CS 603 (Park) **Assignment IIII** **Due: Feb. 21 (Fri.), 1997**

PROBLEM 1

Complete reading Chapt. 3.

PROBLEM 2

As a continuation of Problem 3 (b), Assignment III, implement your version of Cristian's algorithm. To estimate the network delay, compute the round trip time (RTT) and divide it by 2. Test your algorithm using the same parameter set-up as used in Problem 3 (a). Compare the performance results and give your evaluation of the effect of using the clock synchronization mechanism.

PROBLEM 3

Implement the distributed mutual exclusion algorithm discussed in class which uses Lamport's logical clock. I.e., the first task it to create the logical clock which satisfies the causality properties—increment the clock between every two successive events and ensure that the receive time stamp of a message is always bigger than its send time stamp. Test the algorithm using 5 Xinu machines where 5 hosts run drivers that execute the following loop 3 times:

```
P-dist();
print time of day;
sleep for 3 secs;
print time of day;
V-dist();
```

`P-dist()` and `V-dist()` are the library calls for getting the semaphore and releasing it, respectively. Note, `P-dist()` is a *blocking* call. From your screen-dump of time stamps, verify that mutual exclusion is indeed being achieved.

PROBLEM 4

Consider the *bully algorithm* for leader election discussed in class. It is clear that it works in the presence of single faults, i.e., host failures. Does it work, without any further amendments, in the case of two host failures? Approach this question by considering all possible scenarios involving two faults (sequentially in time, both "well-separated" and "close together" as well as concurrent in time) and verifying whether the system eventually settles down into a *consistent* state, i.e., a state where every group member has the same view of the group leader and this stays invariant forever.