

UFES - DEPARTAMENTO DE INFORMÁTICA

2ª. Prova de Sistemas Operacionais - Período: 2015/1 - Profª. Roberta Lima Gomes

- 1) **(2,5)** No primeiro trabalho prático da disciplina, para cada linha de comando, deveria ser criado um processo “gerente”, e este por sua vez criaria n processos para executarem os comandos passados na linha. Sobre o tratamento de sinais, foi solicitado: “... *todos os processos criados pelo processo “gerente” devem ignorar o sinal SIGTSTP ...*”. Explique por que não seria possível implementar esta parte do trabalho se ao invés de “ignorar”, fosse solicitado que os processos criados pelo gerente “bloqueassem” o sinal SIGTSTP.
- 2) **(2,5)** Escreva um programa concorrente (em *C-like*) formado por três processos, um consumidor (c) e dois produtores (p0 e p1), que executam um loop infinito. O processo consumidor recebe informações (um valor inteiro) do processo p0 no buffer b0, e do processo p1 no buffer b1 (primeiro consome de b0 e depois consome de b1). Os buffers b0 e b1 possuem capacidade de armazenar um único elemento. O processo consumidor somente pode consumir se existirem informações no buffer, e os processos produtores somente podem voltar a produzir depois do processo consumidor haver retirado as informações do buffer. Utilize semáforos para sincronização.
- 3) **(2,5)** Sincronização com monitores usa variáveis de condição e duas operações especiais sobre elas (WAIT e SIGNAL). Uma forma mais geral de sincronização utilizaria uma única primitiva, como WAITUNTIL, tendo como parâmetro um predicado booleano qualquer. Por exemplo, poderíamos usar como:

WAITUNTIL ($x < 0$ AND $y+z < n$)

A primitiva SIGNAL não seria mais necessária. Este esquema é mais geral que o proposto por Hoare ou Hansen, e também mais fácil de ser utilizado. Por que então ele não é usado?

- 4) **(2,5)** Como são definidas e usadas áreas de memórias compartilhadas entre processos no UNIX? Explique quais são os passos necessários para que processos possam compartilhar memória.

Boa Prova !!!