

## Sistemas Operacionais – Avaliação 2º Bimestre - 2019.1

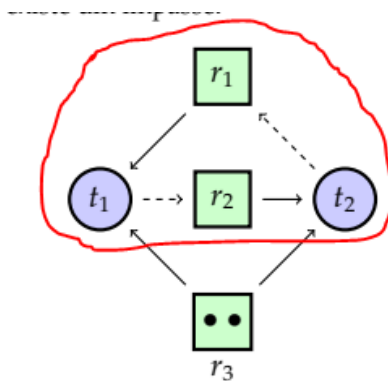
**Aluno:** Rafael Lopes Hozano.

**Matricula:** 20142014040400.

01-Uma vez detectado um impasse, quais as abordagens possíveis para resolvê-lo? Explique-as e comente sua viabilidade.

Utilizar o recurso de abortar a execução dos processos, eliminando ao menos um processo do ciclo, é a forma mais simples de resolver um empasse.

02-Nos grafos de alocação de recursos da figura a seguir, indique o(s) ciclo(s) onde existe um impasse.



03-A figura a seguir representa uma situação de impasse em um cruzamento de trânsito. Todas as ruas têm largura para um carro e sentido único. Mostre que as quatro condições necessárias para a ocorrência de impasses estão presentes nessa situação. Em seguida, defina uma regra simples a ser seguida por cada carro para evitar essa situação; regras envolvendo algum tipo de informação centralizada não devem ser usadas.

**Exclusão mútua:** Cada fluxo de carro está atribuído a um mesmo intervalo de tempo.

**Posse e espera:** Um fluxo de carro pode solicitar nova movimentação em quanto ocorre outras movimentações.

**Não-preempção:** Um movimento concedido a um fluxo de carro somente pode ser liberado pelo fluxo.

**Espera circular:** Existe uma cadeia circular de dependência entre os fluxos.

**Regra:** instalar um semáforo no fluxo azul e amarelo, no qual estando aberto estes, os outros (vermelho e verde) estarão fechados e vice-versa.

04-Explique em que consiste a resolução de endereços nos seguintes momentos: codificação, compilação, ligação, carga e execução.

**Codificação:** programa escolhe a posição de cada variável e do código do programa.

**Compilação:** Compilados escolhe a posição das variáveis na memória, código fonte faz parte do programa que deve ser conhecido no momento da compilação para evitar conflito em endereços na memória.

**Ligação:** Compilador gera símbolos que representem as variáveis.

**Carga:** Define os objetos de variáveis de carga do código em memória para lançamento de novo processo.

**Execução:** São analisados e convertidos pelo processador para a memória final(real).

### 05-Como é organizado o espaço de memória de um processo?

**Text:** Contém o código a ser executado pelo processo, gerado durante a compilação e a ligação das bibliotecas.

**Data:** Dados estáticos usado pelos programas.

Heap: Armazena dados para alocação dinâmica, malloc.

**Free.Slock:** Mantém a pilha de execução do processo.

06-Explique a diferença entre endereços lógicos e endereços físicos e as razões que justificam o uso de endereços lógicos.

Os endereços de memória gerados pelo processador na medida em que executa algum código, são chamados de endereços lógicos, porque correspondem à lógica do programa, mas não são necessariamente iguais aos endereços reais das instruções e variáveis na memória real do computador, que são chamados de endereços físicos.

07-Considerando a tabela de segmentos a seguir (com valores em decimal), calcule os endereços físicos correspondentes aos endereços lógicos 0:45, 1:100, 2:90, 3:1.900 e 4:200.

??

08-Considerando a tabela de páginas a seguir, com páginas de 500 bytes<sup>5</sup>, informe os endereços físicos correspondentes aos endereços lógicos 414, 741, 1.995, 4.000 e 6.633, indicados em decimal.

??

09- Considere um banco de memória com os seguintes “buracos” nãocontíguos:

??

10- O que é uma falta de página? Quais são suas causas possíveis e como o sistema operacional deve tratá-las?

É uma interrupção disparada pelo *hardware* quando um programa acessa uma página mapeada no espaço de memória virtual, mas que não foi carregada na memória física do computador. O sistema operacional tenta tratar esta ocorrência fazendo a página acessível em uma localização da memória física ou interrompendo a execução do processo em caso de acesso ilegal.

11- Por que os sistemas operacionais exigem de todos os drivers de dispositivo (device-drivers) a mesma interface padrão? Não seria mais apropriado deixar cada driver de dispositivo definir rotinas de interface que fazem sentido para aquele tipo específico de dispositivo?

Porque o uso para o usuário ficaria complicado e confuso tendo interfaces diferentes.

