

## Deadlock

1) Liste três exemplos de deadlocks não relacionados a um ambiente de computação.

**Um cruzamento de trânsito.**

**Uma linha de produção em que uma etapa depende da outra e a última da primeira.**

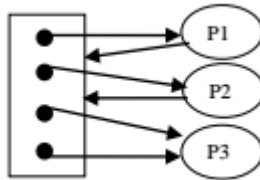
**Três programadores que não se comunicam muito bem, e para finalizar seu trabalho um necessita do trabalho do outro.**

2) É possível haver um deadlock envolvendo apenas um processo? Explique sua resposta.

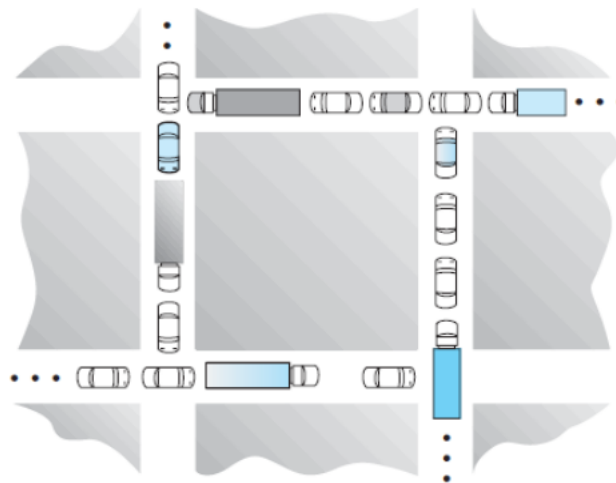
**Não, já que para haver um deadlock deve existir um conjunto de processos em espera de forma circular, um processo deve estar de posse de um recurso e aguardando um recurso que está em posse de outro processo, dessa forma, apenas com um processo não é possível haver deadlock.**

3) Considere um sistema composto de quatro recursos do mesmo tipo compartilhados por três processos, cada um deles necessitando de no máximo dois recursos. Mostre que o sistema é livre de deadlock.

**Pelo grafo de alocação de recursos, podemos perceber que não há um ciclo e, portanto, não tem como haver deadlock.**



4) Considere o deadlock de trânsito indicado na figura abaixo:



a) Mostre que as quatro condições para o deadlock de fato estão presentes nesse exemplo.

**Exclusão mútua:** cada rua está alocada para apenas um fluxo de carros;

**Posse e espera:** Os carros no fluxo baixo-cima estão aguardando a liberação da esquina pelo fluxo direita-esquerda.

**Não preempção:** Apenas os carros que estão naquele fluxo podem libera a esquina que ocupam;

**Espera circular:** O fluxo baixo-cima aguarda a liberação da esquina do fluxo direita-esquerda, que aguarda a liberação da esquina do fluxo cima-baixo, que aguarda a liberação da esquina do fluxo; esquerda-direita que por sua vez aguarda a liberação da esquina do fluxo baixo-cima. Tendo assim uma espera circular.

b) Apresente uma regra simples que evite deadlock nesse sistema.

**Uma regra simples seria a de posse e espera. Dessa forma, não permitiria que os fluxos peçam recursos aos poucos, ou seja, cada fluxo pede todas as esquinas necessárias de uma vez ou liberam todas as que detêm antes de pedir outras.**