**Discussões - Aula06**

**Disciplina:** Sistemas Operacionais **Professor:** Cléver Ricardo Guareis de Farias

**Nomes:**

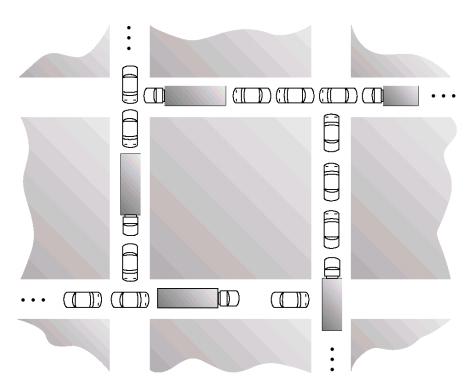
Felipe Marcelo **nºUSP: 10684743**

Felipe Limão **nºUSP: 10797843**

Gabriel Modina **nºUSP: 10684771**

Douglas Feltrim **nºUSP: 9001585**

**Discussão 1 -** Considere o cenário ilustrado abaixo:



**a)** mostre que as quatro condições necessárias para o deadlock são mantidas de fato neste exemplo.

**b)** indique uma regra simples para evitar deadlocks neste caso.

**R: a)** Chamemos os cruzamentos de recursos R1, R2, R3, R4. A primeira condição é satisfeita, pois assim que um veículo qualquer tem a posse do recurso, esse não pode ser utilizado por outro veículo ao mesmo tempo, logo a exclusão mútua se faz presente. A segunda condição também é satisfeita. Tendo posse do recurso R1, uma fila de veículos espera pela liberação de um recurso R2 para prosseguir sua execução, logo essa fila mantém a posse de um recurso e espera pela liberação de outro. A terceira condição também é satisfeita, pois estando um veículo em posse do recurso R1, outro veículo não pode utilizar R1 enquanto o primeiro veículo não terminar a sua utilização, logo a não-preempção se faz presente. Por fim, temos que existem veículos em posse de R1, esperando R2 ser liberado, o qual está em posse de outros veículos, que aguardam pela liberação de R3, que também está em posse de outros veículos que aguardam a liberação de recursos de R4, que por sua vez aguardam a liberação de R1, ou seja, a espera circular também se faz presente.

**Discussão 2 -**Considere as matrizes de alocação, requisição e recursos disponíveis abaixo. Verifique se há algum processo em deadlock.

**R:**

**Requisição**  **Disponível**

**P1:** [0 0 0] <= [0 0 0]: P1 libera recursos.

**>** **A** = A[0 0 0] + C(0, j)[0 1 0] = [0 1 0]

**P2:** [2 0 2] !<= [0 1 0]

**P3:** [0 0 0] <= [0 1 0]: P3 libera recursos.

**> A** = A[0 1 0] + C(2, j)[3 0 1] = [3 1 3]

**P4:** [1 0 0] <= [3 1 3]: P4 libera recursos.

**> A** = A[3 1 3] + C(3, j)[2 1 1] = [5 2 4]

**P5:** [0 0 2] <= [5 2 4]: P5 libera recursos.

**> A** = A[5 2 4] + C(4, j)[0 0 2] = [5 2 6]

**P2:** [2 0 2] !<= [5 2 6]: P2 libera recursos.

**Não há processos em deadlock!**

**Discussão 3 -**Considere as matrizes de alocação, requisição e recursos disponíveis abaixo. Verifique se há algum processo em deadlock.

**R:**

**Requisição**  **Disponível**

**P1:** [3 1 1 1] !<= [1 1 0 0]

**P2:** [2 3 1 0] !<= [1 1 0 0]

**P3:** [1 0 0 2] !<= [1 1 0 0]

**P4:** [1 0 0 0] <= [1 1 0 0]: P4 libera recursos.

**> A** = A[1 1 0 0] + C(3, j)[0 1 0 2] = [1 2 0 2]

**P1:** [3 1 1 1] !<= [1 2 0 2]

**P2:** [2 3 1 0] !<= [1 2 0 2]

**P3:** [1 0 0 2] !<= [1 2 0 2]: P3 libera recursos.

**> A** = A[1 2 0 2] + C(2, j)[1 0 2 0] = [2 2 2 2]

**P1:** [3 1 1 1] !<= [2 2 2 2]

**P2:** [2 3 1 0] !<= [2 2 2 2]

**Os processos 1 e 2 estão em deadlock!**