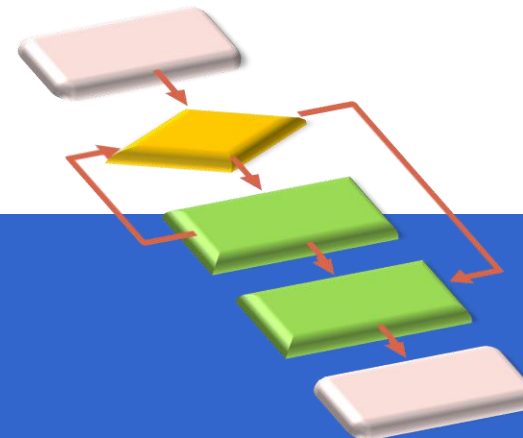




INSTITUTO DE COMPUTAÇÃO



Algoritmos II

ARQUIVOS

Linguagem C – Características Gerais e Arquivo Texto

Prof.^a Vanessa de Oliveira Campos

Arquivos – Linguagem C

- Em C, um arquivo é um **fluxo de dados**.
- Fluxos de dados são conjuntos sequencias de bytes (sem qualquer tipo de estrutura interna).
- As operações de entrada e saída de dados sobre um arquivo são apoiadas pelo armazenamento temporário de fluxos de dados em áreas da memória principal, denominadas *buffers*.



Declaração de Arquivos – Linguagem C

- Para que um programa possa manipular um arquivo, deve-se declarar:

```
FILE *nomeVariavel;
```

Exemplos:

```
FILE *arq1 , *arqSaida;
```

```
FILE *arqEntrada;
```



Arquivos – Linguagem C

- Algumas funções disponíveis:
 - Abrir Arquivo (**fopen**)
 - Fechar Arquivo (**fclose**)
 - Ler Arquivo (**fread** ou **fscanf**)
 - Escrever Arquivo (**fwrite** ou **fprintf**)
 - Detectar o Fim do arquivo (**feof**)
 - Posicionar o Curso no Arquivo (**fseek**)



Abertura de Arquivos – Linguagem C

Abrir Arquivo (**fopen**)

- Essa função associa uma **variável** declarada como um tipo de arquivo, o qual representa um nome lógico do arquivo, a um arquivo físico (externo) reconhecido pelo sistema de arquivos.

`FILE *fopen (char *<nome de arquivo >, char *<modo de abertura >)`



Abertura de Arquivos – Linguagem C

- modo de abertura:
 - o tipo do arquivo, se binário ou texto;
 - as operações que podem ser realizadas sobre o arquivo;
 - se for encontrado um arquivo com mesmo nome, se esse deve ser eliminado e um novo arquivo deve ser criado em seu lugar
 - a partir de qual os dados devem ser gravados no arquivo (começo ou fim)



Abertura de Arquivos – Linguagem C

Exemplos:

```
arq1 = fopen ( "dados . dat " , "ab" ) ;  
arqEntrada = fopen ( " info.txt " , " r" ) ;
```



Abertura de Arquivos – Linguagem C

Quadro resumo de modos de abertura para arquivos binários

Modo de Abertura	Arquivo existe	Arquivo não existe	Permissão de leitura	Permissão de escrita
rb	ok	erro	X	
wb	cria novo, antigo eliminado	cria		X
r+b ou rb+	ok	erro	X	X
w+b ou wb+	cria novo, antigo eliminado	cria	X	X
ab	novos dados acrescentados ao final	cria (como wb)		X
a+b ou ab+	novos dados acrescentados ao final	cria (como wb+)	X	X

Os modos de abertura para arquivos de texto são r, w, r+, w+, a, a+.



Abertura de Arquivos – Linguagem C

Exemplos:

```
arq1 = fopen ( "dados . dat " , "ab" ) ;
```

```
arqEntrada = fopen ( " info.txt " , " r" ) ;
```

```
arqcadastro = fopen("c:\\documentos\\dadoscadastro.dat,"rb");
```

```
arqprodutos = fopen("c:\\loja\\arqs\\produtos.dat,"rb");
```



Fechamento de Arquivos – Linguagem C

Fechar Arquivo (**fclose**)

■ Protótipo:

```
int fclose ( variavelArquivo ) ;
```

■ Se a função for executada corretamente, será devolvido zero.

