

Lista de Exercícios de INF1613

Obs: estas questões servem apenas para seu “aquecimento” na matéria, e para revisão dos conceitos vistos na disciplina. A chance de caírem na prova é de apenas 0.01%

Introdução

1. Quem foi o inventor e primeiro desenvolvedor do sistema Unix? Em que situação isso aconteceu?
2. Que papel a linguagem C teve na popularização do sistema operacional Unix, e por que?
3. Quais são as principais abstrações de “processamento”, de “persistência de dados” e de “grupos de dados” que o Unix apresenta ao usuário (e que foram posteriormente adotados por todos os sistemas operacionais modernos)?
4. O que é um sistema multiprocessamento (ou multi-tarefa)? Como se previne a interferência de um processo na execução de um outro processo?
5. O que é um sistema multi-usuário? Quais cuidados precisa-se tomar para que os usuários usem a mesma máquina “harmonicamente”?
6. Como saber se um processo pode ter acesso a um arquivo, por exemplo, para escrita no arquivo? Como isso é verificado pelo sistema operacional?
7. Qual é a diferença entre o *modo núcleo* e o *modo usuário* de processamento?

Processos

8. Defina o que é um processo?
9. Quais são os possíveis estados de um processo? Quais são os eventos que ocasionam a mudança de um estado para outro,? Para cada um dos estados acima.
10. Dê três exemplos de processos que oferecem serviços do sistema operacional. Por que você acha que esses serviços não são realizados diretamente pelo núcleo?
11. Perante o núcleo, há diferença entre um processo de sistema e um processo de um aplicativo (p.ex. um editor de texto)?
12. O que acontece (no núcleo) quando o seu programa executa a chamada `fork()`, sem um posterior `exec()`;
13. Porque o projetista do Unix optou por criar duas chamadas – `fork` & `exec` – em vez de uma só, por exemplo: `run-executable()`
14. Quais são as seções de memória RAM que estão disponíveis para cada processo?
15. Um processo pode não ter um processo pai? Pode ter mais de um processo pai? Explique.
16. Diga quais tipos gerais de interrupção existem. Interrupções podem se interromper? Dê exemplos concretos de interrupções.
17. O que faz o tratamento de interrupção de primeiro nível? E o `dispatching`?

18. Como o núcleo lida com interrupções simultâneas, ou muito próximas uma da outra.
19. O que significa troca de contexto? E o que exatamente compreende o contexto de um processo? Que componente do sistema operacional realiza a troca de contexto?
20. O que é o vetor de interrupções? O que contém cada entrada do vetor? E como é obtido o índice para o vetor de interrupção?
21. Por que o núcleo também precisa manipular uma pilha local (kernel stack)?

Comunicação e Sincronização de Processos

22. Explique o que é uma condição de corrida, e dê um exemplo do dia-a-dia (que não seja o exemplo da conta bancária compartilhada).
23. Explique qual é a diferença entre a espera ocupada e a alternativa de bloqueio na implementação de sessão crítica.
24. Para um semáforo, o que garante a atomicidade da consulta e atualização ao contador do semáforo.
25. Assuma que seu sistema só implementa semáforos binários. Como você implementaria semáforos contadores usando esses semáforos binários? Mostre os pseudo-códigos para as operações `SC_Init(int x)`, `SC_Down()` e `SC_Up()`.
26. Explique o que é uma variável de condição, e em qual situação ela encontra uso.
27. Quais são as vantagens e desvantagens do uso de Monitores para a sincronização no acesso concorrente de estruturas de dados compartilhadas, quando comparada ao uso de semáforos?
28. Como você implementaria a exclusão mútua da execução dos procedimentos de um monitor usando semáforos.
29. O Test-and-Set-Lock (TSL) é uma instrução de máquina que é usada para espera ocupada, e que funciona bem para arquiteturas multi-core. Discuta os problemas de uma sincronização por bloqueio em um sistema multi-core, onde threads rodando em diferentes cores precisam se sincronizar (p.ex. controlar a entrada em uma região crítica).

Escalonamento

30. Considere os seguintes 5 processos que são iniciados nessa ordem no instante $t=0$.

Process	Temp	Process	Priority
P1	10		3
P2	1		1
P3	2		3
P4	1		4
P5	5		2

- a. Desenhe o Diagrama de Grant¹ ilustrando a execução desses processos usando políticas de escalonamento: FIFO, SJF, Prioridade-não preemptiva, e RoundRobin (quantum = 1).
 - b. Qual é o “turnaround” (tempo de giro) de cada processo para cada uma das políticas.
 - c. Qual é o tempo de espera de cada processo para cada uma das políticas.
 - d. Qual das políticas de escalonamento gera o menor tempo médio de espera (para todos os 5 processos)
31. Por que o atraso médio do escalonamento de N jobs usando Round Robin é sempre maior do que usando escalonamento ShortesJobFirst. Justifique.

