

UFES – Departamento de Informática
2ª. Prova de Sistemas Operacionais – 2019/2 – Profª. Roberta L. Gomes

Aluno: _____

- 1) **(2,0)** Na especificação do primeiro trabalho, foi solicitado que “durante a execução do tratador do sinal SIGINT, todos os demais sinais devem ser BLOQUEADOS”. Explique como seu grupo implementou esse bloqueio. Em que situações pode ser importante implementar tratadores de sinais dessa forma (i.e., com bloqueio de outros sinais)?
- 2) **(1,5)** (POSCOMP 2008 - *adaptação*) Considerando as diferenças existentes entre a execução de um algoritmo sequencial e a execução de um algoritmo distribuído (*multi-threaded*), analise as seguintes afirmativas com V (verdadeiro) ou F (falso), explicando sua resposta. **[RESPONDER NA FOLHA DE PROVA]**
- a. Somente na execução sequencial de um algoritmo existe a possibilidade de ocorrer um *deadlock*. (___)
- _____
- _____
- _____
- b. Um algoritmo sequencial apresenta mais de uma execução possível para uma dada entrada. (___)
- _____
- _____
- _____
- c. Um algoritmo distribuído exige necessariamente mais de um computador para ser executado. (___)
- _____
- _____
- _____
- d. A execução de um algoritmo distribuído pode ser não-determinística. (___)
- _____
- _____
- _____
- 3) **(2,0)** Sistemas operacionais antigos não ofereciam suporte a semáforos para a sincronização de processos/threads de usuário, mas eles ofereciam suporte a Pipes. Lembrando que Pipe é uma estrutura que implementa um *buffer* de memória de tamanho fixo (normalmente 4KB) usado para se transferir um *stream* de caracteres. Ler de um Pipe vazio bloqueia o processo/thread até que algum dado seja escrito no Pipe. Escrever em um Pipe cheio bloqueia o processo/thread até que algum dado seja lido.
- a) Explique como implementar um semáforo usando Pipe.
- b) Um semáforo implementado usando-se Pipe pode causar *deadlock* em uma situação que, normalmente, um semáforo convencional não causaria. Descreva essa situação.
- 4) **(3,0)** Considere o problema dos leitores e escritores, em que existem diversos processos que eventualmente fazem acessos de leitura a uma base de dados (“leitores”) e diversos processos que eventualmente fazem acessos de escrita à mesma base (“escritores”). Vários acessos de leitura podem ocorrer simultaneamente, mas um acesso de escrita não pode ocorrer simultaneamente com acessos de nenhum tipo. Você deve propor uma solução (em pseudo código) para esse problema usando semáforos que deve priorizar os processos leitores, isto é,

escretores só devem acessar a base de dados quando não houver mais leitores querendo acessar a base de dados. Responda usando o quadro abaixo **[RESPONDER NA FOLHA DE PROVA]**:

<pre>//Dados compartilhados //Código do processo escritor</pre>	<pre>// Código do processo Leitor</pre>
------------------------------------------------------------------	-----------------------------------------

- 5) (1,5) Vimos que em um monitor de Hoare, o processo sinalizador é interrompido indo para uma fila de sinalizadores (caso haja algum processo bloqueado na variável de condição na qual foi feito o **signal()**). Suponha agora que a chamada **signal()** a uma variável de condição somente possa aparecer como o último comando de um procedimento/método de entrada do Monitor (a mesma forçará um *return* do método). Explique qual o impacto dessa restrição sobre o comportamento do monitor.

Boa prova!!!