



UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO  
DEPARTAMENTO DE COMPUTAÇÃO

**Bacharelado em Sistemas de Informação**  
**Disciplina: Sistemas Operacionais**  
**Lista – Cap. 4 – Data de entrega: 07/04/2022**

1. Como o MS-DOS implementa o acesso aleatório aos arquivos?  
R= Ele encontra o endereço do primeiro bloco na entrada de diretório, a partir dali, ele segue a cadeia de ponteiros de blocos no FAT até localizar o bloco que precisa. Em seguida, ele guarda o número do bloco para a próxima chamada de sistema de leitura.
2. A chamada de sistema *open* no UNIX é essencial? Quais seriam as consequências de não tê-la?  
R= Não. A consequência seria que o sistema teria que abrir o arquivo como parte da primeira leitura ou gravação. Isso exige uma versão de leitura / gravação com o nome do arquivo, ou então, em todas as leituras seria necessário especificar o nome do arquivo a ser aberto.
3. Sistemas que dão suporte a arquivos sequenciais sempre têm uma operação para rebobinar os arquivos. Os sistemas que suportam acesso aleatório precisam disso também?  
R= Não. Para se ler novamente um arquivo é só acessar o seu byte 0.
4. Alguns sistemas operacionais fornecem uma chamada de sistema *rename* para atribuir um novo nome a um arquivo. Há alguma diferença entre usar esta chamada e apenas copiá-lo para um novo arquivo com o novo nome e depois remover o antigo?  
R= Para o usuário final não, logo que é um processo imperceptível, mas a nível computacional sim, logo que somente renomear de fato um arquivo somente será feita alterações nos blocos que contêm este nome, e remover um arquivo e criar outro com o mesmo conteúdo é um processo mais complicado. Também vale destacar que criando um novo arquivo ele estará em outra posição na memória secundária.
5. Um certo sistema de arquivos usa blocos de disco de 2 KB. O tamanho médio do arquivo é de 1 KB. Se todos os arquivos forem exatamente de 1 KB, qual a fração de espaço em disco que será desperdiçada? Você acha que o desperdício para um sistema de arquivos real será mais alto ou mais baixo que este? Explique.  
R= 50% de desperdício,  $1\text{KB}/2\text{KB} = 0.5$ . Num sistema de arquivos real, o desperdício será menor, pois o sistema terá arquivos maiores que os blocos de disco.
6. Quantas operações em disco são necessárias para buscar o i-node do arquivo `/usr/ast/cursos/os/handout.t`? Suponha que o i-node para o diretório-raiz esteja na memória, mas nenhum outro componente ao longo do caminho se encontre na

**Boa Prova!**

memória. Suponha também que todos os diretórios caibam em um único bloco de disco.

R= O disco lê na seguinte ordem:

diretório para /

i-node para /usr

diretório para /usr

i-node para /usr/ast

diretório para /usr/ast

i-node para /usr/ast/cursos

diretório para /usr/ast/cursos

i-node para /usr/ast/cursos/os

diretório para /usr/ast/cursos/os

i-node para /usr/ast/cursos/os/handout.t

10 leituras são obrigatórias.

7. Cite uma vantagem das ligações estritas (*hard links*) sobre as ligações simbólicas e uma vantagem destas sobre aquelas.

R= Hard links não precisam de espaço em disco extra, apenas um contador no i-node para manter o controle de quantas existem ligações are.Symbolic precisa de espaço para armazenar o nome do arquivo apontado. Os links simbólicos podem apontar para arquivos em outras máquinas. Hard links são restritos a apontando para arquivos dentro de sua própria partição.

8. Duas estudantes de SI estão discutindo sobre i-nodes. A estudante A argumenta que as memórias têm se tornado tão abundantes e baratas que, quando um arquivo é aberto, é mais simples e mais rápido apenas buscar uma nova cópia do i-node na tabela de i-nodes em vez de buscá-lo na tabela inteira e verificar se já está lá. A estudante B discorda argumentando que isso causaria problemas. Qual delas está certa e porque?

R= A estudante B está certa. Ter duas cópias do i-node na tabela ao mesmo tempo é ruim, a menos que as duas sejam de somente leitura. O pior caso é quando as duas estão sendo atualizadas simultaneamente. Quando os i-nodes são gravados de volta no disco, qualquer um que ficar escrito por último vai apagar as alterações feitas pelo outro, e os blocos de disco vão ser perdidos.

9. A alocação contígua de arquivos leva a uma fragmentação do disco. Essa fragmentação é externa ou interna? Faça uma analogia com o que foi discutido no capítulo 3 (gerenciamento de memória).

R= Como o armazenamento desperdiçado está entre as unidades de alocação (arquivos), não dentro delas, isso é fragmentação externa. É precisamente análogo à fragmentação externa da memória principal que ocorre com um sistema de troca (swapping) ou um sistema usando segmentação pura.