

**Questão 1:** Explique como funciona o mecanismo de DMA (Direct Memory Access).

**Resposta:** Na multiprogramação, a técnica de DMA é útil pois permite que a CPU se dedique a atender/ processar outra tarefa em execução, enquanto a operação de E/S ocorre em paralelo, sem necessitar ficar ociosa esperando a E/S. O DMA é útil para operações em que haja a necessidade da transferência de informação em blocos de algum dispositivo para a memória principal, com o intuito do uso posterior dos dados pela CPU. O DMA permite que os dados sejam transferidos sem a necessidade do gasto de computação durante a transferência, liberando a CPU para outro processamento durante este período. Após a conclusão da transferência, o chip DMA manda um sinal de interrupção para a CPU, informando-a do fim. Uma organização sem DMA na multiprogramação, torna mais lenta a transferência de dados presente em uma operação de E/S, pois sempre que houver a transferência de dados entre a memória e o dispositivo, haverá utilização da CPU; enquanto que através do DMA, a configuração da transferência de vários blocos é feita somente no início da transação

**Questão 2:** Considere um sistema de arquivo baseado em i-nodes, que utiliza blocos de dados de 1KB e 32 bits para o endereçamento de cada bloco. A figura 1 mostra a estrutura de cada i-node, que possui 12 campos que apontam para blocos, sendo 10 apontamentos diretos, um indireto simples e um indireto duplo. Qual é o máximo tamanho de arquivo permitido neste sistema?

**Questão 3:** Considere um sistema de arquivos baseado em i-node. Qual a diferença entre links simbólicos e hard links?

**Resposta:** Um **link simbólico (soft link)** se comporta como um "atalho" para outro arquivo ou diretório. Ou seja, ele praticamente só aponta para este outro arquivo. Ao deletar o arquivo original, o link simbólico irá falhar.

Ao deletar o link simbólico, nada irá acontecer com o arquivo original.

Um **hard link** é um ponteiro para o inode de um arquivo ou diretório.

Ao deletar o arquivo original ou o hard link, o outro ainda existe (porque o inode é mantido)

Ao modificar qualquer link apontando para um mesmo inode, todos links, incluindo o arquivo original, são modificados.

**Questão 4:** Explique como funciona Mapa de Bits, utilizado por diversos sistemas de arquivos para o controle de blocos livres e ocupados.

**Resposta:** Forma mais simples de implementar uma estrutura de espaços livres; cada entrada da tabela é associada a um bloco do disco representado por um bit que pode ser 0 (livre) ou 1 (ocupado).

**Questão 5:** Cite e explique pelo menos dois tipos de inconsistência que um desligamento abrupto do sistema pode ocasionar ao sistema de arquivo.

**Resposta:** O controle de consistência se dá em dois níveis: blocos e arquivos. Para controle de consistência no nível de bloco, o utilitário constrói uma tabela com dois contadores por bloco, ambos iniciados em 0. O primeiro contador rastreia quantas vezes o bloco aparece no arquivo; o segundo registra com que frequência ele aparece na lista de blocos livres. O utilitário lê todos os i-nodes. Começando de um i-node, é possível construir uma lista de todos os números de blocos usados no correspondente arquivo. Assim que cada número de bloco é lido, seu respectivo contador na primeira tabela é incrementado. A seguir, é examinada a lista de blocos livres rastreando todos os blocos que não estão em uso. Cada ocorrência de um bloco na lista de blocos livres resulta no incremento do respectivo contador na segunda tabela. Se o sistema de arquivo for consistente, cada bloco terá o valor 1 na primeira tabela ou na segunda tabela. Contudo, em caso de falha, pode detectar-se blocos que não ocorrem em nenhuma das tabelas (blocos perdidos). Embora blocos perdidos não causem um dano real, eles desperdiçam espaço, reduzindo assim a capacidade do disco. A solução para blocos perdidos é direta: o verificador do sistema de arquivos acrescenta-os na lista de blocos livres. Outra situação possível de ocorrer é a repetição de blocos na lista de blocos livres. A solução neste caso também é simples: reconstruir a lista de blocos livres, eliminando-se as duplicações.

**Questão 6:** Explique o funcionamento do cache de blocos de disco.

**Resposta:** Cache de bloco ou cache de buffer; é um conjunto de blocos que pertencem logicamente ao disco, mas são colocados na memória para melhorar o desempenho do sistema (reduzir acesso em disco); quando um bloco é requisitado, o sistema verifica se o bloco está na cache; se sim, o acesso é realizado sem necessidade de ir até o disco; caso contrário, o bloco é copiado do disco para a cache.