Gerenciamento de Memória

32. Qual é principal tarefa da Memory Management Unit (MMU)?
33. Qual é a vantagem em se distinguir endereços virtuais de endereços lógicos? [Com resposta.](#)
34. Explique as etapas que compõem um page fault.
35. Qual é o principal benefício de usar um TLB, e a influência de seu tamanho.
36. É possível que ocorra um page-fault para um endereço de um processo P1, mesmo que na conversão do endereço lógico para o físico tenha havido um TLB-hit? Justifique. [Com resposta.](#)
37. Seja uma arquitetura de 32 bits. Sabendo-se que os 15 bits mais significativos são usados como índice para a entrada na tabela de páginas, e que a memória do seu computador possui 10 MBs livres para os processos. Quantos quadros de página ela comporta para os processos? [Com resposta.](#)
38. Dada uma memória física com 4 quadros de página (frames) e a seguinte sequência de acessos a índices de páginas. Então simule os algoritmos LRU, FIFO, e Ótimo e conte o número total de Page-faults (PF) que cada algoritmo ocasiona:
- 1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6
- [Com resposta.](#)
39. Indique pelo menos 3 dados/informações que estão invariavelmente estão presentes em todas as entradas da tabela de páginas.
40. que é gerenciamento de memória através de overlay? Dê um exemplo de um programa que poderia ter overlays.

¹ O Diagrama de Grant mostra os períodos de tempo em que um processo detém o controle da CPU e os períodos em que está no estado pronto (ou bloqueado).

Sistema de Arquivos

39. Quais partes do Sistema de Arquivos executam no núcleo? [Com resposta.](#)
40. Como sistema de arquivos consegue diminuir o tempo médio geral de acesso ao disco, para todos os processos? Pense no acesso a blocos consecutivos.
41. O que é um i-node, e para que serve?
42. Uma partição pode conter mais i-nodes em uso do que blocos?
43. Dentre os atributos (meta-dados) de um arquivo existe um contador de links. Para que serve?
44. Seja um disco em que o bloco de disco tem 1K bytes e o endereço de cada bloco ocupa 64 bits, quantos blocos um arquivo de tamanho 513 K ocupa no disco? [Com resposta.](#)
45. Para o arquivo da questão anterior, quantos blocos do disco são lidos do disco quando se acessa o endereço a) 4097; b) 12K; c) 13K, d) 512K. [Com resposta.](#)
46. O que significa montar um sistema de arquivos SA no diretório /A/B/C de um sistema de arquivos SB?
47. Digamos que exista um arquivo com o path /A/B/C/x no SB, um arquivo também chamado x em /x do SA. O que acontece após o mount de SA em SB?
48. A partição de swap também possui i-nodes e blocos? Justifique.

Respostas de algumas questões

Lista de Exercícios de INF1019

Versão 2018

Gerenciamento de Memória

33: Resposta: processos podem ser realocados em diferentes partes da memória, já que a tradução é feita a cada vez que a CPU expele um endereço lógico.

36: Resposta: Sim. Basta que a página correspondente tenha sido removida pelo núcleo a pedido de outro processo, P2. Apesar da tabela de páginas de P1 ter sido atualizada para bit R=0, isso não necessariamente ocorre com o conteúdo do TLB, sobretudo se é usado o algoritmo WorkingSet, onde o conteúdo do TLB faz parte do contexto de P1 e é carregado na MMU quando o processo (P1) é escalonado.

37: Resposta: 80 quadros

38: Resposta: LRU: 10 PFs; Ótimo: 8 PFs; FIFO: 14 PFs

Sistema de arquivos

39: Resposta: Apenas o caches de i-nodes, o cache de blocos, a mount table, e a file table (onde estão relacionados todos os arquivos por qq processo).

44: Resposta: Note que em 1K de bloco de índice cabem 128 endereços de blocos. Logo o arquivo de 513 K tem 12+ 128+ 373 blocos de dados de 1K + 5 blocos de índice (1x indireto simples, 1x indireto duplo e seus 3 blocos). Ou seja, no total o arquivo ocupa 518 blocos no disco.

45: Resposta: a) ponteiro direto => 1 bloco; b) ponteiro direto => 1 bloco; c) ponteiro indireto simples: 1 bloco de índice + 1 bloco; d) ponteiro indireto duplo: 2 blocos de índice + 1 bloco.