



# Arquivos em C



# Arquivos

- ▶ Podemos dizer que um arquivo em C é um fluxo de bytes (stream). Todo lugar que tem capacidade para receber bytes da memória do computador ou transferi-los para ela recebe o nome de arquivo. Como exemplos, arquivo em disco, teclado, vídeo, impressora, portas de comunicação, etc.
- ▶ Para trabalhar com arquivo em C, usamos uma estrutura do tipo FILE. A definição desta estrutura está em stdio.h



# Arquivos



- ▶ Para se trabalhar com Arquivos devemos ter sempre em mente:
- ▶ Se o arquivo não existe, devo criar o arquivo.
  - ▶ Uma vez criado, este arquivo está aberto e pronto para ser preenchido por dados.
- ▶ Se o arquivo já existe,
  - ▶ Então devo abrir este arquivo para ler os dados e imprimir ou ler e adicionar novos dados.



# Arquivo tipos

- ▶ **Arquivo texto** - É um arquivo cujo conteúdo é baseado em uma sequência de caracteres que formam linhas determinadas por um caractere de nova linha ( “ \ n ”). Dentro destes arquivos podem ser gravados apenas dados em forma de texto. Ou seja, não poderemos gravar, por exemplo, um valor numérico nestes arquivos, a não ser que este valor seja transformado em sequências de caracteres
- ▶ **Arquivos binários** – Arquivos binários tem o seu conteúdo baseado em uma estrutura ou dado que respeita um determinado tipo de dado. Este tipo de dado pode ser um tipo simples (int, float, char) ou um tipo estruturado como registro (struct). Portanto esses arquivos são construídos como uma sequência de bytes respeitando uma determinada estrutura. Esses arquivos podem ser visualizados como um tabela onde cada linha possui um determinado conjuntos de campos (colunas). Cada linha dessa tabela possuirá também um número que identificará sua posição no arquivo. Por essas características, tais arquivos têm a vantagem de permitir o acesso aleatório a posições.



# Abrindo um arquivo com fopen()

- Esta função executa duas tarefas.
  - 1º cria e preenche uma estrutura FILE com as informações necessárias para o programa e para o sistema operacional, de maneira que possamos nos comunicar.
  - 2º retorna um ponteiro do tipo FILE que aponta para a localização na memória desta estrutura criada.

# Modos de abertura de arquivos

Lista para arquivos modo texto

Sigla	Significado
"r"	Abre um arquivo para leitura em modo texto. Se o arquivo não existir ou der falha na abertura é retornado NULL.
"w"	Cria um arquivo em modo texto para gravação. Se o arquivo já existir, os dados serão apagados e novos dados serão incluídos.
"a"	Abre um arquivo em modo texto para adição de novos dados, a partir do seu final. Se o arquivo não existir ou der falha na abertura é retornado NULL.
"r+"	Abre um arquivo em modo texto para atualização, ou seja, tanto para leitura como para gravação. Se o arquivo não existir ou der falha na abertura é retornado NULL.
"w+"	Cria um arquivo em modo texto para atualização, ou seja, tanto para leitura como para gravação. Se o arquivo já existir o conteúdo anterior será destruído.
"a+"	Abre um arquivo em modo texto para atualização, gravando dados novos a partir do seu final. Se o arquivo não existir ele será criado.

# fopen()

Ponteiro de arquivo  
(nome lógico do arquivo)

Nome físico  
do arquivo

Modo de  
abertura

```
arquivo = fopen ("nomefisico.txt", "r");
```

- A função **fopen()** abre um arquivo. Para tanto, devem ser passados o nome físico do arquivo e o modo de abertura. Caso o arquivo possa ser aberto, retorna um ponteiro referente; caso contrário, retorna NULL (nulo).

```
// Verificação de abertura de arquivo  
  
if ((arquivo = fopen("teste.txt", "r")) == NULL)  
{  
    puts ("Arquivo nao pode ser aberto...");  
    exit (1);  
}
```

# fclose()

Ponteiro de arquivo  
(nome lógico do arquivo)

**fclose** (arquivo);

- O comando **fclose()** fecha um arquivo em nível de sistema operacional. Para tanto, deve ser passado o nome lógico do arquivo a ser fechado.
- Terminar um programa, sem fechar um arquivo aberto, pode provocar perda de dados no arquivo ou corrompê-lo.
- Como, normalmente, há limite do sistema operacional para o número de arquivos abertos ao mesmo tempo, pode ser necessário fechar um arquivo antes de abrir outro.

