Larry Bush
Student ID: 660 220 742 email: Lawrence_Bush@dps.state.ny.us

March 26, 2002
Professor: Costas Busch

Final Project Proposal
Distributed Algorithms and Systems
Lock-Free Linked List Implementation

1. General Description

For my project, I would like to implement a Lock-Free Linked List. This will entail the creation of new data structures and algorithms, the simulation of the compare&swap syncronization primitive and writing a multi-threaded simulation and test program. The data structures will be templatized so that they can store any data type. I will be following the method that John Valois explains in his paper "Lock-Free Linked Lists Using Compare&Swap" and in his Thesis "Lock-Free Data Structures" with some modifications.

2. What are Lock-Free Linked Lists?

A Lock-Free Linked List is a data structure that implements concurrent objects without using mutual exclusion. Lock-Free Linked Lists allow traversal, insertion and deletion operations by different processes to occur at the same time without corrupting the data struture.

This is accomplished by using auxiliary nodes, the Compare&Swap primitive to swing pointers and careful manipulation and checking of the data structure operations.

3. Platform

a. C++
b. Windows 2000

4. This implementation will include

a. Data Structures
   i. Node
      - The list nodes will be templatized so that they can store any data type.
   ii. List
   iii. Auxiliary Node

b. Algorithms
   i. Traverse (forward).
   ii. Insert
   iii. Delete

c. Sycronozation primitive
   i. Compare&Swap
      - This will be simulated using a higher level operation.

d. Simulation and Test Program
   - Multiple User level threads (I will start with 6) will do simultaneous insertions, deletions and traversals.

5. Output, Results and Reporting

After completing the coding and testing, I will write a 5 page report explaining my implementation, reporting my test results and an assesment of the results.

6. Individual Project – (No Partners)