

Linguagens de Programação 1

Francisco Sant'Anna

Sala 6020-B

`francisco@ime.uerj.br`

`http://github.com/fsantanna-uerj/LP1`

Arquivos

Escrevendo Caracteres

```
#include <stdio.h>

int main (void) {
    FILE* f = fopen("/tmp/arq-01.txt", "w");
    fputc('a', f);
    fputc('b', f);
    fputc('c', f);
    fclose(f);
    return 0;
}
```

Lendo Caracteres

```
#include <stdio.h>

int main (void) {
    FILE* f = fopen("/tmp/arq-01.txt", "r");
    int v1 = fgetc(f);
    int v2 = fgetc(f);
    int v3 = fgetc(f);
    int v4 = fgetc(f);
    printf("%d %d %d %d\n", v1, v2, v3, v4);
    fclose(f);
    return 0;
}
```

Resumo

- Incluir biblioteca: `stdio.h`
- Abrir arquivo: `fopen`
- Fechar arquivo: `fclose`
- Escrever caractere: `fputc`
- Ler caractere: `fgetc`
- Fim de arquivo (-1): EOF

Exercício 8.1

- Leia uma string com `scanf` (máx 25 caracteres)
- Escreva a string para um arquivo, caractere a caractere, usando um `for`
 - A função `strlen` retorna o tamanho de uma string
 - É necessário incluir a biblioteca `string.h`

Exercício 8.2

- Leia o arquivo do exercício 8.1, caractere a caractere, para uma string (máx 25 caracteres)
- Exiba a string lida

Exercício 8.3

- Leia 10 strings com `scanf` (máx 25 caracteres) para um vetor de strings
- Escreva cada string para um arquivo, caractere a caractere, usando um `for` dentro de outro `for`

Exercício 8.4

- Altere o exercício 8.1 e 8.3 para usar a função a seguir a ser definida:
- `void escreva_string (FILE* f, char* str);`
- A função recebe um arquivo e uma string e escreve a string no arquivo, caractere a caractere