

1ª Lista de Exercícios

Supondo um disco com as características abaixo, responda as questões de 1 a 7.

- 10 superfícies
- 4.096 trilhas/superfície
- 110 setores/trilha
- 512 bytes/setor
- Velocidade de rotação = 5.400 RPM
- Tempo médio de seek = 10 ms
- Latência média = 5,6 ms

1. Quantos cilindros o disco possui?
2. Quantos cilindros serão necessários para armazenar um arquivo 80.000 registros de 128 bytes cada?
3. Sabendo que o tempo de transferência de uma trilha é igual ao tempo de uma rotação, qual é o tempo (em milissegundos) de transferência de uma trilha?
4. Qual é o tempo médio (em milissegundos) estimado para a leitura de uma trilha aleatória do disco?
5. Qual é o tempo médio (em milissegundos) estimado para a leitura de um setor aleatório?
6. Qual seria o tempo estimado para ler 110 trilhas distribuídas aleatoriamente pelo disco?
7. Qual seria o tempo estimado para ler todas as trilhas de um cilindro aleatório?

Usando a linguagem C como referência, responda as questões a seguir.

8. Qual é a diferença em termos de retorno entre as chamadas *open()* e *fopen()*?
9. Conforme mostrado nos slides da aula, o protótipo da função *fread* é o seguinte:

```
int fread (void * ptr, int size, int count, FILE * fd )
```

Explique o que essa função, explicitando o que deve ser passado em cada parâmetro e também o que é retornado pela função.

10. Como você pode usar a chamada de sistema *lseek()* para encontrar a posição corrente em um arquivo?
11. Mostre como mudar as permissões para um arquivo *myfile* de modo que o “dono” do arquivo (*owner*) tenha permissão de leitura e escrita, os membros do grupo (*group*) tenham permissão de execução e os outros (*world*) não tenham permissão nenhuma.

Obs.: O comando para mudança de permissões de arquivos em sistemas *Unix-like* é *chmod pmode nome_do_arquivo*

12. Foi visto em aula um programa C chamado *list* (slide 21 do arquivo OperaçõesArquivos.pdf). Modifique o programa *list* (qualquer uma das versões dadas) para que ele leia os dados a partir da entrada padrão (stdin), em vez de um arquivo, e escreva os dados lidos em um arquivo, em vez da saída padrão (stdout).
13. Analise o código C abaixo e responda as questões a seguir.

```
1. #include <stdio.h>
2. #include <fcntl.h>
3. int main(){
4.     int a,s,i,c;
5.     char b[64];
6.     char n1[20], n2[20];
```

```

7.     c = 0;
8.     gets(n1);
9.     gets(n2);
10.    a = open(n1, O_RDONLY | O_CREAT, 0755);
11.    s = open(n2, O_RDWR | O_APPEND | O_CREAT, 0777);
12.    while ((i = read(a, b, sizeof(b))) > 0){
13.        write(s, b, i);
14.        write(stdout, b, i);
15.    }
16.    lseek(s, 0L, SEEK_END);
17.    write(s, &c, sizeof(c));
18.    close(a);
19.    close(s);
20. }

```

- a. Explique o que acontece quando as linhas 10 e 11 são executadas.
- b. Explique o que acontece quando o laço codificado nas linhas 12 a 15 é executado.
- c. Explique o que acontece quando as linhas 16 e 17 são executadas.

14. Adapte o código mostrado na questão 13 para que sejam usadas as funções da biblioteca *stdio* em vez de chamadas de sistema *open*, *close*, *read*, *write* e *lseek*.