

Lista 1

A idéia desta lista é exercitar manualmente as técnicas de programação e verificação de sistemas concorrentes.

Cada exercício deve ser entregue na semana seguinte à da apresentação do assunto em sala de aula

1) Simular as soluções apresentadas por Dijkstra e Peterson (dos slides) explorando vários cenários de execução (cenário(s) cuja seqüência “dá certo” e cenário(s) cuja seqüência leva ao problema – quando for o caso).

A simulação deve ser feita manualmente (como apresentada em sala de aula)

2) Simular as várias soluções do problema dos produtores e consumidores com buffer limitado, com semáforos, apresentadas em sala de aula manualmente e através de Redes de Petri.

a) simular manualmente como em sala de aula (vários cenários)

b) Para a solução completa apresentar (manualmente):

o grafo de lugares e transições em Redes de Petri

árvore de alcançabilidade deve ser feita manualmente para uma das soluções

d) Para todas as soluções fazer as simulações utilizando o JSARP (link na página) e comprovar aquela feita manualmente.

A simulação manual pode ser feita com pelo menos 1 produtor e 1 consumidor, mas o ideal seria fazê-lo com 2 processos de cada tipo. A simulação com o JSAP deve ser feita com dois processos de cada.

4) Propor e simular apenas com Redes de Petri (na ferramenta) a solução para os produtores e consumidores utilizando 2 semáforos para garantir a sincronização (*item e vaga*), mas aumentando a concorrência entre produtores e consumidores (por exemplo, permitindo o acesso concorrente de 1 produtor e 1 consumidor ao buffer).

5) Verificar o que acontece se a ordem do acesso aos semáforos for invertida (nos produtor e/ou no consumidor).

Ex.:

```
/* Produtor: */  
...  
wait (mutex);  
wait (item);  
<acesso ao buffer>  
signal (item);  
signal (mutex);  
...
```