# Exame Preliminar de Doutorado em Sistemas Operacionais

Duração: 3 horas

25 de agosto de 2000

**Atenção:**
Apesar da soma dos pontos das sete questões desta prova 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.

**Boa sorte!**

1. (Valor 1.0) Qual é o papel das variáveis de condição? Faz sentido inicializar variáveis de condição?

2. (Valor 1.0) Se um processo termina mas ainda existem *threads* deste processo rodando, elas continuarão rodando? Explique.

3. (Valor 2.0) Você foi contratado pela CET para sincronizar o tráfego sobre uma nova ponte construída na cidade. Devido a problemas orçamentários, a estrutura da ponte não é muito boa e seu uso envolve algumas restrições. O tráfego pode cruzar a ponte somente em uma direção a cada vez, e se houver mais de três carros sobre a ponte, ela desmorona. No seu trabalho, você deve representar cada carro por um processo que executa o procedimento *OneVehicle* quando chega na ponte:

```
OneVehicle(int direc)
{
  ArriveBridge(direc);   // chega na ponte
  CrossBridge(direc);    // cruza a ponte
  ExitBridge(direc);     // sai da ponte
}
```

   No código acima, *direc* é 0 or 1, e indica a direção em que o carro deseja cruzar a ponte.

   (a) Escreva as rotinas *ArriveBridge* e *ExitBridge* (*CrossBridge* deve somente imprimir alguma mensagem de debug), usando semáforos. *ArriveBridge* não deve retornar até que seja seguro para o carro cruzar a ponte na direção indicada (deve garantir que não haverá uma colisão e a ponte não colapsará). *ExitBridge* é usada para indicar que o "chamante" já terminou de cruzar a ponte; *ExitBridge* deve tomar os passos adicionais necessários para permitir que outros carros cruzem a ponte.

   Você não precisa se preocupar em garantir justiça nem evitar que um carro espere para sempre.

   (b) Na sua solução, se um carro chega quando o tráfego está se movendo na direção que ele deseja, mas há um outro carro já aguardando do outro lado da ponte que a direção se reverta, o que acontece : o novo carro irá cruzar antes do carro que já estava na espera, ou depois ... ou não é possível afirmar? Justifique sua resposta sucintamente.

4. (Valor 2.0) Verifique o seguinte estado de alocação dos recursos A (9 instâncias), B (11), C(8) e D (10) com relação a segurança:

| **Processo** | **Aloc** | | | | **Max.** | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | A | B | C | D |
| $P_1$ | 1 | 0 | 2 | 0 | 3 | 2 | 4 | 2 |
| $P_2$ | 0 | 3 | 1 | 2 | 3 | 5 | 1 | 2 |
| $P_3$ | 2 | 4 | 5 | 1 | 2 | 7 | 7 | 5 |
| $P_4$ | 3 | 0 | 0 | 6 | 5 | 5 | 0 | 8 |

Se for seguro, existe algum pedido que não pode ser atendido imediatamente? Se não for, o que deve ser feito para torná-lo seguro?

No algoritmo do banqueiro, há um passo em que se busca um processo $i$ (ainda não marcado) tal que a sua requisição de recursos sejam menos ou igual ao vetor que indica a disponibilidade corrente. Por que não é importante escolher tal processo $i$ que possa ser atendido de forma a consumir um número mínimo ou máximo de recursos ?

5. (Valor 1.0) Compare o tempo médio de execução e o tempo médio de término (*turnaround)* para os seguintes processos, de acorto com as políticas *FCFS* e *Shortest Job Next*:

| **Processo** | **Chegada** | **Processamento** |
|---|---|---|
| A | 0 | 3 |
| B | 1 | 5 |
| C | 3 | 2 |
| D | 9 | 5 |
| E | 12 | 5 |

6. (Valor 2.0) Quais destas definições de *working set* são mais adequadas para aproximar a localidade de referência? Comente cada uma do ponto de vista de efetividade e de implementação.

   (a) páginas referenciadas nos últimos $n$ segundos (tempo de parede).

   (b) pág. ref. nos últimos $n$ segundos de tempo virtual (CPU).

   (c) últimas $k$ páginas referenciadas pelo processo.

   (d) páginas onde o processo fez as últimas $r$ referências.

7. (Valor 1.5) Assumindo tamanho de página de 4 kbytes e cada entrada na tabela de página ocupando 4 bytes, quantos níveis de tabelas de páginas são necessários para mapear um espaço de endereçamento de 64 bits se a tabela de página de nível mais alto cabe em uma única página ?

8. (Valor 2.0) Considere as seguintes políticas de escalonamento de disco: **FCFS**, **SSTF**, **SCAN** e **C-SCAN**. Quais as condições ótimas para cada uma delas, em termos de padrão e volume de requisições? Em quais condições a latência passa a ser importante? Por quê?

9. (Valor 1.5) Por que se usa polling (*busy-waiting*) para fazer entrada e saída em alguns dispositivos? Quais as características que determinam esta escolha?