

LABORATÓRIO DE ALGORITMOS E ESTRUTURAS DE DADOS I

Prof. Caio César de Freitas Dantas

Manipulação de Arquivos em C

Arquivos

- O armazenamento de dados em variáveis e matrizes é temporário. Esses dados são perdidos quando um programa é encerrado.
- Os arquivos são estruturas de dados que permitem que os dados sejam armazenados, persistidos, de forma permanente.
- Quando um arquivo é aberto, um objeto é criado e um stream, um fluxo de dados, é associado ao objeto.

Manipulação de Arquivos em C

- Um STREAM é associado a um arquivo por uma operação de abertura do arquivo e, a partir da associação, todas as demais operações de escrita e leitura podem ser realizadas.

Os streams fornecem canais de comunicação entre arquivos e programas. Existem dois tipos de streams:

- Stream de Texto – formado por sequência de caracteres; o padrão C ANSI permite que um stream de texto seja organizado em linhas terminadas por um caractere de nova linha.
- Stream Binário – formado por sequência de bytes, a quantidade de byte é a mesma do dispositivo, ou seja, não há tradução.

Manipulação de Arquivos em C

- O sistema de entrada e saída do ANSI C, sendo composto por uma série de funções, cujos protótipos estão reunidos em `stdio.h`
- Todas as funções relacionadas a arquivos trabalham com o conceito de ponteiro de arquivo.
- Esta definição também está no arquivo `stdio.h`, e um ponteiro de arquivo pode ser declarado da seguinte maneira:

```
FILE *Arquivo;
```

Manipulação de Arquivos em C

- Pela declaração do ponteiro anterior, passa a existir uma variável de nome Arquivo, que é ponteiro para um arquivo a ser manipulado.
- O ponteiro de arquivo une o sistema de E/S a um buffer e não aponta diretamente para o arquivo em disco, contendo informações sobre o arquivo, incluindo nome, status (aberto, fechado e outros) e posição atual sobre o arquivo

Manipulação de Arquivos em C

| Função | Ação |
|--|--|
| <code>fopen()</code> | Abre um arquivo |
| <code>Fclose()</code> | Fecha um arquivo |
| <code>putc()</code> e <code>fputc()</code> | Escreve um caractere em um arquivo |
| <code>getc()</code> e <code>fgetc()</code> | Lê um caractere de um arquivo |
| <code>fseek()</code> | Posiciona em um registro de um arquivo |
| <code>fprintf()</code> | Efetua impressão formatada em um arquivo |
| <code>fscanf()</code> | Efetua leitura formatada em um arquivo |
| <code>feof()</code> | Verifica o final de um arquivo |
| <code>fwrite()</code> | Escreve tipos maiores que 1 byte em um arquivo |
| <code>fread()</code> | Lê tipos maiores que 1 byte de um arquivo |

Manipulação de Arquivos em C

Abrindo um Arquivo

- A função que abre um arquivo em C é a função `fopen()`, que devolve o valor `NULL` (nulo) ou um ponteiro associado ao arquivo.
- Deve ser passado para função o nome físico do arquivo e o modo como este arquivo deve ser aberto.

```
Arquivo = fopen ("texto.txt","w");
```

Manipulação de Arquivos em C

Abrindo um Arquivo

- De acordo com a instrução anterior, está sendo aberto um arquivo de nome “texto.txt”, habilitado apenas para escrita (w-write)
- Por exemplo, pode-se codificar a instrução de abertura de arquivo da seguinte maneira:

```
if ((Arquivo = fopen("texto.txt","w")) == NULL) {  
    printf("\n Arquivo TEXTO.TXT não pode ser aberto : TECLE ALGO");  
    getch();  
}
```


