

Parâmetros da função main() e Arquivos

Os Argumentos argc e argv

- A declaração mais completa que se pode ter para a função **main()** é:

```
void main (int argc, char *argv[]);
```

- O **argc** (argument count): é um inteiro e possui o número de argumentos com os quais a função **main()** foi chamada na linha de comando.
- O **argv** (argument values): é um ponteiro para uma matriz de strings. Cada string desta matriz é um dos parâmetros da linha de comando.

1º.Exemplo-Argumentos

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

void main (int argc, char *argv[]) {
    if (argc!=2) {
        printf ("Você não digitou o seu nome\n");
        exit(1);
    }
    printf ("Boa Dia %s \n", argv[1]);
    getch();
}
```

C:\ Prompt de Comando - ola Juliana

```
C:\temp>ola Juliana  
Boa Dia Juliana
```

2º.Exemplo-Argumentos

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

void main (int argc, char *argv[])
{
    char *s;
    int t;
    s=malloc(900);

    if (s==NULL){
        printf ("Falha na solicitação de memória. \n");
        exit (1);
    }
    printf ("Entre com uma palavra:");
    gets (s);
    for (t=strlen(s)-1; t>=0; t--)
        putchar(s[t]);
    getch();
    free(s);
}
```

64. Prompt de Comando - teste2

```
C:\temp>teste2
```

```
Entre com uma palavra:Felicidade  
edadicileF_
```

Abrindo e Fechando um Arquivo

Principais utilizadas:

- ❑ `fopen`
- ❑ `exit`
- ❑ `fclose`

1º. Exemplo – Cria um arquivo para escrita.

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

main()
{
    FILE *p;
    char frase[50];
    int i;
    if ((p = fopen("teste.txt", "w")) == NULL)
    {
        printf("Erro! Impossivel abrir o arquivo!");
        exit(1);
    }
    printf ("Entre com uma frase:\n");
    gets(frase);
    for (i=0; frase[i]; i++)
        putc(frase[i],p);
    fclose(p);
}
```

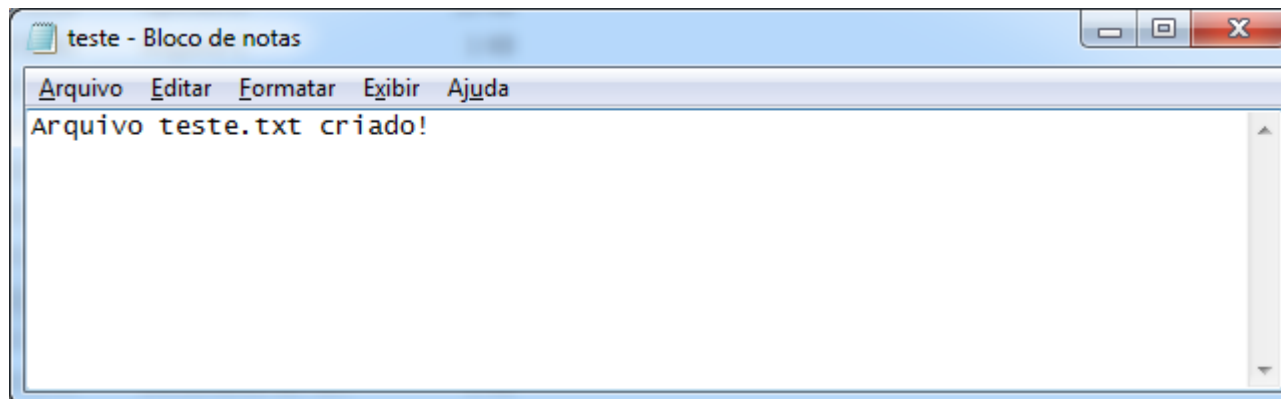

1º. Exemplo – Abre um arquivo para escrita.

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

main()
{
    FILE *p;
    char frase[50];
    int i;
    if ((p = fopen("testeprint.txt", "a")) == NULL)
    {
        printf ("Erro! Impossível abrir o arquivo!");
        exit(1);
    }
    printf ("Entre com uma frase:\n");
    gets(frase);
    // for (i=0;frase[i];i++)
    // putc (frase[i],p);
    fprintf (p,frase);
    fclose (p);
}
```

C:\JULIANA\2013\Fatec\disciplinas\LP\02-06-2013-Arquivo\ex

Entre com uma frase:
Arquivo teste.txt criado!



fopen

□ Protótipo:

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

□ Esta é a função de abertura de arquivos.

