Lista de SO

Professor: Tiago Alves e-mail: tiago@ime.uerj.br

Novembro 2019

1) Para cada afirmação abaixo diga se a mesma é verdadeira ou falsa e justifique.

(a) O uso de DMA sempre é vantajoso. Falso, o custo do estabelecimento de conexão para DMA pode não vale a pena quando o tamanho dos dados a serem transferidos entre dispositivo de I/O e Memória é muito pequeno.

(b) O espaço de endereçamento de memória é separado por usuários. Falso. É separado por processos.

(c) Um programa que faz polling (espera ocupada) pode ter desempenho superior ao equivalente usando interrupções. Verdadeiro, pois na espera ocupada o processo pode continuar sua execução assim que o dado está pronto, sem precisar de troca de contexto.

(d) Um processo no estado de espera tem uma requisição feita por chamada de sistema que ainda não foi atendida. Verdadeiro, o processo só entra no estado de Espera (ou bloqueado) quando faz uma requisição ao SO que o faz, então, aguardar uma interrupção.

2) Compare as arquiteturas de microkernel e kernel monolítico e liste vantagens/desvantagens de cada uma. Em uma arquitetura de microkernel boa das funcionalidades do SO são implementadas como componentes isolados fora do kernel, os chamados serviços. Uma chamada de sistema é feita através do estabelecimento de uma conexão entre o processo do usuário e o serviço correspondente. Em um kernel monolítico todas as funcionalidades correspondentes às chamadas de sistema são implementadas dentro do próprio kernel, sem isolamento. Dessa forma, as vantagens do microkernel seriam: (a) robustez, pelo fato dos serviços serem componentes isolados, o que faz com que um erro em um serviço não afete o SO como um todo; (b) Possibilidade de trocar um serviço por uma outra versão sem precisar que o buffer seja preenchido totalmente.

3) Em que situações é necessário invocar a rotina `fflush()`?

Quando o usuário deseja que a operação de I/O seja feita imediatmente, sem esperar que o buffer seja preenchido.

4) João, um programador não muito esperto, escreveu um programa que faz muitas leituras de posições aleatórias de um arquivo armazenado em memória. Certo dia, João trocou o arquivo original por um arquivo mil vezes maior, mas

manteve o número de leituras aleatórias constante. Ao fazer essa troca pelo arquivo maior, João notou significativa perda de desempenho. Como você justificaria essa perda de desempenho para o João, visto que o número de leituras em arquivo continuou o mesmo?

Como o tamanho do arquivo aumentou, o espaço amostral para a seleção aleatória das posições das n leituras a serem feitas aumenta. Logo, a probabilidade de ums leitura ser de uma posição do arquivo que já está na cache diminui, havendo, portanto, mais *cache misses*.

5) Dois processos A e B fazem apenas cálculos de ponto flutuante e apenas uma chamada de I/O no final para imprimir o resultado. Considerando a chamada de I/O insignificante em relação ao resto do programa, há vantagem na multiprogramação neste cenário (com relação ao tempo total de execução dos dois processos)?

Não, executar A e B sequencialmente pode até mesmo ter um desempenho melhor, por evitar trocas de contexto.

6) Em relação diagrama de processos de 5 estados (pronto, executando, bloqueado, novo e terminado) responda:

(a) Faria sentido adicionar uma transição do estado de pronto para o estado bloqueado? Justifique. Não, pois um processo só vai para o estado de bloqueado após invocar uma chamada de sistema que provoque com que ele seja desescalonado. Se tal processo está em Pronto, ele não está executando nada no momento.

(b) Em quais circunstâncias um processo pode sair do estado de executando e para qual estado ele é enviado em cada situação? Pode sair de de Executando para Terminado, ao ser finalizado, para Bloqueado quando estiver esperando alguma interrupção correspondente a um evento decorrente de uma chamada de sistema e para Pronto caso ocorra preempção.

7) Explique o conceito de multi-programação e exemplifique cenários para os quais ele foi desenvolvido.

8) O que é troca de contexto de processo? Como esse mecainsmo funciona?

9) Que mecanismo do sistema computacional garante que diversos processos CPU-bound de mesma prioridade obtenham o mesmo consumo da CPU?