

Notas Práticas e Exercícios 1

Disciplina: Laboratório de Estrutura de Dados II

1 Comandos em C para Manipulação de Arquivos e Strings

Leitura e Escrita

| Comando | Descrição | Exemplo |
|-----------|-------------------------------|-----------------------------------|
| fopen() | Abre um arquivo. | FILE* f = fopen("data.txt", "r"); |
| fclose() | Fecha um arquivo aberto. | fclose(f); |
| fputc() | Escreve um caractere. | fputc('A', f); |
| fgetc() | Lê um caractere. | char c = fgetc(f); |
| fprintf() | Escreve uma string formatada. | fprintf(f, "Age: %d", 25); |
| fscanf() | Lê entrada formatada. | fscanf(f, "Age: %d", &age); |
| fgets() | Lê uma linha. | fgets(line, 100, f); |
| fputs() | Escreve uma linha. | fputs("Hello, World!", f); |

Outras Operações

| Comando | Descrição | Exemplo |
|-----------------------|---|---|
| <code>fseek()</code> | Move o ponteiro de arquivo. <code>SEEK_SET</code> a partir do início do arquivo, <code>SEEK_CUR</code> da posição atual e <code>SEEK_END</code> do fim do arquivo | <code>fseek(f, 5, SEEK_SET);</code> |
| <code>ftell()</code> | Retorna a posição do ponteiro. | <code>long pos = ftell(f);</code> |
| <code>rewind()</code> | Move o ponteiro ao início. | <code>rewind(f);</code> |
| <code>feof()</code> | Testa fim de arquivo. | <code>if (feof(f)) { /*...*/ }</code> |
| <code>ferror()</code> | Testa indicador de erro. | <code>if (ferror(f)) { /*...*/ }</code> |

Manipulação de Strings

| Função | Descrição | Exemplo |
|------------------------|--|--|
| <code>strcpy()</code> | Copia uma string para outra. | <code>strcpy(dest, "hello");</code> |
| <code>strcat()</code> | Concatena duas strings. | <code>strcat(str1, str2);</code> |
| <code>strlen()</code> | Retorna o comprimento. | <code>int len = strlen("hello");</code> |
| <code>strcmp()</code> | Compara duas strings. | <code>if(strcmp(str1, str2) == 0) { /*...*/ }</code> |
| <code>strchr()</code> | Localiza o primeiro ocorrência de um caractere. | <code>char *p = strchr(str, 'e');</code> |
| <code>strstr()</code> | Localiza uma substring. | <code>char *p = strstr(str, "hel");</code> |
| <code>strdup()</code> | Duplica uma string. | <code>char *copy = strdup(original);</code> |
| <code>strtok()</code> | Divide a string em tokens. | <code>char *token = strtok(str, " ");</code> |
| <code>sprintf()</code> | Escreve formatado em uma string. | <code>sprintf(buffer, "Age: %d", age);</code> |
| <code>sscanf()</code> | Lê a entrada formatada de uma string. | <code>sscanf(str, "Age: %d", &age);</code> |
| <code>strcspn()</code> | Retorna o índice do primeiro caractere na string 1 que corresponda a qualquer caractere na string 2. | <code>size_t index = strcspn(nome, "\n");</code> |

*mais informações podem ser encontradas em: <https://www.geeksforgeeks.org/c-programming-language/>

2 Exercícios

Lembre-se de que, para alguns programas de leitura funcionem corretamente, um arquivo “frase.txt”, “file.txt” ou “nomes.txt” deve existir no diretório onde o programa é executado ou você deve especificar o caminho completo para o arquivo.

2.1 Escrita em Arquivo

Escreva um programa em C que peça ao usuário para inserir uma frase e então salve essa frase em um arquivo chamado “frase.txt”.

```
#include <stdio.h>

int main() {
    FILE *f;
    char frase[100];

    printf("Digite uma frase: ");
    scanf("%99s", frase);

    f = fopen("frase.txt", "w");
    fprintf(f, "%s", frase);
    fclose(f);

    return 0;
}
```

2.2 Leitura de Arquivo

Escreva um programa em C que leia o conteúdo do arquivo “frase.txt” e o exiba na tela.

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

```

int main() {
    FILE *f;
    char frase[100];

    f = fopen("frase.txt", "r");
    fgets(frase, 100, f);
    printf("Frase lida do arquivo: %s", frase);
    fclose(f);

    return 0;
}

```

2.3 Busca em Arquivo

Escreva um programa em C que leia um arquivo contendo uma lista de nomes (um nome por linha) e pergunte ao usuário um nome para buscar. Se o nome estiver no arquivo, exiba “Nome encontrado!”.

```

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

int main() {
    FILE *f;
    char nome[50], nomeBusca[50];
    int encontrado = 0;

    printf("Digite o nome para busca: ");
    scanf("%49s", nomeBusca);

    f = fopen("nomes.txt", "r");
    while(fgets(nome, 50, f) != NULL) {
        // Remover o '\n' do final da string lida
        nome[strlen(nome)] = 0;
        if(strcmp(nome, nomeBusca) == 0) {
            encontrado = 1;
            break;
        }
    }
    fclose(f);
}

```

```
    if(encontrado) {
        printf("Nome encontrado!\n");
    } else {
        printf("Nome nao encontrado.\n");
    }

    return 0;
}
```

2.4 Adição ao Final de um Arquivo

Escreva um programa em C que peça ao usuário um nome e adicione esse nome ao final do arquivo “nomes.txt”.

```
#include <stdio.h>

int main() {
    FILE *f;
    char nome[50];

    printf("Digite um nome: ");
    gets(nome);

    f = fopen("nomes.txt", "a");
    fprintf(f, "\n%s", nome);
    fclose(f);

    return 0;
}
```

2.5 Contagem de Linhas

Escreva um programa em C que conte e exiba o número de linhas presentes no arquivo “nomes.txt”.

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

```

int main() {
    FILE *f;
    char ch;
    int count = 0;

    f = fopen("nomes.txt", "r");
    while((ch = fgetc(f)) != EOF) {
        if(ch == '\n') {
            count++;
        }
    }
    fclose(f);

    printf("O arquivo tem %d linhas.\n", count);

    return 0;
}

```

2.6 Ler o 10º caractere de um arquivo

Escreva um programa que leia o 10º caractere de um arquivo e o exiba na tela.

```

#include <stdio.h>

int main() {
    FILE *f;
    char ch;

    f = fopen("file.txt", "r");
    fseek(f, 9, SEEK_SET);
    ch = fgetc(f);
    printf("O Decimo caractere eh: %c\n", ch);
    fclose(f);

    return 0;
}

```

2.7 Mover-se para a 5ª linha de um arquivo

Escreva um programa que se mova para a 5ª linha de um arquivo e leia e exiba seu conteúdo.

```
#include <stdio.h>

int main() {
    FILE *f;
    char line[100];
    int count = 0;

    f = fopen("file.txt", "r");
    while (count < 4 && fgets(line, sizeof(line), f)) {
        count++;
    }
    if (count == 4) {
        fgets(line, sizeof(line), f);
        printf("Quinta linha: %s", line);
    }
    fclose(f);

    return 0;
}
```

2.8 Atualizar um caractere específico

Escreva um programa que substitua o 15º caractere de um arquivo por 'X'. Depois altere para substituir mais de um caractere, ou seja por uma string "XYX"

```
#include <stdio.h>

int main() {
    FILE *f;

    f = fopen("file.txt", "r+");
    fseek(f, 14, SEEK_SET);
    fputc('X', f);
    fclose(f);
}
```



```
    return 0;  
}
```
