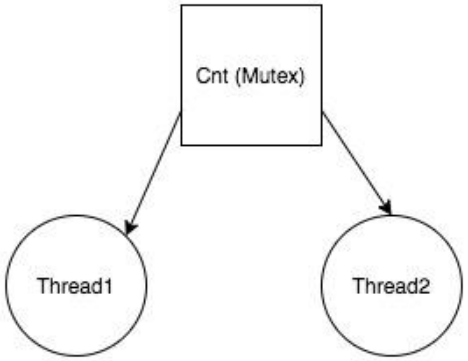
Q1) Deadlock is not possible in modified program “igoodcnt.c”. There is only one semaphore, and although it can be arbitrary to determine which one will gain access to the semaphore first. There will not be a situation where thread 1 is waiting on access to a resource held by thread 2 and vice versa, which would create deadlock and the counter would not increment. As can be seen in the resource allocation graph below, the same resource (or semaphore) is accessed by both threads but once at a time. This will ensure that the threads will increment only when the cnt resource is not currently being used, and there is no deadlock since the both threads release the resource control when they are done incrementing for the other to use.



Q2) a) Need Matrix:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 6 & 4 & 2 \end{pmatrix}$$

b) This can be determined by comparing the available, 1,5,2,0, and since it less than or equal to P0 and as well as P3 in the need matrix, these can be executed. After these two processes have ran and released their allocation, the available becomes 1,11,6,4 and which point any of the processes can run in any arbitrary order. A possible safe sequence is P0, P3, P1, P2, P4. The system is a safe system.