CSE 763 **Spring 2007**
Prof. Paul Sivilotti

Assignment #4: Time

DUE: in class, Friday Apr 27th.

---

1. **Vector Clocks** (15 points)

   Consider the implementation of vector clocks given in the notes:

   Program $VectorClock$ j
   var j, k : processes,
   $ch.j.k$: channel from j to k ,
   m : message,
   A : event,
   $vclock.j$: vector time of j ,
   $vtime.A$: vector time of A ,
   initially $vclock.j.j = 1$
   $\wedge$ $(\forall k : k \neq j : vclock.j.k = 0)$
   assign
   local event $A \longrightarrow$ $vclock.j.j := clock.j.j + 1$
   ; $vtime.A := vclock.j$
   $[]$ send event $A \longrightarrow$ $vclock.j.j := vclock.j.j + 1$
   (to k) ; $vtime.A, vtime.m := vclock.j, vclock.j$
   ; $ch.j.k := ch.j.k \mid m$
   $[]$ rcv event $A \longrightarrow$ $vclock.j := \mathbf{max}\,(vtime.m, vclock.j)$
   (m from k) ; $vclock.j.j := vclock.j.j + 1$
   ; $vtime.A, ch.j.k := vclock.j, tail(ch.j.k)$

   Prove the following invariants:

   (a) $(\forall j, k :: vclock.k.j \leq vclock.j.j)$

   (b) $(\forall A_j, B_k :: vtime.A_j < vtime.B_k \Rightarrow A_j \longrightarrow B_k)$ (where A_j denotes event A that occurred at process j).

2. **Gossip** (10 points)

   Complete the proof of progress for the Gossip algorithm. Use the metric given in class. The first half of the proof is already given in the lecture notes. (There is no need to repeat this part, but you should understand it in order to write the second part.)