Homework 2, Total points = 60

Assigned 9/17/09 due 9/24/09

Please submit typewritten solutions through ICON, preferably in the pdf format. Late assignments will not be accepted without prior approval.

Question 1 (10 points): A round-robin scheduler guarantees that between two consecutive actions by the same process, every other process with enabled guards executes their action at least once.

Consider a completely connected network of \( n \) processes numbered 0 through \( n-1 \). Assuming that only one process is scheduled at any time, implement a round-robin scheduler. Provide brief arguments in support of your implementation.

Question 2 (15 points): Three members of a library, Alice, Bob and Carol, are trying to share single copies of three books \( A, B, C \) from the library. Alice periodically wants both \( A \) and \( B \), Bob periodically wants both \( B \) and \( C \), Carol periodically wants both \( C \) and \( A \). None of them is required to return the books until they acquire both of their choices and use them for a bounded period of time. The librarian will always issue a book if it is available, but no wait-list is maintained.

(a) Assuming that a member can ask for one book at a time, write the program for the three members so that every member eventually receives their preferred books.

(b) If a member is allowed to ask for multiple books, and they check out their choices only if both are available, then are they guaranteed to receive their requested books?

Does fairness play any role in the correctness of your solutions in the above two cases?

Question 3 (15 points): Consider a system of unbounded clocks ticking at the same rate, and displaying the same value. Due to electrical disturbances, the phases of these clocks might occasionally be perturbed. The following program claims to synchronize their phases in a bounded number of steps following a perturbation:

{program for clock \( i \)}

\[ \text{define } c[i] : \text{integer} \ (\text{non-negative integer representing value of clock } i) \]

\( (N(i) \text{ denotes the set of neighbors of clock } i) \)
\[
\text{do } \; \text{true } \Rightarrow c[i] := 1 + \max \{c[j] : j \in N(i) \cup i\} \; \text{od}
\]

Assume a synchronous model where all clocks execute the above action with each tick, and this action requires zero time to complete, verify if the claim is correct. Prove using a well-founded set and an appropriate variant function that the clocks will be synchronized in a bounded number of steps. Also, what is the round complexity of the algorithm?

**Question 4** (10 points): There are two processes \(P\) and \(Q\). \(P\) has a set of integers \(A\), and \(Q\) has another set of integers \(B\). Using the message passing model, develop a program by which processes \(P\) and \(Q\) exchange integers, so that eventually every element of \(A\) is greater than every element of \(B\). Present a correctness proof of your program.

**Question 5** (10 points): In Ricart and Agrawala's distributed mutual exclusion algorithm (Chapter 6), the timestamp increases in an unbounded manner. Unfortunately, unbounded timestamps cannot be handled using finite resources. Is there a problem if one uses bounded timestamps (i.e. timestamps are incremented modulo \(M\) for some value of \(M\))? Explain your answer using a two-process scenario.