Manipulação de Arquivos em C

| Modo | Significado |
|------|---|
| “r” | Abre um arquivo texto para leitura. O arquivo deve existir antes de ser aberto. |
| “w” | Abrir um arquivo texto para gravação. Se o arquivo não existir, ele será criado. Se já existir, o conteúdo anterior será destruído |
| “a” | Abrir um arquivo texto para gravação. Os dados serão adicionados no fim do arquivo ("append"), se ele já existir, ou um novo arquivo será criado, no caso de arquivo não existente anteriormente. |
| “rb” | Abre um arquivo binário para leitura. Igual ao modo "r" anterior, só que o arquivo é binário |
| “wb” | Cria um arquivo binário para escrita, como no modo "w" anterior, só que o arquivo é binário. |
| “ab” | Acrescenta dados binários no fim do arquivo, como no modo "a" anterior, só que o arquivo é binário. |
| “r+” | Abre um arquivo texto para leitura e gravação. O arquivo deve existir e pode ser modificado. |
| “w+” | Cria um arquivo texto para leitura e gravação. Se o arquivo existir, o conteúdo anterior será destruído. Se não existir, será criado. |
| “a+” | Abre um arquivo texto para gravação e leitura. Os dados serão adicionados no fim do arquivo se ele já existir, ou um novo arquivo será criado, no caso de arquivo não existente anteriormente. |