Exemplo:

```
if ( fclose ( arq ) == 0 )  
    printf ( "Arquivo fechado corretamente" ) ;
```



Final de Arquivos – Linguagem C

Fim do arquivo (**feof**)

- Protótipo:

```
feof ( FILE *arq );
```

- A função feof devolve o valor verdadeiro (não zero) caso o final do arquivo tenha sido atingido, ou falso (zero) em caso contrário.



Leitura e Escrita em Arquivos – Linguagem C

- Ler Arquivo (**fscanf**)

`fscanf (<FILE>, <"%codigoformato">, <var1, var2 , . . . >)`

- Escrever em Arquivo (**fprintf**)

`fprintf (<FILE *>, <"%codigoformato">, <var1, var2, . . . >)`



Leitura e Escrita em Arquivos – Linguagem C

Exemplo:

```
char aux;  
FILE *arqt = fopen ( " teste.txt " , " r " ) ;  
while ( fscanf (arqt , "%c " , &aux ) != EOF)  
    printf ( "%c " , aux ) ;  
fclose (arqt ) ;
```



Leitura e Escrita em Arquivos – Linguagem C

Exemplo:

```
void grava_arq (FILE * arquivo, int matriz[] [], int mlin, int mcol)
{
    int i, j;
    for ( i = 0; i < mlin; i++ )
    {
        for ( j = 0; j < mcol; j++ )
            fprintf(arquivo, "%d ", matriz[i][j]); //gravação de dados em arquivo
        fprintf(arquivo, "\n"); //nova linha
    }
}
```



Leitura e Escrita em Arquivos – Linguagem C

Exemplo:

```
void leia_arq (FILE * arquivo, int matriz[] [], int mlin, int mcol)
{
    int i, j;
    // Leitura da matriz a partir do arquivo
    for ( i = 0; i < mlin; i++ )
        for ( j = 0; j < mcol; j++ )
            fscanf(arquivo, "%d", &matriz[i][j]); // leitura dos valores
                                                    // do arquivo para a matriz
}
```



Lendo dados de um arquivo texto

- Ao realizar a leitura de um caractere, automaticamente o indicador de posição do arquivo se move para o próximo caractere.
- Ao chegar no fim do arquivo, a função **fscanf** retorna o valor especial **EOF**.
- Para voltar ao início do arquivo:
 - fechar e abrir o arquivo novamente, ou
 - usar o comando **rewind**.



Lendo dados de um arquivo texto

Exemplo:

```
while ( fscanf(arquivo, "%c", &aux) != EOF )  
    printf( "%c", aux);  
printf( "\n\n Imprimindo novamente \n\n");  
rewind(arquivo);  
while ( fscanf(arquivo, "%c", &aux) != EOF )  
    printf( "%c", aux);
```





Exercício de Fixação

- 1) Faça um programa que conte o número de caracteres de um arquivo texto e imprima o resultado na tela.



Exercício de Fixação

2) Considere um arquivo texto que armazene caracteres variados (letras, pontuações, números, etc), ou seja, um arquivo com um texto digitado. Escreva um programa que leia esse arquivo e gere um novo arquivo chamado "resumo.txt" que contenha somente as letras (letras de A-Z ou a-z) do arquivo original. Na gravação do arquivo deve-se manter a ordem em que as letras aparecem no arquivo original.



Exercício de Fixação

3) Faça um programa que leia as informações de nome, idade e salário de 4 pessoas e salve em um arquivo chamado "dados.txt".



Exercício de Fixação

4) Faça um programa que leia seu arquivo "dados.txt" e coloque as informações em um tipo de registro chamado TPessoas. Declare o tipo e o vetor de registro para as 4 pessoas e salve em um arquivo chamado "dados.txt". Mostre o salário médio das pessoas depois de ler.

