

Exercício 1) Resolva todos os exercícios da Lista de Exercícios P1

Exercício 2) Defina Thread. O que difere três processos monothreads de um processo com três threads? O que as threads de um mesmo processo compartilham?

- Thread é uma forma de um processo se dividir em duas ou mais tarefas que podem ser executadas simultaneamente. Um processo com uma única thread só faz uma coisa por vez e se der problema tem que ser fechado, um multithread faz mais de uma coisa e só uma thread morre. Mesmo lugar na memória, dados e arquivos.

Exercício 3) Uma thread é sempre bom para aplicativos orientados a E/S? Mesmo em computadores monoprocessados? E para aplicativos orientado a CPU? Sempre é bom?

- Sim, porque enquanto uma thread está bloqueada aguardando o E/S, as outras estão trabalhando. Sim, existe escalonamento. Não necessariamente, se for monoprocessado, não tem nenhum ganho.

Exercício 4) Sobre thread, quais são as possíveis formas de implementar um Servidor? O que são Threads Despachantes e operárias?

- Threads – com paralelismo, chamadas ao sistema com bloqueio;
- Monothread – sem paralelismo, chamadas ao sistema com bloqueio;
- Estado finito – com paralelismo, chamadas ao sistema sem bloqueio, interrupções.
- Despachante recebe requisições e passa o trabalho pra um das operárias fazer.

Exercício 5) Explique e aponte as vantagens e desvantagens das implementações de thread no espaço de usuário, núcleo e híbrido (n.m).

- Usuário – mais fácil de invocar, mas as chamadas ao sistema tem bloqueio.
- Núcleo – criar e destruir é feito por chamadas ao sistema, mas o núcleo tem alto custo para gerenciar a execução de threads.
- Híbridas – thread com múltiplos usuários sobre uma thread de núcleo. Cria uma nova thread pra lidar com uma nova chamada.

Exercício 6) O que é uma condição de disputa? Quando ocorre? Como identificar e proteger uma região crítica? Por que deve ser protegida?

- É quando dois processos tentam fazer a mesma coisa. Quando há concorrência, objetos

compartilhados e mudança de estado. Região crítica é um local onde dados compartilhados são acessados. Deve ser protegida porque se for acessada por mais de um processo ao mesmo tempo pode causar problemas.

Exercício 7) Desabilitar a interrupção da CPU pode proteger um processo ao entrar em uma região crítica? Quais as vantagens e os problemas deste método?

- Sim porque garante que o processo termine de usar determinado recurso antes de sair de execução. Não é legal dar esse poder a um processo, não funciona com vários processadores, impedir interrupções pode dar merda.

Exercício 8) Cite e explique um algoritmo que resolve via software o acesso em uma região crítica.

- Algoritmo de Peterson – usa flags e uma variável turn pra impedir que outro processo acesse a região crítica quando um estiver fazendo isso.

Exercício 9) O que é TSL (*Test and Set Lock*)? Esta é uma solução por software ou hardware? Qual a vantagem e relação a uma solução por software?

- Uma trava que bloqueia o barramento de memória até que a instrução tenha terminado. Hardware. Fácil de usar e não dá aos processos a possibilidade de desabilitar interrupções, funciona em máquinas com vários processadores, mas busy wait (um processo entra em um laço inútil) e starvation (sempre que um processo tenta executar, pega lock em 1).

Exercício 10) Explique o conceito de dormir e acordar em IPC (*sleep* e *wakeup*).

- Sleep bloqueia um processo e faz ele esperar um sinal (não tem busy wait) e wakeup sinaliza pra acordar o processo.

Exercício 11) O que é um semáforo? Como funciona? Quais são as operações/funções que podem ser aplicadas no semáforo? Quanto utilizar? Cite exemplos.

- Uma variável que só pode ser testada (P) ou incrementada (V). P decrementa o semáforo bloqueia o processo ou manda ele pra fila e V incrementa e acorda.

Exercício 12) O que é um mutex? O que é um monitor? Quais as diferenças, vantagens e desvantagens entre eles? Compare com o Semáforo.

- Semáforo binário. Monitores marcam a região crítica e assim o compilador garante a exclusão

mútua. É mais simples, mas nem toda linguagem tem monitores.

Exercício 13) Como o uso de “Passagem de Mensagem” pode resolver o problema da região crítica? Quais as vantagens e desvantagens desta abordagem?

- Não compartilham memória. Não há região crítica.

Exercício 14) Sobre sincronismo entre processos, o que são Barreiras? As barreiras servem para evitar acesso concorrente em região crítica? Quando utilizá-los?

- Serve para sincronizar processos. Não, servem para bloquear processos até que eles estejam no mesmo ponto, então librá-los.

Exercício 15) Descreva e exemplifique o problema do Jantar dos Filósofos.