Manipulação de Arquivos em C

Fechando um Arquivo

- Para o esvaziamento da memória de um arquivo é utilizada a função `fclose()`, que associa-se diretamente ao nome lógico do arquivo (STREAM):

```
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.

Manipulação de Arquivos em C

Leitura em Arquivo Texto

Caractere lido
do arquivo

Ponteiro de arquivo
(nome lógico do arquivo)

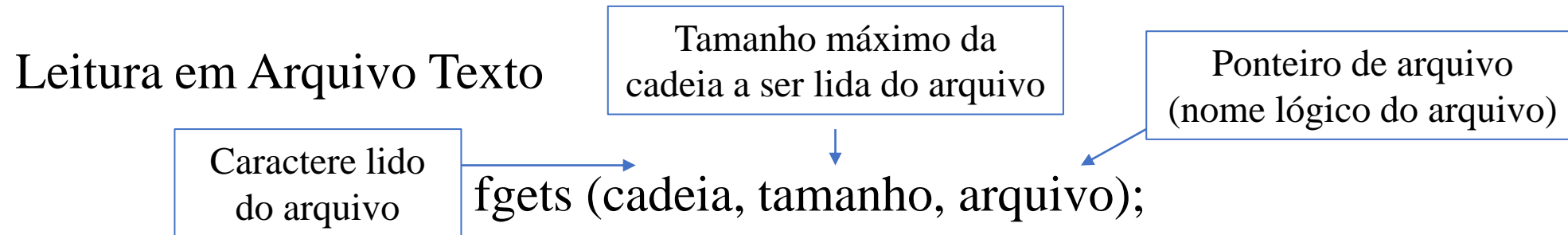
`caractere = getc (arquivo);`

A função `getc()` ou `fgetc()` lê um caractere de um arquivo texto, retornando-o. Para tanto, deve ser passado o nome lógico do arquivo aberto.

Se o ponteiro do arquivo estiver no final do mesmo ou ocorrer um erro na leitura, a função retorna EOF.

Existem duas funções para preservar a compatibilidade com versões mais antigas de C/C++.

Manipulação de Arquivos em C

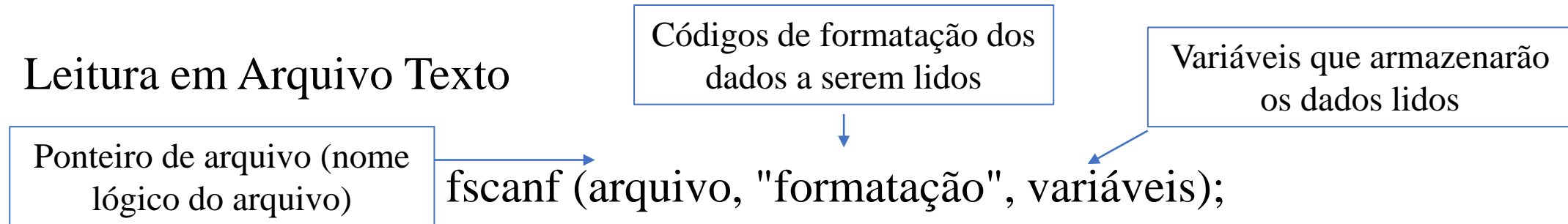


A função `fgetc()` lê uma cadeia de um arquivo texto. Para tanto, devem ser passados a cadeia a ser lida (dado de saída), o tamanho máximo da cadeia e o nome lógico do arquivo aberto.

- A função lê a cadeia até que um caractere de nova linha seja alcançado ou (tamanho - 1) caracteres tenham sido lidos.
- A função inclui os caracteres `"\n"` e `NULL` ao final da cadeia.
- Se a leitura ocorrer devidamente, a função retorna um ponteiro para a cadeia lida; caso contrário, retorna nulo.

Manipulação de Arquivos em C

Leitura em Arquivo Texto

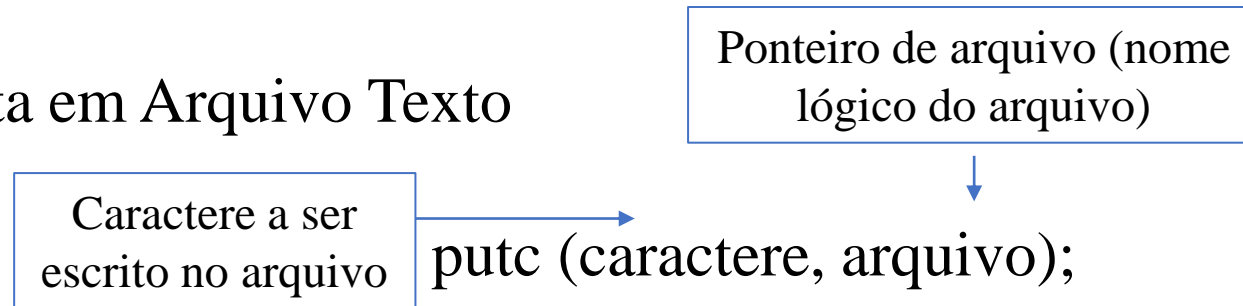


A função `fscanf()` lê dados formatados de um arquivo texto. Para tanto, devem ser passados o nome lógico do arquivo aberto, os códigos de formatação referentes aos dados a serem lidos e as variáveis que receberão os dados lidos.

- 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

Manipulação de Arquivos em C

Escrita em Arquivo Texto

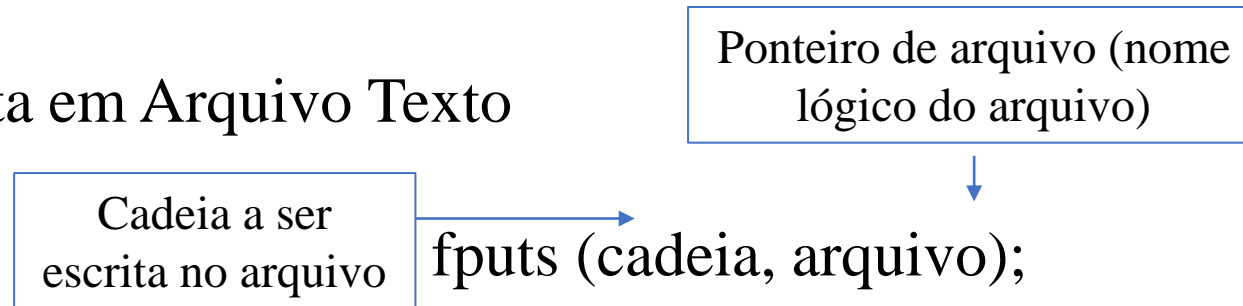


A função `putc()` ou `fputc()` escreve um caractere em um arquivo texto. Para tanto, devem ser passados o caractere a ser escrito e o nome lógico do arquivo aberto.

- Se a escrita ocorrer devidamente, a função retorna o caractere escrito; caso contrário, retorna EOF.
- Existem duas funções para preservar a compatibilidade com versões mais antigas de C/C++.

Manipulação de Arquivos em C

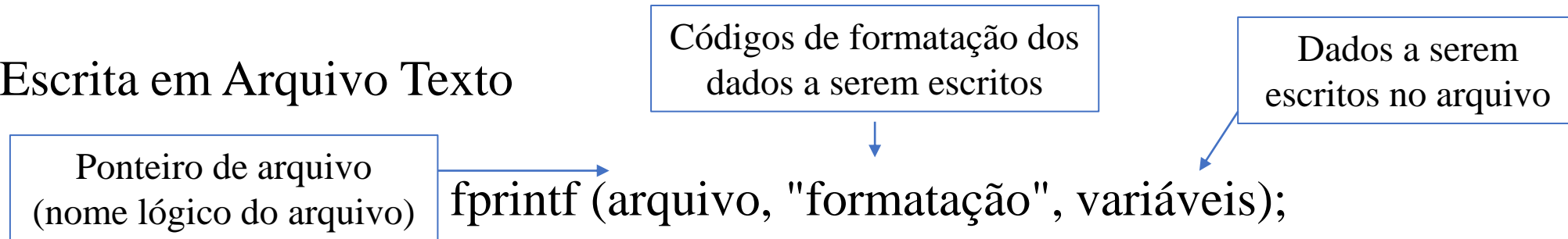
Escrita em Arquivo Texto



- A função `fputs()` escreve uma cadeia em um arquivo texto. Para tanto, devem ser passados a cadeia a ser escrita e o nome lógico do arquivo aberto.