# Escrevendo em um arquivo texto com `fputc()`

Caractere a ser  
escrito no arquivo

Ponteiro de arquivo  
(nome lógico do arquivo)

**putc** (caractere, arquivo);

- A função **putc()** ou **fputc()** escreve um caractere em um arquivo texto. Para tanto, devem ser passados o caractere a ser escrito e o nome lógico do arquivo aberto.
  - Se a escrita ocorrer devidamente, a função retorna o caractere escrito; caso contrário, retorna EOF.

# Exemplo do uso de fopen()

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

```
int main(){
    FILE *fptr;
    char ch;
```

Ponteiro para o arquivo

```
fptr = fopen("arquivo1.txt", "w");
```

Abrindo e **criando** o arquivo  
o "w" indica que é  
modo de gravação

```
while ((ch=getche()) != '\r')
```

```
    fputc(ch,fptr); // quando teclar enter ele sai e grava o arquivo
```

Inserindo dados no arquivo via fputc()

```
fclose(fptr);
return 0;
```

Fechando o arquivo com fclose()

```
}
```

# Gravando dados em um arquivo texto com putchar()

```
#include <stdio.h>
int main()
{
    FILE *fp;
    char string[100];
    int i;
    fp = fopen("arquivo.txt","w");
    if(!fp)
    {
        printf( "Erro na abertura do arquivo");
        exit(1);
    }
    printf("Entre com a string a ser gravada no arquivo:");
    gets(string);
    for(i=0; string[i]; i++)
        putchar(string[i], fp);
    fclose(fp);
    return 0;
}
```

# Abre e imprime o arquivo na tela

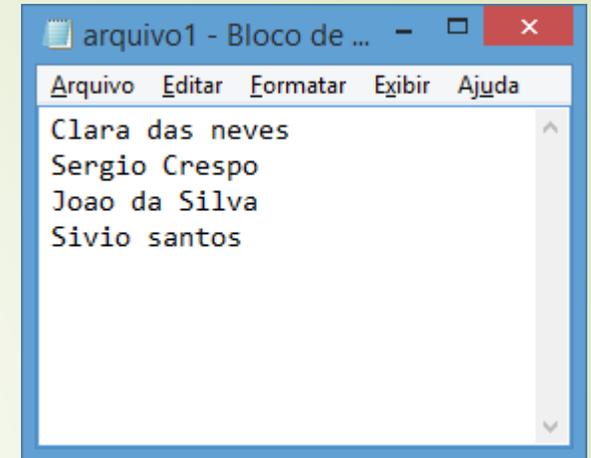
```
#include<stdio.h>
#include<stdlib.h>

int main (void ) {
    int c;
    FILE *pa;
    char *nome = "arquivo1.txt";

    /* Abre o arquivo para leitura */

    if (( pa = fopen(nome, "r")) == NULL) {
        printf("\n\nNao foi possivel abrir o arquivo.\n");
        exit(1);
    }

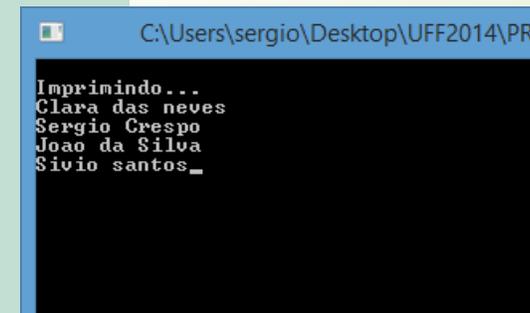
    printf("\nImprimindo...\n");
    c = fgetc(pa);
    while (!feof(pa)) {
        putchar(c);
        c = fgetc(pa);
    }
    fclose(pa);
    getchar(); /* Espera o usuario digitar alguma coisa */
    return 0;
}
```



```
Arquivo  Editar  Formatar  Exibir  Ajuda
Clara das neves
Sergio Crespo
Joao da Silva
Sivio santos
```