□ Parâmetros da função:

- O **nome_do_arquivo** determina qual arquivo deverá ser aberto.
- O modo de abertura diz à função `fopen()` que tipo de uso você irá fazer do arquivo.

fopen

Modo	Significado
"r"	Abre um arquivo para leitura
"w"	Cria um arquivo para escrita
"a"	Acrescenta dados no fim do arquivo ("append")
"rb"	Abre um arquivo binário para leitura
"wb"	Cria um arquivo binário para escrita
"ab"	Acrescenta dados binários no fim do arquivo
"r+"	Abre um arquivo para leitura e escrita
"w+"	Cria um arquivo para leitura e escrita
"a+"	Acrescenta dados ou cria um arquivo para leitura e escrita
"r+b"	Abre um arquivo binário para leitura e escrita
"w+b"	Cria um arquivo binário para leitura e escrita
"a+b"	Acrescenta dados ou cria um arquivo binário para leitura e escrita

fopen

- Como criar arquivo:
 - ▣ Para abrir um arquivo para escrita:

```
FILE *p;  
if ((p = fopen("teste.txt","w"))==NULL)  
{  
    printf("Erro! Impossível abrir o arquivo!");  
    exit(1);  
}
```

exit

- **Protótipo:**

```
void exit (int codigo_de_retorno);
```

- Esta função aborta a execução do programa.
- Pode ser chamada de qualquer ponto no programa
- Retorne, para o sistema operacional, o código_de_retorno.

exit

```
FILE *p;
```

```
if ((p = fopen("teste.txt","w"))==NULL)
{
    printf("Erro! Impossível abrir o arquivo!");
    exit(1);
}
```

fclose

- **Protótipo:**

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

- Esta função fecha um arquivo aberto.
- A função retorna zero no caso de sucesso.

Lendo e Escrevendo Caracteres em Arquivos

Funções:

- `putc`
- `getc`
- `fprintf`
- `fscanf`
- `feof`

putc

- **Protótipo:**

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

- **Escreve um caractere no arquivo, na posição corrente do ponteiro.**

getc

- **Protótipo:**

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

- Retorna um caractere lido do arquivo. Este caractere é o caractere sob o qual o ponteiro esta posicionado e depois move o ponteiro para próxima posição

fprintf

Exemplos:

- ❑ `fprintf (arq1, "Bom Dia!\n");`
- ❑ `fprintf (arq1, "Resultado: %d\n", resultado);`
- ❑ `fprintf (arq1, "soma=%10.6f quociente=%10.6f\n");`
- ❑ Escreve no arquivo informado -> variável `arq1`.

fscanf

Exemplos:

- ❑ `char c;`
- ❑ `fscanf(entrada, "%c", &c);`
- ❑ Usada para ler arquivos.

feof

- Protótipo:

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

- Verifica se um arquivo chegou ao fim:

- ▣ Retorna um valor diferente de zero se o arquivo chegou ao EOF(end-of-file)
- ▣ Caso contrário retorna zero.

2º. Exemplo – Lendo um arquivo

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

main()
{
    FILE *p;
    char c;

    if ((p = fopen("teste.txt", "r")) == NULL)
    {
        printf("Erro! Impossível abrir o arquivo!");
        exit(1);
    }

    while (!feof(p))
    {
        c = getc(p);
        printf("%c", c);
    }
    getch();
    fclose(p);
}
```

2º. Exemplo – Lendo um arquivo

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

main()
{
    FILE *p;
    char c;

    if ((p = fopen("teste.txt", "r")) == NULL)
    {
        printf("Erro! Impossivel abrir o arquivo!");
        exit(1);
    }

    while (!feof(p))
    {
        fscanf(p, "%c", &c);
        printf("%c", c);
    }
    getch();
    fclose(p);
}
```


Exercício

- Refaça os exemplos apresentados em aula.