1. Names and Communications

Construct an algorithm for reaching agreement among unreliable processes in terms of associative broadcast communications. It is acceptable to use the algorithm given in lecture and in the text as the basic algorithm and reformulate the communications this algorithm in associative broadcast.

Let the set of processes be $\{P_i\}$
Let : n = number of processes
m = upper bound on number of unreliable processes
Let : V_k be the element of common state held for P_k.

Desired : An algorithm to construct a vector

$X_i = (A_{i,1} - - - - A_{in})$ in each P_i such that
(1) if P_j is valid then $A_{i,j} = V_j$ in X_i.
(2) if P_i and P_j are both valid then $X_i = X_j$.

a) Define the profiles for the processes

b) Define the messages (selector and data) which are used.

c) Define the algorithms which are executed upon receipt of each message type.

A correct algorithm can only be devised if $n \geq 3 \times m + 1$

2. Processes and Threads - You are the designer for a threads package for a shared address space multiprocessor with non-uniform memory access. The memory which makes up the shared address space is actually physically local memory for each processor in the multiprocessor. The processor local segment of the shared address space has an average access time of 4 clock cycleswhile non-local references incur an average 10 clock cycle access time. This multiprocessor will be used for running parallel programs where the programs are composed of sequential tasks which interact with other tasks only during initiation and termination. (Each task is atomic by design.)

Define the characteristics of a threads package which will allow users to take advantage of the atomicity property of the tasks and execute efficiently in the context of the non-uniform memory access times by first given general characteristics including whether there are user threads and kernel threads separately and if so the mapping between them and the model of communication and syhnchronization. Then define the syntax and semantics of a subset of the interface to the threads package. In particular specify:

a) the thread_create operation
b) the mapping (if any) among user and kernel threads
c) operations for synchronization among threads

d) the thread_terminate operation

The definitions of operations should include the arguements and the pre-and post-conditions for the execution of each operation.