## **fgetc(FILE \*f)**

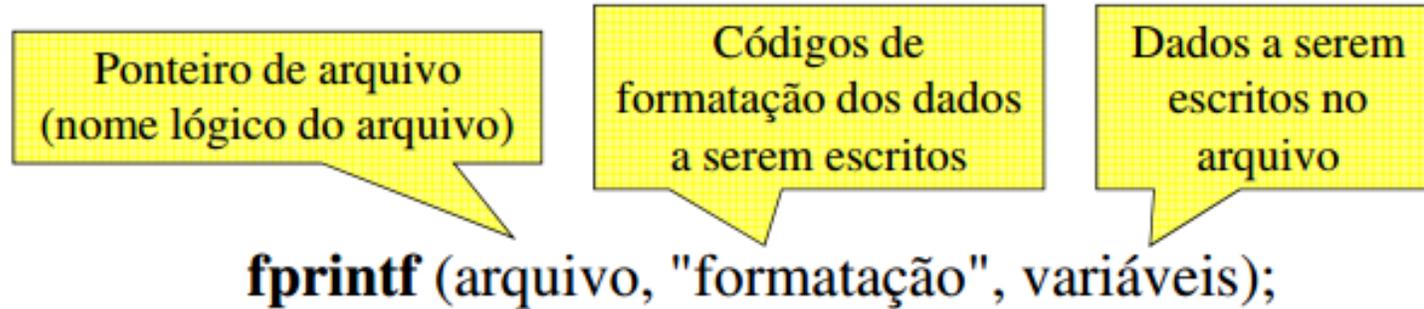
Argumentos: apontador para arquivo.  
Retorna: próximo caráter ou EOF se fim de arquivo.



```
C:\Users\sergio\Desktop\UFF2014\PR
Imprimindo...
Clara das neves
Sergio Crespo
Joao da Silva
Sivio santos_
```

EOF = Fim do arquivo

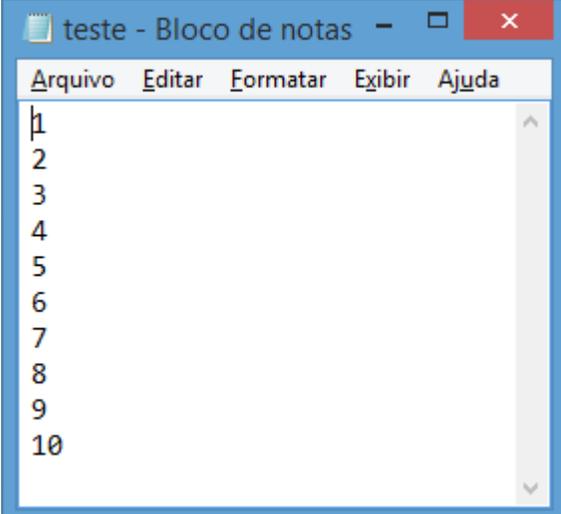
# Escrevendo dados em um arquivo texto com fprintf()



- A função **fprintf()** escreve dados formatados em um arquivo texto. Para tanto, devem ser passados o nome lógico do arquivo aberto, os códigos de formatação e as variáveis referentes aos dados a serem escritos no arquivo.
  - Se a escrita ocorrer devidamente, a função retorna a quantidade de *bytes* escritos com sucesso no arquivo; caso contrário, retorna 0.

Imprimir os números de 1 até 10 no arquivo teste.txt

```
#include <stdio.h>
int main()
{
    FILE *f = fopen("teste.txt", "w");
    int i;
    for (i=1; i<=10; i++)
        fprintf(f, "%d\n", i);
    fclose(f);
}
```



teste - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

```
1
2
3
4
5
6
7
8
9
10
```

# Leitura de dados de um arquivo texto usando fscanf()

Ponteiro de arquivo  
(nome lógico do arquivo)

