# CSCE-4600 Operating Systems Concepts
## Homework #2
## Due 06-19-2019

1.  (25pts) Prove that *"A mutually non-interfering system of processes is determinate"*

2.  (25pts) Consider a system of 9 processes, $P = \{p1, \ldots, p9\}$ 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:
    $\square$ = {(1,2),(1,6),(2,3),(2,4),(2,5),(3,6),(3,8),(4,6),(4,7),(5,7),(5,8),(6,8),(6,9),(7,9),(8,9)}

    a.  (10pts) Construct the Precedence Graph (not containing redundant edges) from the given precedence relation.
    b.  (5pts) As per the given precedence relation, which processes can be executed concurrently.
    c.  (5pts) Make the necessary modifications to guarantee that the system is determinate for all interpretations fp.
    d.  (5pts) Modify the precedence relation above, to construct a maximally parallel system.

3.  (50pts) Write a simple sequence-number system through which three concurrent processes, P1, P2, and P3, **each** obtain unique integers in the range [1, 500]. Use the fork() call to create P, P2, and P3. Given a file, F, containing a single number, each process must perform the following steps:

    a.  Open F.
    b.  Read the sequence number N from the file.
    c.  Close F.
    d.  Output N and the process' PID (either on screen or test file).
    e.  Increment N by 1
    f.  Open F.
    g.  Write N to F.
    h.  Flush F.
    i.  Close F.

Describe the behavior of your program and explore the reason for this behavior. Provide evidence for your conclusion in form of test-output. You must clearly document your code.

4. (25pts) There are 3 employees in a company who has 100 jobs to complete with 2 computers given to them. You have to design a non-blocking work schedule for them with no work-interference that can assure best possible assignment of jobs for each of the employees. You will need to make sure that all three employees get equal amount of jobs assigned in the schedule. Write a program depicting this scenario with critical section component as a consideration to your solution.

[Hint: the critical section here refers to the two computers available, you will need to consider 100 jobs as numbers and assign to Employee1, Employee2 and Employee3 with work proportion necessary and lock the access to the computers when occupied, later release them]

NOTE: All programs must compile and execute on the CSE machines. It is imperative that the sequence number files is located on the local disk. On Linux, the \tmp directory is located on the local file system.