- Filósofos são processos e talheres são recursos compartilhados. Um trava o outro.

Exercício 16) Descreva e exemplifique o problema do Produtor e Consumidor.

- Dois processos compartilham o mesmo buffer. Um preenche as informações e o outro consome. Não dá para fazer um sem o outro.

Exercício 17) Descreva e exemplifique o problema do Barbeiro Sonolento.

- Buffer de espera de cliente. Três barbeiros cortam, até 7 clientes aguardam. Se não tiver nada, dorme. Usa mutex para proteger a região crítica.

Exercício 18) Descreva e exemplifique o problema dos Leitores e Escritores.

- Banco de dados. Um escritor, vários leitores. Usa mutex para proteger a região crítica.

Exercício 19) Sobre gerenciamento de memória, o que é endereçamento lógico e endereçamento físico?

- Lógico é um endereço virtual gerado pela CPU ou pelo programa. Programas de usuário veem apenas endereços lógicos. Físico é a posição física dos dados na memória. Lógico vira físico no momento da execução do programa.

Exercício 20) O que é a MMU? Quais são as suas atribuições?

- Memory Management Unit (unidade de gerência de memória). Faz o mapeamento entre

endereço lógico e endereço físico.

Exercício 21) Como o endereçamento de memória é amarrado no programa? Em quais fases o endereçamento pode ser amarrado ao programa? Qual o papel da MMU em cada possibilidade?

- O endereço da posição de memória pode ser ligado ao programa em três momentos:
- Compilação – se a localização da memória é conhecida. Se mudar, tem que recompilar.
- Carga – o código é relocável caso a localização da memória não seja conhecida na compilação.
- Execução – na hora que é executado move o processo pro lugar.

Exercício 22) Sobre gerenciamento de memória, qual a diferença entre alocação contínua e não contínua? Sobre alocação contínua, qual a diferença entre alocação simples e particionada? Cite um exemplo para cada.

- Alocação contígua simples divide a memória principal em duas partições e o usuário tem controle total. Particionada são múltiplas partições impostas pela multiprogramação.

Exercício 23) Sobre alocação contínua particionada, qual a diferença da partição estática e a dinâmica? Quais as vantagens e desvantagens de cada uma?

- Estática tem partições de tamanho fixo e dinâmica tem partições com tamanhos variados.
- Estática – uso ineficiente da memória, não importa o tamanho do processo, sempre ocupa uma partição inteira. Dinâmica reduz a fragmentação interna.

Exercício 24) Nas alocações contínuas particionadas fixas, as partições sempre têm o mesmo tamanho? Justifique.

- Sim, porque se não tiverem ela será dinâmica, não fixa.

Exercício 25) O que é um código relocável? Quais os benefícios de um código realocável em sistemas de gerenciamento de memória com alocações contínuas?

- Relocável pode ser alocado em qualquer parte da memória. Se todas as partições estão ocupadas, é possível realocar, tirando temporariamente um processo da memória e colocando outro. Não fica sem executar.

Exercício 26) O que é Swapping no contexto de gerenciamento de memória?

- Rodízio de processos na memória. É movido pra fora, e depois volta.

Exercício 27) Comente sobre políticas de alocação de partições fixas. Qual o benefício de um código relocável neste contexto? E do swapping?

- Tamanho igual → não importa qual partição é usada. Tamanhos diferentes → atribui ao processo a menor partição livre capaz de armazenar o processo. Minimiza a fragmentação interna. Swap permite ter mais processos que partições.

Exercício 28) O que é fragmentação interna e externa? Quando acontece?

- Interna → dentro da partição. Processos menores que o tamanho da partição armazenados dentro dela deixam espaços vazios. Externa → partições variáveis pequenas demais pra armazenar um processo, vão ficando vazias.

Exercício 29) Sobre o gerenciamento de memória contínua particionada dinâmica, quando ocorre a fragmentação externa? E a fragmentação interna? Quais são os principais algoritmos de alocação contínua dinâmica? Simule cada algoritmo citado.

- Um processo pega uma partição de 2k, sai e entre outro de 1,5k, fica 0,5k vazio. Interna não há porque as partições vão sendo criadas conforme entram os processos.
- Best fit – pega o menor bloco livre possível, pra deixar espaços pequenos.
- Worst fit – pega o maior bloco livre pra deixar espaços grandes.
- First fit – pega o primeiro espaço livre possível.
- Next fit – igual, mas circular.

Exercício 30) Em gerenciamento de memória, o que é paginação? Como a paginação diminui a fragmentação interna e elimina a fragmentação externa?

- Dividir a memória em pequenos quadros de 4k e a memória lógica em páginas. Uma página é ligada a um quadro. Não precisa estar tudo alocado em sequência. Como está tudo na tabela de páginas, dá pra colocar um pedaço em cada lugar da memória. Elimina a externa porque ocupa todos os lugares. Interna só na última página.

