string resultante sempre terminará com '\0' (por isto somente *tamanho-1* caracteres, no máximo, serão lidos).

Manipulação de Arquivos

A função `fgets` é semelhante à função `gets()`, porém, além dela poder fazer a leitura a partir de um arquivo de dados e incluir o caracter de nova linha na string, ela ainda especifica o tamanho máximo da string de entrada. Como vimos, a função `gets()` não tinha este controle, o que poderia acarretar erros de "estouro de buffer". Portanto, levando em conta que o ponteiro *fp* pode ser substituído por *stdin*, como vimos anteriormente, uma alternativa ao uso de `gets` é usar a seguinte construção:

```
int tamanho=11;  
char str[11];  
fgets (str, tamanho, stdin);
```

Manipulação de Arquivos

Exercício:

Construa um programa em C que receba o nome de um arquivo, fornecido pelo usuário, através da entrada padrão, e efetue a abertura do mesmo para uma leitura. O nome do arquivo fornecido deve possuir no máximo 20 caracteres.

Manipulação de Arquivos

- fputs

A função fputs() escreve uma string num arquivo. Seu protótipo é:

```
char *fputs (char *str, FILE *fp);
```

Manipulação de Arquivos

- fflush

Seu protótipo é:

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

A função **fflush()** esvazia o buffer do arquivo referenciado pelo ponteiro passado como argumento, se o arquivo passado for um arquivo de saída o conteúdo do buffer é gravado no mesmo. Se a função for chamada sem parâmetros, os buffers de todos os arquivos abertos para saída serão descarregados. A função devolve 0 para indicar sucesso, caso contrário devolve EOF.

Manipulação de Arquivos

- **error**

A função **error()** determina se uma operação com arquivo produziu um erro.

Seu protótipo é:

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

A função **error()** retorna um inteiro não nulo se ocorreu um erro durante a última operação no arquivo; caso contrário retorna zero.

```
/* Exemplo da utilização da função ferror() */  
#include <stdio.h>  
int main()  
{  
    FILE *pf;  
    char string[100];  
    if(!(pf = fopen("arquivo.txt","wb")))  
        exit(1);  
    do  
    {  
        printf("\nDigite uma nova string. Para terminar,  
        digite <enter>: ");  
        fgets(string,100,stdin);  
        fputs(string, pf);  
        if(ferror(pf))  
            exit(1);  
    } while (strlen(string) > 1);  
    fclose(pf);  
    return 0;
```

Manipulação de Arquivos

Exercício:

Construa um programa em C que receba da linha de comando, com a qual o programa foi executado, o nome de um arquivo texto existente, o qual deve ser analisado, afim de se determinar quantas linhas o mesmo tem, após a referida análise, o programa deve incluir uma nova linha no final do arquivo contendo o número de linhas que o mesmo possuía. Caso ocorra algum erro durante o processamento, o mesmo deve ser identificado através da exibição de uma mensagem na saída padrão. Observação o número máximo de caracteres que cada linha contém é 80.