Exame Preliminar de Doutorado em Sistemas Operacionais

Duração: 3 horas

17 de março de 2000

Atenção: Apesar da soma dos pontos das questões ser 14, tente responder corretamente o número máximo de questões, a fim de maximizar as suas chances de uma boa avaliação. A nota é o máximo das somas das notas de questões, tomado sobre os subconjuntos de questões cujo valor soma 10. Seja claro e preciso em suas respostas. Se quiser, use figuras para ilustrar suas respostas.
Boa sorte!

Questão 1 *(1 ponto)*

As tarefas executadas pelo kernel podem ser implementadas na forma de *co-rotinas* (como no Linux). Outra possibilidade é construir um kernel *multi-threaded* (Solaris). Discuta as vantagens e desvantagens de cada estratégia.

Questão 2 *(2 pontos)*

Esta questão é sobre a implementação de uma system call como a `fork` do Unix, em diferentes arquiteturas. Suponha que você tem um computador sem MMU e sem nenhum registrador que esteja sob controle exclusivo do sistema operacional e que desempenhe o papel de registrador base (para a tradução de endereços lógicos para físicos), ou de apontador para a tabela de páginas.

1. Caso um programa de usuário possa manipular qualquer registrador e tenha acesso irrestrito à memória, explique porque é difícil ou impossível (escolha um destes dois qualificativos) implementar uma system call como a `fork`. Justifique a sua escolha de qualificativo e exemplifique com problemas concretos.

2. Suponha que você tem uma máquina muito simples que não tem modos de endereçamento indireto, isto é que aceitem deslocamentos relativos ao conteúdo de registradores (por exemplo: `k + R4`, onde `k` é uma constante e `R4` é o valor atual do registrador $R4$). É possível implementar algo como `fork` nessa máquina? Se voce achar que for possível, diga quais convenções de codifição seriam necessárias, senão descreva as razões da impossibilidade.

Questão 3 *(1 ponto)*

Complete os procedimentos `entra_regiao(int p)` e `sai_regiao(int p)` para a implementação de exclusão mútua usando o algorítmo de Peterson (que funciona para 2 processos).

```
#define N 2
#define TRUE 1
#define FALSE 0
int turn;
int interested[N];

void enter_region(int process) {
...
}
void leave_region(int process) {
...
}
```

Questão 4 *(3 pontos)*

Seja um sistema simulando um restaurante com N mesas e com um garçom, que atende (uma vez só) cada cliente sentado em uma mesa. Enquanto o restaurante está vazio, o garçom fica descansando. Um novo cliente só entra no restaurante se alguma das mesas estiver vazia. Se isto for o caso, o cliente senta-se em uma mesa vaga qualquer (fornecida pela função `vacant_table()`), e chama o garçom. Assim que é atendido, o cliente faz a refeição (função `eat()`), depois se levanta (e sai do restaurante), liberando a sua mesa. Quando o garçom for chamado e estiver à toa, o mesmo atende o cliente imediatamente. Quando um cliente chama pelo garçom e este está atendendo um outro cliente, o chamado só é atendido após o atendimento dos clientes que já fizeram a chamada.

Para este sistema, descreva as variáveis compartilhadas (com seus valores iniciais) e o pseudo-código para os processos *client* e *waiter* usando semáforos.

Questão 5 *(1 ponto)*

De que maneira a escolha da estrutura de dados em um programa (para grandes quantidades de dados) bem como a escolha da linguagem de programação têm influência sobre a taxa de page-faults em sistemas com paginação por demanda (*demand-paging*).

Questão 6 *(2 pontos)*

A) Discuta qual é a influência que o tamanho da janela do working set tem sobre o grau de utilização da CPU e o grau de multiprogramação.

B) Explique o que é necessário (em termos de características do HW) para implementar uma aproximação do modelo de working set como política de liberação de frames.

Questão 7 *(2 pontos)*

Seja um disco de 10 cilindros (cilindros 0 a 9) no qual o braço de leitura leva uma unidade de tempo (p.ex. 1 milisegundo) para se deslocar de um cilindro para o próximo. Assuma que o tempo de latência e de leitura é desprezível quando comparado com o *seek time*, e que inicialmente ($t = 0$) o braço de leitura se encontra no cilindro 0. Qual é a ordem de atendimento e o número total de unidades de tempo necessários para atender a seguinte sequência de requisições para as políticas de atendimento:

- First Come First Serve (FCFS)
- Shortst seek time first (SSFT) e
- SCAN (Alg. do elevador)

onde t significa o instante de tempo da requisição. *Que conclusão voce tiraria deste exemplo?*

t (instante)	cilindro
0	5
3	2
4	6
9	9
10	7

Questão 8 *(1 ponto)*

Descreva as principais etapas que ocorrem durante um page-fault em um sistema com memória virtual e quais são as tarefas de cada uma das seguintes componentes do kernel envolvidas: *pager, escalonador, driver do swap disk, FLIH (tratador de interrupção de primeiro nível),* assumindo-se que não há nenhum frame livre na memória no momento do page-fault e que todas as páginas tenham o seu *modify bit* setado.

Questão 9 *(1 ponto)*

Dado um programa que efetua a leitura de um arquivo byte a byte, discuta por que é mais eficiente faze-lo através da chamadas à biblioteca `stdio` do que diretamente através da system call `read`.