Exercício 31) Contextualize página, quadro (frame) e MMU. Como é feita a tradução do endereçamento lógico para o endereçamento físico? Para que serve as tabelas de paginação? O que são tabelas multiníveis? Quando adotá-la?

- A MMU tem uma tabela com qual página é ligada a um frame. CPU tem apenas o endereço

lógico. MMU procura na tabela e encontra o físico. Cada processo tem uma tabela de páginas. Tabela principal e outras em níveis. Fica menor assim.

Exercício 32) Quais as vantagens e desvantagens de estabelecer páginas muito grandes? E muito pequenas?

- Grandes → processos tem menos páginas, mas tem mais fragmentação interna. Pequenas → o contrário.

Exercício 33) É possível adotar uma política de proteção na paginação? Como? E política de compartilhamento? Cite um exemplo prático para ambos os casos.

- Bits de controle nas tabelas. Controle se a página é read-only, read-write, etc. Um bit de controle de compartilhamento.

Exercício 34) O que é Memória Virtual?

- Extensão da memória física. Move informações pro disco pra liberar memória ram.

Exercício 35) Como implementar a Memória Virtual através de Paginação?

- Com paginação o tamanho dos frames é pequeno, isso diminui o tempo de swapping pra trazer informações do disco pra memória e vice-versa. Não precisa mover todo o programa, só algumas páginas.

Exercício 36) O que acontece quando uma página não está disponível na memória principal, e sim na memória secundária? O que é Thrashing? Quando ocorre?

- Dá uma falha e gera uma interrupção. Enquanto o SO traz a página, o processo fica em espera. Thrashing é quando o SO passa mais tempo trocando páginas que executando instruções. Quando há mais processos do que a memória pode aguentar.

Exercício 37) Qual a diferença de paginação por demanda e paginação por antecipação? Quais são as políticas de substituição de página? Descreva e exemplifique cada algoritmo de substituição de página citado.

- Demanda → só carrega algumas páginas com o programa e vai trazendo o resto conforme houverem requisições. Antecipação → antes do processo ser executado, o SO já carrega as páginas na memória.

- Working set – retira páginas que não estejam sendo usadas por nenhum processo.
- Aleatório – isso aí
- Não acessada a mais tempo – exige um temporizador. Gasta bastante espaço.
- Não usado frequentemente – contador de quantas vezes usou a página. Páginas antigas não saem
- Aging – bit a esquerda vai rodando pra direita. Evita deixar páginas antigas eternamente.

Exercício 38) Relacione a quantidade de falha de página e o uso de memória. Qua a conclusão desta relação? Justifique a sua resposta.

Exercício 39) Onde ficam as tabelas de páginas? São grandes? Elas podem sofrer paginação? Justifique!

Exercício 40) O que é TLB? Como o TLB pode evitar o acesso duplo à memória principal? Justifique a sua resposta.

Exercício 41) Descreva as ações da TLB nos seguintes casos: (1) A tabela de página já está na TLB; (2) A tabela de página está na memória primária e; (3) A tabela de página está na memória secundária.

Exercício 42) O que é segmentação? Quais as vantagens e desvantagens do uso da segmentação? Existe fragmentação (interna ou externa) na segmentação?

Exercício 43) Existe compartilhamento e proteção nos segmentos? Como? Quais as vantagens?

Exercício 44) Compare segmentação com paginação. É possível implementá-los juntos? Como? Neste contexto, o que é um endereço lógico, linear e físico?

Exercício 45) Defina o que é um periférico. Como é feita a comunicação entre o SO e o dispositivo de E/S? Qual o papel dos *Drivers* e dos Controladores?

Exercício 46) Quais são os tipos de dispositivos de E/S? (orientado a bloco, orientado a caractere...) Quais as diferenças entre eles?

Exercício 47) Como podemos classificar os tipos dos dispositivos?

Exercício 48) O que significa DMA? Quais as vantagens? Quando utilizá-lo?

Exercício 49) O que é um Buffer? Qual o papel de um Buffer no sistema de E/S?

Exercício 50) Compare os tipos de armazenamento secundários: Fita Magnética, Disco Óptico, Disco Magnético e Discos SSD. Cite as vantagens e desvantagens de cada um.

Exercício 51) Sobre discos magnéticos, o que é tempo de busca? E tempo de atraso rotacional? Quais são as políticas de escalonamento de disco que podem ser adotadas

para minimizar o problema do tempo de busca?

Exercício 52) Quais são os tipos de interface de disco?

Exercício 53) O que é RAID? Compare RAID 0, 1, 2, 3, 4, 5, 6, 10 e 50. Cite um exemplo de uso para o RAID 0 e 5. Quando utilizar o RAID 50?