 - Se a escrita ocorrer devidamente, a função retorna um valor não negativo; caso contrário, retorna EOF.

Manipulação de Arquivos em C


Escrita em Arquivo Texto



- 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.

Manipulação de Arquivos em C

Final de Arquivo


Ponteiro de arquivo
(nome lógico do arquivo)  feof (arquivo);

A função lógica feof() serve para indicar que o final de um arquivo binário (preferencialmente) ou texto foi encontrado. Para tanto, deve ser passado o nome lógico do arquivo aberto.

- Se o fim de arquivo não tiver sido atingido, a função retorna 0; caso contrário, retorna um valor diferente de 0.
- É geralmente usada em leitura de arquivos binários, pois um valor inteiro pode ser lido, erroneamente, como sendo o EOF e não como parte do arquivo, finalizando a leitura

Manipulação de Arquivos em C

Verificação de Erro


Ponteiro de arquivo
(nome lógico do arquivo)  ferror (arquivo);

A função `ferror()` retorna verdadeiro se ocorreu um erro durante a última operação com o arquivo; caso contrário, retorna falso. Para tanto, deve ser passado o nome lógico do arquivo aberto.

- Como cada "operação de arquivo" modifica a condição de erro, a função deve ser chamada logo após a operação a ser avaliada

Manipulação de Arquivos em C

Limpeza de Buffer

Ponteiro de arquivo
(nome lógico do arquivo)  fflush (arquivo);

A função fflush() escreve o conteúdo do buffer em um arquivo, esvaziando-o. Para tanto, deve ser passado o nome lógico do arquivo aberto para escrita.

- Se a limpeza do buffer ocorrer devidamente, a função retorna o valor 0; caso contrário, retorna EOF.

Manipulação de Arquivos em C

Remoção de Arquivo

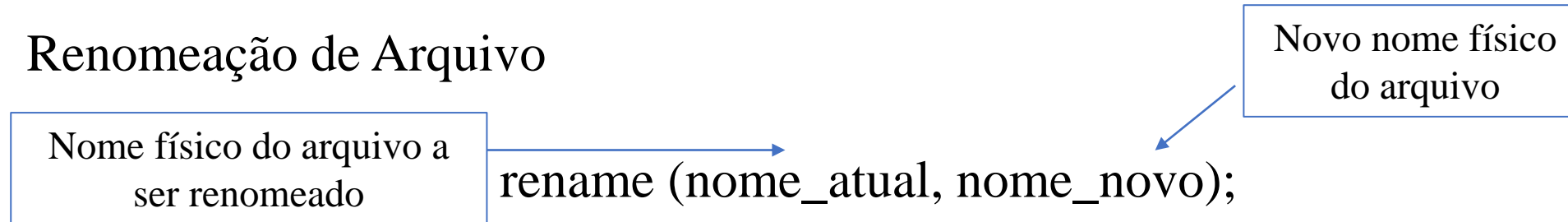
Nome físico do arquivo → `remove (nome_arquivo);`

A função `remove()` remove um arquivo. Para tanto, deve ser passado o nome físico do arquivo fechado.

- Se a remoção do arquivo ocorrer devidamente, a função retorna o valor 0; caso contrário, retorna um valor diferente de zero.

Manipulação de Arquivos em C

Renomeação de Arquivo



A função `rename()` renomeia um arquivo. Para tanto, devem ser passados o nome físico atual do arquivo fechado e o novo nome físico do mesmo.

- Se a renomeação do arquivo ocorrer devidamente, a função retorna o valor 0; caso contrário, retorna um valor diferente de zero.
- A função também serve para mover um arquivo.
 - É preciso incluir a nova localização como parte do novo nome do arquivo.

FIM!