Códigos de  
formatação dos  
dados a serem lidos

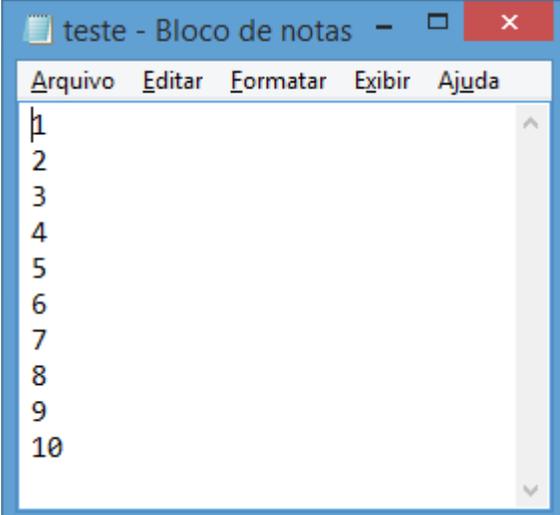
Variáveis que  
armazenarão os  
dados lidos

**fscanf** (arquivo, "formatação", variáveis);

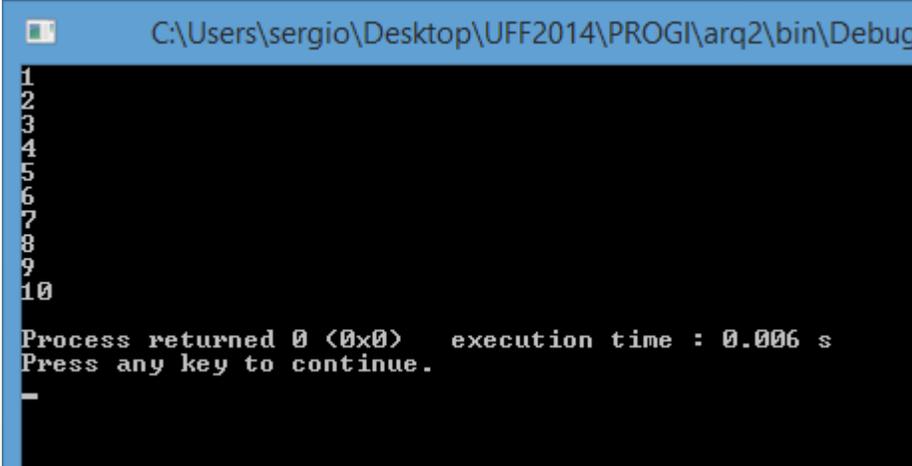
- A função **fscanf()** lê dados formatados de um arquivo texto. Para tanto, devem ser passados o nome lógico do arquivo aberto, os códigos de formatação referentes aos dados a serem lidos e as variáveis que receberão os dados lidos.
  - Se a leitura ocorrer devidamente, a função retorna a quantidade de dados lidos com sucesso; caso contrário, retorna 0.
  - Se a função tenta ler o fim de arquivo, retorna EOF.

# Imprimir os números que estão no arquivo teste.txt

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *f = fopen("teste.txt", "r");
    int i;
    while (fscanf(f, "%d", &i) == 1)
        printf("%d\n", i);
    fclose(f);
}
```



A screenshot of a Notepad window titled "teste - Bloco de notas". The window has a menu bar with "Arquivo", "Editar", "Formatar", "Exibir", and "Ajuda". The text area contains the numbers 1 through 10, each on a new line.



A screenshot of a Windows Command Prompt window. The title bar shows the path "C:\Users\sergio\Desktop\UFF2014\PROGI\arq2\bin\Debug". The window displays the output of the C program: the numbers 1 through 10, each on a new line. Below the numbers, it says "Process returned 0 (0x0) execution time : 0.006 s" and "Press any key to continue." followed by a cursor.

# Contando o número de linhas do arquivo teste.txt

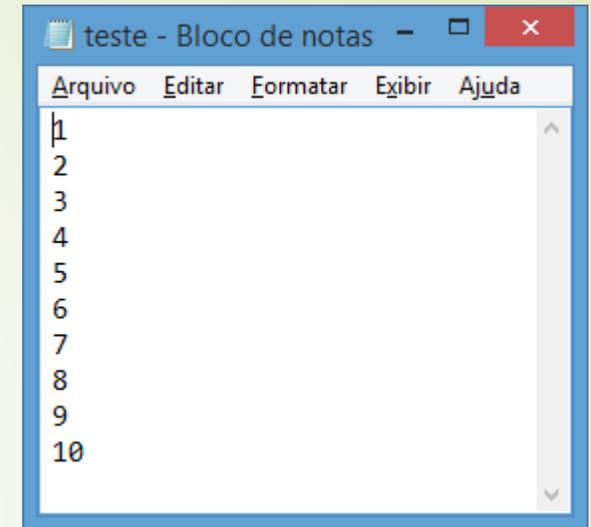
```
#include <stdio.h>

int main(void)
{
    char url[]="teste.txt",
        ch;
    int num=0;
    FILE *arq;