**Questão 7:** Explique como funciona o algoritmo de agendamento do braço de disco do elevador.

**Resposta:** Requisições chegam continuamente e aleatoriamente □ Muitos elevadores tentam conciliar eficiência e justiça □ Continuam se movendo na mesma direção até não haver mais requisições pendentes naquela direção □ Então trocam de direção.

No disco, o driver deve manter 1 bit a direção (up ou down). Quando uma

requisição termina, o driver verifica o bit. Se for up, o braço é movido à próxima requisição mais alta. Se não houver requisições pendentes nessa direção, o bit é feito down, e o braço se move à próxima requisição mais baixa.

**Questão 8:** Explique como funciona o algoritmo de agendamento do braço de disco FCFS (First-Come, First-Served).

**Resposta:** O driver aceita uma requisição por vez, e as executa nessa ordem. Pouco pode ser feito para otimização. As requisições pendentes são colocadas na tabela e mantidas em lista ligada uma para cada cilindro requisitado.

**Questão 9:** Quais as vantagens do cache de blocos de disco mantido pelo kernel? Qual (is) problema (s) o uso deste cache pode ocasionar?

**Resposta:** **Vantagens Buffer Cache**

1. Minimização do número de transferências entre o disco e a memória (cache hits);
2. O acesso a disco é tratado de uma forma uniforme. Existe apenas uma interface para acessar os dados de um disco;
3. Ajudar a manter a coerências dos blocos. Se dois processos tentam acessar o mesmo bloco a buffer cache ordena os acessos;

**Desvantagens da Buffer Cache**

4. Torna o sistema mais vulnerável a acidentes de parada total (falta de energia, por exemplo) ("crashes");
5. Torna a transferência de grandes quantidades de informação mais lenta.

**Questão 10:** O que é deadlock?

**Resposta:** Um conjunto de processos estará em situação de deadlock se todo processo pertencente ao conjunto estiver esperando por um evento que somente um outro processo desse mesmo conjunto poderá fazer acontecer

**Questão 11:** Cite e explique um mecanismo que pode ser utilizado para sincronismo de processos.

**Resposta:** Muitas vezes, em uma aplicação concorrente, é necessário que processos se comuniquem entre si, e essa comunicação pode ser implementada de diversos mecanismos, como variáveis compartilhadas na memória principal ou troca de mensagens. Nesta situação, é necessário que os processos tenham sua execução sincronizada pelo sistema operacional.

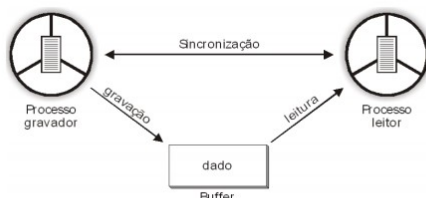


Figura 1 – Sincronização e comunicação entre processos

**Questão 12:** Explique o mecanismo de IPC por passagem de mensagens.

**Resposta:** Ao fornecer mecanismos de IPC, o S.O implementa “Canais” de comunicação (implícitos ou explícitos) entre processos.

Fundamentalmente, existem duas abordagens:

- Suportar alguma forma de espaço de endereçamento compartilhado.
- Utiliza comunicação via núcleo do S.O., que ficaria então responsável por transportar os dados de um processo a outro.

São exemplos:

Pipes e Sinais (ambiente centralizado)

Troca de Mensagens (ambiente distribuído)

RPC – Remote Procedure Call (ambiente distribuído)

**Questão 13:** Explique o que é e qual a função da MMU.

**Resposta:** É um dispositivo de hardware que traduz endereços virtuais em endereços físicos, é geralmente implementada como parte da CPU, mas pode também estar na forma de um circuito integrado separado. Um MMU é efetivo em gerenciamento de memória virtual, manipulação e proteção de memória, controle de cache e, em arquiteturas mais simples de computador, como em sistemas de 8 bits, bank switching.

**Questão 14:** Explique o que é TLB (Translation Lookaside Buffer ).

**Resposta:** também conhecida como memória associativa, é um dispositivo de hardware cujo propósito é mapear endereços virtuais em endereços físicos sem passar pela tabela de páginas. Usualmente, ela faz parte da MMU.

TLBs aceleram a tradução entre endereços virtuais e reais

**Questão 15:** Explique o que são Tabelas de Páginas Invertidas e qual é a vantagem de sua utilização.

**Resposta:** Ela possibilita o aumento da quantidade de molduras (memória real) e não da quantidade de páginas (memória virtual).

