Vivian Do
CSC 4320
Feb 26 2023

Spring 2023 – 4320/6320 Section 006 Operating Systems

Homework 3: Due 02/26/2022 at 11:59 PM

Your programs – if requested – must compile with gcc and execute on snowball.cs.gsu.edu! Please see https://cscit.cs.gsu.edu/sp/guide/snowball for more details. You may use whatever IDEs / text editors you like, but you must submit your responses on iCollege.

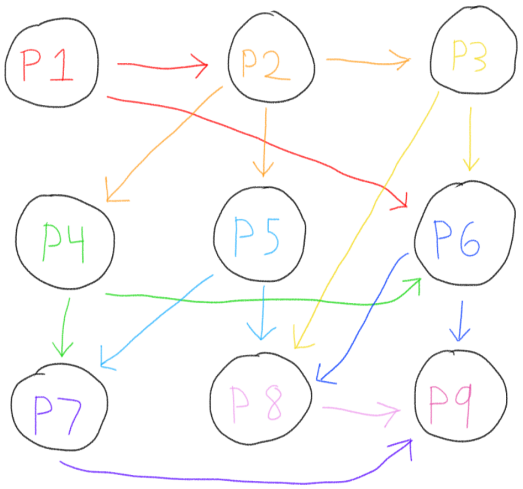
1. Consider a system of 9 processes, **P** = {p1, …, p9}.
   Associated with the system are 6 memory cells, **M** = {M1, …, M6}.
   The domain and range for each process is given in the following table:

Process pi	Domain D(pi)	Range R(pi)
p1	M1, M2	M3
p2	M1	M5
p3	M3, M4	M1
p4	M3, M4	M5
p5	M3	M4
p6	M4	M4
p7	M5	M5
p8	M3, M4	M2
p9	M5, M6	M6

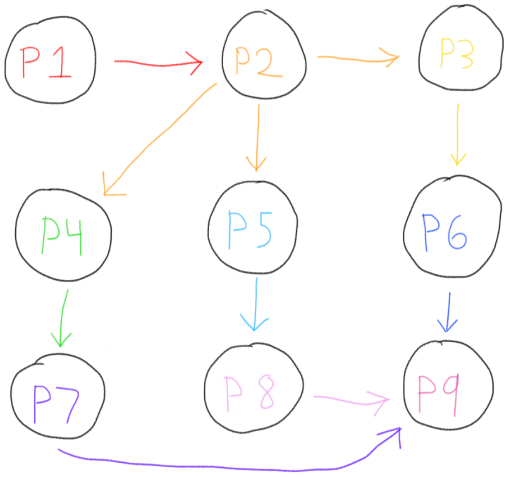
In addition, you are given the following precedence relation:
 = {(P1,P2), (P1,P6), (P2,P3), (P2,P4), (P2,P5),(P3,P6),(P3,P8),(P4,P6),
(P4,P7), (P5,P7), (P5,P8), (P6,P8), (P6,P9), (P7,P9), (P8,P9)}

a. Construct the Precedence Graph (not containing any redundant edges; also modify accordingly). Use PowerPoint, diagrams.net, or any other app to draw the graph. (15 points)

My Precedence Graph:



Without Redundant Edges:



Redundancies: (P1, P6), (P3, P8), (P4, P6), (P5, P7), (P6, P8)

b. Is the system above determinate for all interpretations of its processes? If it is not, add to necessary elements to make it determinate (no graph drawing needed). Explain all of your reasoning. (20 points)

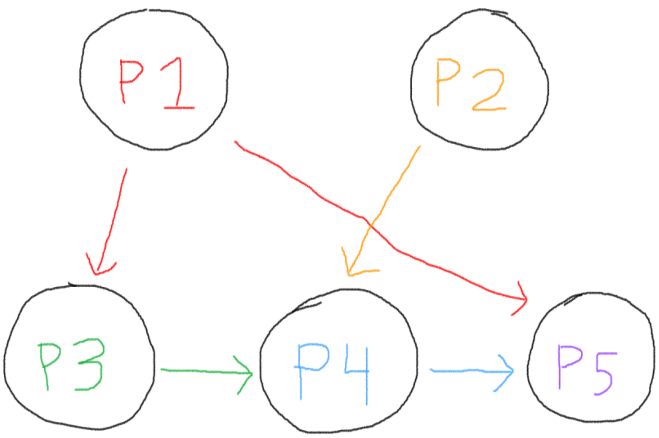
No, the system above is not determinate for all interpretations of its processes.
P1 has *one* necessary element from P2.
P2 has *one* necessary element from P1.
P3 has *two* necessary elements from P2 and P4.
P4 has *one* necessary element from P3.
P5 has *two* necessary elements from P1 and P2.
P6 has *one* necessary element from P5.
P7 has *two* necessary elements from P3 and P4.
P8 has *one* necessary element from P7.
P9 has *two* necessary elements from P6 and P8.

2. In the first problem, there were 9 processes, many of which were listed as pairs under the precedence relation. Suppose we are now dealing with a system of only 5 processes named P1 through P5. You are given a set of constraints that are expressed by the following precedence relation:

 = {(P1,P3), (P1, P5), (P2,P4), (P3, P4), (P4, P5)}

Provide pseudocode for each of those 5 processes to show how you can use semaphores to enforce these constraints (i.e., the precedence relation). Also, you must initialize these semaphores correctly (15 points).

Graph for Visualization:



Pseudocode:

assuming that we have 5 binary semaphores from 5 processes:
s1, s2, s3, s4, s5

start binary semaphore
s1 = 1; s2 = 1; s3 = 0; s4 = 0; s5 = 0s6 = 0; s7 = 0

start processes
P(s1) P1 V(s3) V (s5) // s1 starts P1, processes to P3(s3) and P5(s5)
P(s2) P2 V(s4) // s2 starts P2, processes to P4(s4)
P(s3) P3 V(s6) // s3 starts P3, processes to P4(s6) after P2
P(s4) P(s6) P4 V (s7) // s4 starts P4, processes s6 to P5(s7)
P(s5) P(s7) P5 // s5 starts P5, processes s7
end process

end binary semaphore

// P(s6) = up on semaphore s6
// V(s3) = down on semaphore s3
// P(1) = success; P(0) = fail