

Assignment #5: Resource Allocation

DUE: Wednesday, May 16th.

1. **Mutual Exclusion** (15 points)

Following the pattern and structure of the algorithm given for the token ring, write an implementation of the Raymond's token-based mutual exclusion scheme for tree topologies.

Prove that your program meets its specification.

2. **Dining Philosophers with Hierarchical Resource Allocation** (24 points)

In the dining philosophers problem, each process is associated with a set of resources. A resource can be held by at most one process at a time, and a process must hold all of its resources in order to enter its critical section.

Consider the following algorithm for dining philosophers. The resources are totally ordered and each resource has its own queue of pending requests (initially empty). Hungry philosophers enqueue themselves with their associated resources, respecting the following discipline:

- A philosopher is never in the *body* of more than one queue. (The body of a queue is the set of elements in the queue minus the queue's head.)
- A philosopher is never in the queue of an associated resource without also being in the queue of all of its *lower* associated resources.

A hungry philosopher eats when it is at the head of all of the queues of its associated resources and, after eating, it removes itself from all of these queues.

Part I. Write a UNITY-style program for a philosopher following this algorithm.

Part II. Is this program correct? If so, give a proof of correctness. If not, fix the algorithm (with as minor a modification as possible) and prove that your modified algorithm is correct.