

## **Aluna: Anna Júlia Fernandes Barbosa**

01º Resposta: Um driver de dispositivo, ou simplesmente driver, é um componente do sistema operacional responsável por interagir com um controlador de dispositivo. Cada tipo de dispositivo possui seu próprio driver, muitas vezes fornecido pelo fabricante do mesmo. Cada driver é geralmente capaz de tratar um único tipo de dispositivo, ou uma família de dispositivos correlatos do mesmo fabricante. Por exemplo, o driver RTL8110SC(L), da empresa Realtek Corp., serve somente para as interfaces de rede RTL8110S, RTL8110SB(L), RTL8169SB(L), RTL8169S(L) e RTL8169 desse fabricante. Os drivers escondem as diferenças entre os diversos dispositivos, através de uma interface de programação única. A maioria dos SOs define uma interface-padrão para todos os drivers de blocos e uma segunda interface-padrão para todos os drivers de caracteres. Essas interfaces consistem em um número de procedimentos que o restante do SO pode utilizar para fazer o driver trabalhar para ele.

02º Resposta:

(a) O maior espaço útil de disco.

RAID 3: Um disco adicional é usado para armazenar um byte com os bits de paridade dos bytes correspondentes em cada disco, sendo usado para a recuperação de erros nos demais discos.

(b) A maior tolerância a falhas de disco.

RAID 6: Esta redundância extra demanda dois discos adicionais, mas permite tolerar falhas simultâneas de até dois discos.

(c) A maior velocidade média de leitura.

RAID 1: O desempenho em leituras também é beneficiado, pois a controladora pode distribuir as leituras entre as cópias dos dados.

(d) A maior velocidade média de escrita.

RAID 0 (linear): Em teoria, esta estratégia oferece maior velocidade de leitura e de escrita,

pois acessos a blocos em discos físicos distintos podem ser feitos em paralelo.

(e) Equilíbrio entre espaço útil, velocidades e tolerância a falhas.

RAID 5: esta abordagem também armazena informações de paridade para tolerar falhas em blocos ou discos. Todavia, essas informações não ficam concentradas em um único disco físico, sendo distribuídas uniformemente entre eles.

03º Resposta: Um arquivo é uma unidade de armazenamento de informações que podem ser

documentos, imagens, código executável, etc. Além de seu conteúdo, um arquivo é caracterizado por atributos ou metadados, que são informações adicionais relativas ao

conteúdo, e operações ou ações que podem ser realizadas sobre o conteúdo e/ou sobre os

atributos. Os atributos dos arquivos variam de acordo com o sistema de arquivos utilizado.

Os atributos mais usuais, presentes na maioria dos sistemas, são:

Nome: string que identifica o arquivo para o usuário, como “foto1.jpg”, “documento.pdf”, “hello.c”, etc.;

Tipo: indicação do formato dos dados contidos no arquivo, como áudio, vídeo, imagem, texto, etc. Muitos sistemas operacionais usam parte do nome do arquivo para identificar o tipo de seu conteúdo, na forma de uma extensão: “.doc”, “.jpg”, “.mp3”, etc.;

Tamanho: indicação do tamanho do conteúdo do arquivo, geralmente em bytes;

Datas: para fins de gerência, é importante manter as datas mais importantes relacionadas

ao arquivo, como suas datas de criação, de último acesso e de última modificação do conteúdo;

Proprietário: em sistemas multiusuários, cada arquivo tem um proprietário, que deve

estar corretamente identificado;

Permissões de acesso: indicam que usuários têm acesso àquele arquivo e que formas de acesso são permitidas (leitura, escrita, remoção, etc.);

Localização: indicação do dispositivo físico onde o arquivo se encontra e da posição do

arquivo dentro do mesmo;

04º

(a) Resposta: verdadeiro

(b). Resposta: verdadeiro

(c) Resposta: verdadeiro

(d) Resposta: verdadeiro

(e) Resposta: verdadeiro

(f) Resposta: Falso pois o MIME é usado pra identificar arquivos transferidos como anexos de email e conteúdos obtidos de páginas web. Alguns sistemas operacionais, como o BeOS e o MacOS X, definem atributos adicionais usando esse padrão para identificar o conteúdo de cada arquivo dentro do sistema de arquivos.