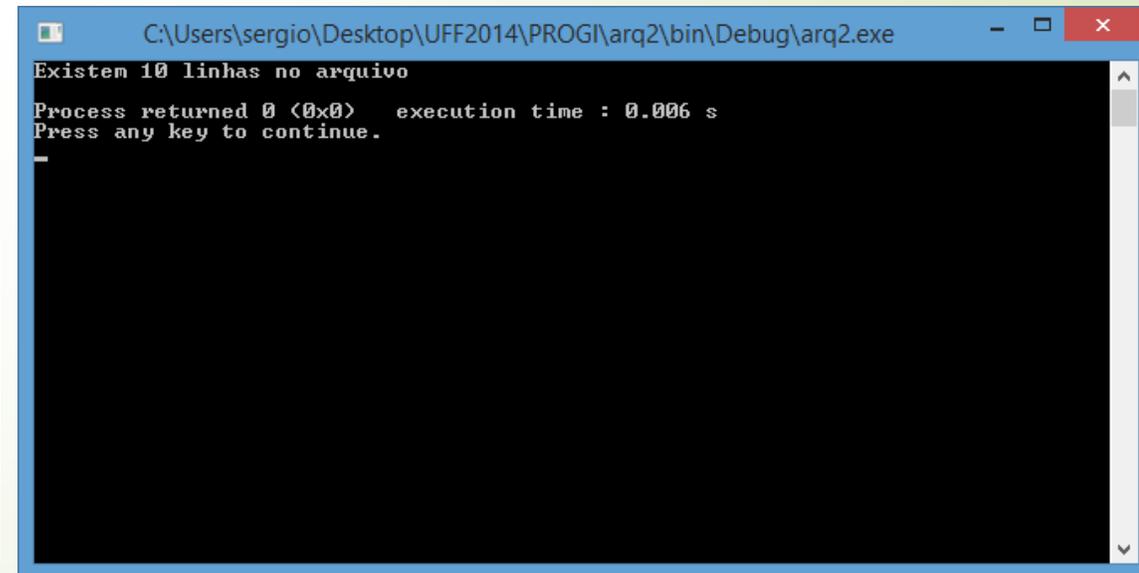
    arq = fopen(url, "r");

    if(arq == NULL)
        printf("Erro, não foi possível abrir o arquivo\n");
    else
        while( (ch=fgetc(arq))!= EOF )
            if(ch == '\n')
                num++;

    printf("Existem %d linhas no arquivo\n", num);
    fclose(arq);
    return 0;
}
```



```
1
2
3
4
5
6
7
8
9
10
```



```
C:\Users\sergio\Desktop\UFF2014\PROGI\arq2\bin\Debug\arq2.exe
Existem 10 linhas no arquivo
Process returned 0 (0x0)   execution time : 0.006 s
Press any key to continue.
```

# Lendo arquivo1.txt e calculando as medias

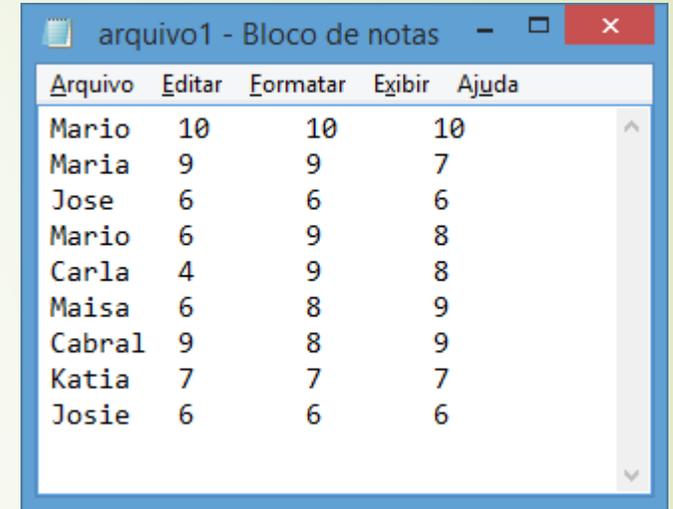
```
#include <stdio.h>

int main(void)
{
    char url[]="arquivo1.txt",
        nome[20];
    float nota1, nota2, nota3;
    FILE *arq;

