

Exercícios Complementares — Comunicação Interprocessos

1. Suponha que, em um sistema, existem diversas aplicações que, em determinados pontos, consomem uma quantidade significativa de memória. Para evitar a exaustão da memória física do sistema, definiu-se que essas aplicações devem ser escritas de modo que apenas uma delas pode executar suas operações que demandam memória de cada vez. Em outras palavras, é necessário garantir a **exclusão mútua** no uso da memória pelas aplicações. Mostre como as aplicações poderiam ser construídas usando semáforos. Sua solução deve eliminar espera ocupada.

Dica: a aplicação pode ser modelada da seguinte forma:

```
void aplicacao() {  
    while (TRUE) {  
        faz_algo();  
        come_memoria();  
    }  
}
```

2. Como no caso anterior, mas agora as aplicações precisam de CPU e memória, e só podem executar quando tiverem ambos os recursos.
3. Suponha que, numa situação semelhante às anteriores, as aplicações podem necessitar de uso exclusivo de dois dos três recursos a seguir: CPU, disco e memória. Um gerente de recursos do SO escolhe, ao acaso, dois desses recursos, e libera a execução de uma aplicação que precise deles. Ao fim do uso, a aplicação devolve os recursos ao gerenciador. Escreva o pseudocódigo das aplicações e do gerenciador usando semáforos. Sua solução deve garantir que uma aplicação só execute quando tiver ambos os recursos necessários, e que ela não retenha um recurso quando o outro recurso do qual ela precisa esteja indisponível.
4. Considere um programa concorrente com três *threads*, X, Y e Z, mostradas abaixo.

<code>int n = 1; /* variável compartilhada entre X, Y e Z */</code>		
<code>void X() { n = n * 16; }</code>	<code>void Y() { n = n / 7; }</code>	<code>void Z() { n = n + 40; }</code>

Mostre como as *threads* poderiam usar semáforos para garantir que o resultado final de *n* seja sempre 8.

5. O Joinville Esporte Clube encomendou um sistema para venda de ingressos para seus jogos através de quiosques eletrônicos. As partidas são realizadas na Arena Joinville, que tem capacidade para MAX torcedores; os lugares são numerados sequencialmente, de 1 a MAX. O sistema deve funcionar da seguinte forma: o cliente escolhe o lugar que deseja (dentre os disponíveis) e efetua o pagamento. O pagamento é feito por cartão, e pode ser aprovado pela operadora ou não (neste caso, o cliente perde o direito ao lugar escolhido, que volta a ficar disponível). O sistema não pode vender duas vezes o mesmo lugar, nem pode marcar como ocupados lugares escolhidos por clientes que não conseguem efetuar o pagamento. Apenas um lugar é vendido por vez. Escreva o pseudocódigo das aplicações que executam nos quiosques, usando semáforos para evitar condições de disputa. Considere que a autorização de pagamento é feita por linha discada, estando sujeita a falhas de conexão e a longos tempos de espera (ou seja, um cliente que aguarda a autorização de pagamento não pode prejudicar a venda de ingressos em outros quiosques).