05º Resposta: Acesso sequencial: No acesso sequencial, os dados são sempre lidos e/ou escritos em sequência, do início ao final do arquivo. Para cada arquivo aberto por uma aplicação é definido um ponteiro de acesso, que inicialmente aponta para a primeira posição do arquivo. A cada leitura ou escrita, esse ponteiro é incrementado e passa a indicar a posição da próxima leitura ou escrita. Quando esse ponteiro atinge o final do arquivo, as leituras não são mais permitidas, mas as escritas podem sê-lo, permitindo acrescentar dados ao final do mesmo. A chegada do ponteiro ao final do arquivo é normalmente sinalizada ao processo através de um flag de fim de arquivo (EoF – End-of-File).

Acesso aleatório: No método de acesso aleatório (ou direto), pode-se indicar a posição no arquivo onde cada leitura ou escrita deve ocorrer, sem a necessidade de um ponteiro de posição corrente. Assim, caso se conheça previamente a posição de um determinado dado no arquivo, não há necessidade de percorrê-lo sequencialmente até encontrar o dado desejado. Essa forma de acesso é muito importante em gerenciadores de bancos de dados e aplicações congêneres, que precisam acessar rapidamente as posições do arquivo correspondentes aos registros desejados em uma operação.

Acesso mapeado: O acesso mapeado em memória é extensivamente usado pelo núcleo para carregar código executável (programas e bibliotecas) na memória. Como somente as partes efetivamente acessadas do código serão carregadas em RAM, esse procedimento é usualmente conhecido como paginação sob demanda (demand paging), pois os dados são lidos do arquivo para a memória em páginas.

Acesso indexado: Alguns sistemas operacionais oferecem também a possibilidade de acesso indexado aos dados de um arquivo, como é o caso do OpenVMS [Rice, 2000]. Esse sistema implementa arquivos cuja estrutura interna pode ser vista como uma tabela de pares chave/valor. Os dados do arquivo são armazenados em registros com chaves (índices) associados a eles, e podem ser recuperados usando essas chaves, como em um banco de dados relacional.

06º Resposta: Travas obrigatórias (mandatory locks): são impostas pelo núcleo de forma incontornável: se um processo obtiver a trava de um arquivo, outros processos que solicitarem acesso ao mesmo arquivo serão suspensos até que aquela trava seja liberada. É o tipo de trava mais usual em sistemas Windows.

Travas recomendadas (advisory locks): não são impostas pelo núcleo do sistema operacional, mas gerenciadas pelo suporte de execução (bibliotecas). Os processos envolvidos no acesso aos mesmos arquivos devem travá-los explicitamente quando forem acessá-los. Contudo, um processo pode ignorar essa regra e acessar um arquivo ignorando a trava, caso necessário.

Travas recomendadas: são úteis para gerenciar concorrência entre processos de uma mesma aplicação. Neste caso, cabe ao programador implementar os controles necessários nos processos para impedir acessos conflitantes aos arquivos compartilhados.

Travas exclusivas (ou travas de escrita): garantem acesso exclusivo ao arquivo: enquanto uma trava exclusiva estiver ativa, nenhum outro processo poderá obter outra trava sobre o mesmo arquivo.

Travas compartilhadas (ou travas de leitura): impedem outros processos de criar travas exclusivas sobre aquele arquivo, mas permitem a existência de outras travas compartilhadas.

07ª Resposta: Semântica imutável: de acordo com esta semântica, se um arquivo pode ser compartilhado por vários processos, ele é marcado como imutável, ou seja, seu conteúdo somente pode ser lido e não pode ser modificado. É a forma mais simples de garantir a consistência do conteúdo do arquivo entre os processos que compartilham seu acesso, sendo por isso usada em alguns sistemas de arquivos distribuídos.