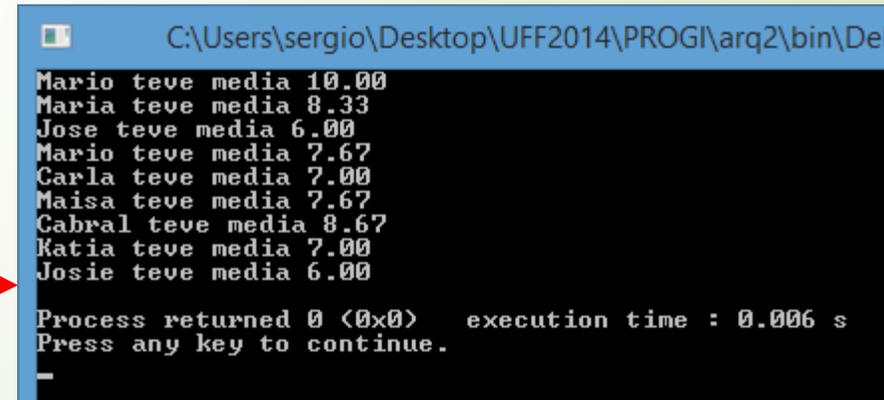
    arq = fopen(url, "r");
    if(arq == NULL)
        printf("Erro, nao foi possivel abrir o arquivo\n");
    else
        while( (fscanf(arq,"%s %f %f %f\n", nome, &nota1, &nota2, &nota3))!=EOF )
            printf("%s teve media %.2f\n", nome, (nota1+nota2+nota3)/3);

    fclose(arq);

    return 0;
}
```



Arquivo	Editar	Formatar	Exibir	Ajuda
Mario	10	10	10	
Maria	9	9	7	
Jose	6	6	6	
Mario	6	9	8	
Carla	4	9	8	
Maisa	6	8	9	
Cabral	9	8	9	
Katia	7	7	7	
Josie	6	6	6	



```
C:\Users\sergio\Desktop\UFF2014\PROGI\arq2\bin\De
Mario teve media 10.00
Maria teve media 8.33
Jose teve media 6.00
Mario teve media 7.67
Carla teve media 7.00
Maisa teve media 7.67
Cabral teve media 8.67
Katia teve media 7.00
Josie teve media 6.00

Process returned 0 (0x0)   execution time : 0.006 s
Press any key to continue.
_
```

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

```
int main(void)
```

```
{
```

```
    char url[]="notas.txt";
```

```
    float nota,
```

```
        media=0.0;
```

```
    FILE *arq;
```

```
    arq = fopen(url, "w");
```

```
    if(arq == NULL)
```

```
        printf("Erro, nao foi possivel abrir o arquivo\n");
```

```
    else{
```

```
        printf("Nota de Matematica: ");
```

```
        scanf("%f", &nota);
```

```
        fprintf(arq, "Matematica: %.2f\n", nota);
```

```
        media+=nota;
```

```
        printf("Nota de Fisica: ");
```

```
        scanf("%f", &nota);
```

```
        fprintf(arq, "Fisica: %.2f\n", nota);
```

```
        media+=nota;
```

```
        printf("Nota de Quimica: ");
```

```
        scanf("%f", &nota);
```

```
        fprintf(arq, "Quimica: %.2f\n", nota);
```

```
        media+=nota;
```

```
        media /= 3;
```

```
        fprintf(arq, "Media final: %.2f\n", media);
```

```
    }
```

```
    fclose(arq);
```

```
    return 0;
```

```
}
```

Escreva um programa em C que peça 3 notas de um aluno (Matemática, Física e Química), e salve esses dados em um arquivo chamado "notas.txt", que deve ter, ao final, a média das três disciplinas.

```
notas - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
Matematica: 10.00
Fisica: 10.00
Quimica: 6.00
Media final: 8.67
```