Se memória real é de 256 Mbytes e páginas de 4 KB: Tem-se 65536 entradas

**Questão 16:** Considere um sistema com memória física de 32KB, tabelas de páginas com 16 entradas e endereçamento virtual que utiliza 12 bits de deslocamento (offset).

Responda:

- a). Qual o tamanho total do endereço virtual (em bits)?
- b). Qual o tamanho de cada página?
- c). Determine quais entradas da tabela de páginas correspondem aos endereços lógicos: 3780, 11264, 61439 e 36875.
- d). Considere a tabela de páginas representada pela Tabela 1. Forneça o endereço físico dos seguintes endereços lógicos: 2000, 16484 e 12588.
- e). Considere que as páginas físicas foram alocadas na tabela de páginas na seguinte ordem: 3, 5, 4, 7, 6, 1, 0 e 2. Utilizando o algoritmo de substituição de páginas FIFO, forneça a tabela de páginas após a requisição ao das seguintes páginas virtuais: 12, 13, 10 e 5.

| Entrada | Página Física |
|---------|---------------|
| 0       | 3             |
| 1       | —             |
| 2       | 5             |
| 3       | 0             |
| 4       | 1             |
| 5       | —             |
| 6       | 6             |
| 7       | —             |
| 8       | —             |
| 9       | —             |
| 10      | —             |
| 11      | 7             |
| 12      | —             |
| 13      | 4             |
| 14      | 2             |
| 15      | —             |

**Questão 17:** Qual a importância do gerenciamento de memória por parte do SO?

**Resposta:** Em sua forma mais simples, está relacionado em duas tarefas essenciais:

- **Alocação:** Quando o programa requisita um bloco de memória, o gerenciador o disponibiliza para a alocação;
- **Reciclagem:** Quando um bloco de memória foi alocado, mas os dados não foram requisitados por um determinado número de ciclos ou não há nenhum tipo de referência a este bloco pelo programa, esse bloco é liberado e pode ser reutilizado para outra requisição.

Ou melhor:

Os que não realizam paginação ou troca entre disco e memória

O que não realizam este processo.

**Questão 18:** Um programador C desatento cometeu um erro no código de seu programa, fazendo com que em determinada condição, o programa acessasse um endereço de memória inválido, ou seja, fora do segmento de dados e/ou pilha. Descreva as ações efetuadas pelo processador e SO (kernel) desde o momento em que a instrução com o endereço inválido é executada até o momento em que o programa é finalizado, onde a mensagem Segmentation Fault (Falha de Segmentação) é exibida ao usuário.

**Resposta:** Descobrimos que há um ponteiro inválido na memória e o programa "crashou" quando tentamos acessá-lo. As duas opções a partir de agora são (1) examinar o código e descobrir que código inseriu um ponteiro inválido em memória ou (2) continuar usando o auxílio do gdb para descobrir a fonte do erro.

**Questão 19:** Cite e explique pelo menos dois algoritmos de troca de páginas de memória.

**Respostas:** O FIFO é um algoritmo de substituição de páginas de baixo custo e de fácil implementação que consiste em substituir a página que foi carregada há mais tempo na memória (a primeira página a entrar é a primeira a sair). Esta escolha não leva em consideração se a página está sendo muito utilizada ou não, o que não é muito adequado pois pode prejudicar o desempenho do sistema.

LRU é um algoritmo de substituição de página que apresenta um bom desempenho substituindo a página menos recentemente usada. Esta política foi definida baseada na seguinte observação: se a página está sendo intensamente referenciada pelas instruções é muito provável que ela seja novamente referenciada pelas instruções seguintes e, de modo oposto, aquelas que não foram acessadas nas últimas instruções também é provável que não sejam acessadas nas próximas.

**Questão 20:** Considere a Tabela 2, que representa a Tabela de Alocação de Arquivo de um determinado sistema de arquivos que utiliza blocos de dados de 4KB.

|    |    |
|----|----|
| 0  | —  |
| 1  | 4  |
| 2  | —  |
| 3  | 5  |
| 4  | 2  |
| 5  | 8  |
| 6  | 11 |
| 7  | 6  |
| 8  | 9  |
| 9  | 10 |
| 10 | 13 |
| 11 | 12 |
| 12 | 14 |
| 13 | —  |
| 14 | —  |
| 15 | 13 |

- Como funciona a Tabela de Alocação de Arquivo?
- Quais blocos pertencem ao arquivo?
- Qual o tamanho total do arquivo?