

## Exercícios – Sistemas de Arquivos

1. Implemente um programa C que faça a cópia de um arquivo. O programa recebe dois parâmetros na linha de comando, o nome do arquivo existente e o nome do arquivo de destino. Caso o arquivo de destino já exista, o programa deve exibir uma mensagem de erro e cancelar a cópia.

O programa não deve usar as funções da biblioteca padrão C (`fopen()`, `fread()`, `fwrite()` e assemelhadas), mas as chamadas POSIX correspondentes (`open()`, `read()`, `write()`, `close()`, `stat()`).

DICA 1: o modo do arquivo de destino deve ser o mesmo do arquivo de origem.

DICA 2: para definir o tamanho do *buffer* usado por `read()` e `write()`, execute o comando `stat arquivo` (onde `arquivo` é um arquivo existente qualquer) e veja o valor do parâmetro `IO Block`.

2. Implemente um programa C que liste os arquivos em um diretório. O nome do diretório é passado como parâmetro na linha de comando; caso seja omitido, liste os arquivos no diretório corrente. Cada linha deve seguir o formato

tipo nome do arquivo

onde `tipo` indica o tipo do arquivo, de acordo com a tabela abaixo:

tipo	tipo de arquivo
A	arquivo regular
D	diretório
O	outro

O programa deve usar as funções POSIX de manipulação de diretórios (`opendir()`, `readdir()`, `closedir()` e assemelhadas).

DICA: você pode considerar que uma `struct dirent` sempre possui o campo `d_type` definido.

DICA GERAL: O comando `man função` mostra maiores informações sobre cada função. Dentro de cada página de manual, a seção “SEE ALSO” sugere outras funções relacionadas.

O tratamento de erros de E/S não deve ser negligenciado na escrita dos programas.