Semântica UNIX: toda modificação em um arquivo é imediatamente visível a todos os processos que mantêm aquele arquivo aberto; essa semântica é a mais comum em sistemas de arquivos locais, ou seja, para acesso a arquivos nos dispositivos locais;

Semântica de sessão: considera que cada processo usa um arquivo em uma sessão, que inicia com a abertura do arquivo e que termina com seu fechamento. Modificações em um arquivo feitas em uma sessão somente são visíveis na mesma sessão e pelas sessões que iniciarem depois do encerramento da mesma, ou seja, depois que o processo fechar o arquivo; assim, sessões concorrentes de acesso a um arquivo compartilhado podem ver conteúdos distintos para o mesmo arquivo. Esta semântica é normalmente aplicada a sistemas de arquivos de rede, usados para acesso a arquivos em outros computadores;

Semântica de transação: uma transação é uma sequência de operações de leitura e escrita em um ou mais arquivos emitidos por um processo e delimitadas por comandos de início e fim de transação (begin ... end), como em um sistema de bancos de dados. Todas as modificações parciais do arquivo durante a execução de uma transação não são visíveis às demais transações, somente após sua conclusão. Pode-se afirmar que a semântica de transação é similar à semântica de sessão, mas aplicada a cada transação (sequência de operações) e não ao período completo de uso do arquivo (da abertura ao fechamento).

08ª Resposta: Os principais elementos que realizam a implementação de arquivos no sistema operacional estão organizados em camadas.

Dispositivos: como discos rígidos ou bancos de memória flash, são os responsáveis pelo armazenamento de dados.

**Controladores:** são os circuitos eletrônicos dedicados ao controle dos dispositivos físicos. Eles são acessados através de portas de entrada/saída, interrupções e canais de acesso direto à memória (DMA).

**Drivers:** interagem com os controladores de dispositivos para configurá-los e realizar as transferências de dados entre o sistema operacional e os dispositivos. Como cada controlador define sua própria interface, também possui um driver específico. Os drivers ocultam as diferenças entre controladores e fornecem às camadas superiores do núcleo uma interface padronizada para acesso aos dispositivos de armazenamento.

**Gerência de blocos:** gerencia o fluxo de blocos de dados entre as camadas superiores e os dispositivos de armazenamento. Como os discos são dispositivos orientados a blocos, as operações de leitura e escrita de dados são sempre feitas com blocos de dados, e nunca com bytes individuais. As funções mais importantes desta camada são efetuar o mapeamento de blocos lógicos nos blocos físicos do dispositivo e o caching/buffering de blocos.

**Alocação de arquivos:** realiza a alocação dos arquivos sobre os blocos lógicos oferecidos pela camada de gerência de blocos. Cada arquivo é visto como uma sequência de blocos lógicos que deve ser armazenada nos blocos dos dispositivos de forma eficiente, robusta e flexível.

**Sistema de arquivos virtual:** o VFS (Virtual File System) constrói as abstrações de diretórios e atalhos, além de gerenciar as permissões associadas aos arquivos e as travas de acesso compartilhado. Outra responsabilidade importante desta camada é manter o registro de cada arquivo aberto pelos processos, como a posição da última operação no arquivo, o modo de abertura usado e o número de processos que estão usando o arquivo.

**Interface do sistema de arquivos:** conjunto de chamadas de sistema oferecidas aos processos do espaço de usuários para a criação e manipulação de arquivos. **Bibliotecas de entrada/saída:** usam as chamadas de sistema da interface do núcleo para construir funções padronizadas de acesso a arquivos para cada linguagem de programação.

**09ª Resposta:** um pequeno conjunto de blocos na área reservada do volume é usado para manter um mapa de bits. Nesse mapa, cada bit representa um bloco lógico da partição, que pode estar livre ou ocupado. O mapa de bits tem como vantagens ser simples de implementar e ser bem compacto: em um disco de 500 GBytes com blocos lógicos de 4.096 bytes, seriam necessários 131.072.000 bits no mapa, o que representa 16.384.000 bytes, ocupando 4.000 blocos (ou seja, 0,003% do total de blocos lógicos do disco).

Por outro lado, a abordagem de alocação FAT (Seção 24.5.3) usa a própria tabela de alocação de arquivos para gerenciar os blocos livres, que são indicados por flags específicos; para encontrar blocos livres ou liberar blocos usados, basta consultar ou modificar as entradas da tabela.

É importante lembrar que, além de manter o registro dos blocos livres, na alocação indexada também é necessário gerenciar o uso dos inodes, ou seja, manter os registros de quais inodes estão livres. Isso geralmente também é feito através de um mapa de bits.

10º Resposta: Os diferentes caracteres especiais usados no caminho de acesso a um diretório, como exemplo, o Windows utiliza “\” como separador em um caminho, já o UNIX usa “/” para a mesma função.