12- Você tem 4 discos rígidos de 8 TB cada, que pode organizar de diversas formas. Indique os arranjos RAID que escolheria para obter:

??

13- Apresente e comente as principais formas de atribuição de tipos aos arquivos. Quais são as vantagens e desvantagens de cada uma?

??

14- Sobre as afirmações a seguir, relativas a formatos de arquivos, indique quais são incorretas, justificando sua resposta:

- A) Errada. São bytes no início do conteúdo.
- B) Certa.
- C) Certa.
- D) Errada. Todos os S.O.
- E) Certa.
- F) Certa.

15- Comente as principais formas de acesso a arquivos. Qual o uso mais apropriado para cada uma delas?

**Sequencial:** A gravação de novos registros só é possível no final do arquivo. Exemplo : Fita magnética.

#### **Acesso Direto:**

- É mais eficiente que o seqüencial;
- Permite a leitura/gravação de um registro diretamente na sua posição através do número do registro, que é a posição relativa ao início do arquivo.
- Não existe restrição à ordem em que os registros são lidos ou gravados, sendo sempre necessário especificar o número do registro.
- Possível apenas quando o arquivo é definido com registros de tamanho fixo.

#### **Acesso Direto + Acesso Seqüencial:**

- Possível acessar diretamente um registro qualquer de um arquivo, e, a partir deste, acessar seqüencialmente os demais.

#### **Acesso Indexado ou Acesso por Chave:**

- É o mais sofisticado dos métodos;
- Tem como base o acesso direto;
- O arquivo deve possuir uma área de índice onde existam ponteiros para os diversos registros.
- Quando a aplicação deseja acessar um registro, deverá ser especificada uma chave através da qual o sistema pesquisará, na área de índice, o ponteiro correspondente, a partir disso, acessando diretamente o arquivo.

16- Apresente e explique os quatro principais tipos de travas sobre arquivos compartilhados disponíveis no sistema operacional.

**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.

17-Apresente e explique as quatro principais semânticas de acesso a arquivos compartilhados em um sistema operacional.

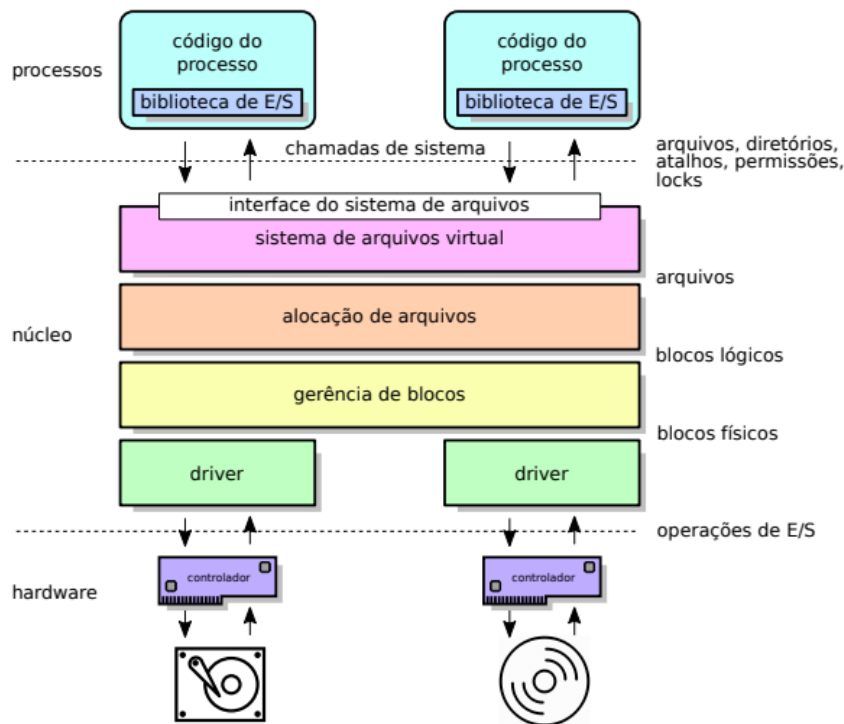
**Semântica imutável:** de acordo com esta semântica, se um arquivo pode ser compartilhado por vários processos, ele é marcado como imutável, ou seja, seu conteúdo somente pode ser lido e não pode ser modificado. É a forma mais simples de garantir a consistência do conteúdo do arquivo entre os processos 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 emitidas 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).

18-Apresente a arquitetura de gerência de arquivos presente em um sistema operacional típico, explicando seus principais elementos constituintes.



**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.

**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.

19-Explique como é efetuada a gerência de espaço livre através de bitmaps.

[illegible]

20-Do ponto de vista lógico, quais as principais diferenças entre a estrutura de diretórios Unix e Windows?

[illegible]