Homework 2
Due: March 18, 2003

Problem 2.1
The purpose of this exercise is to understand in more detail the analysis of algorithm W. In the non-optimized algorithm W, the bottom-up processor enumeration phase uses a tree with P leaves, where P is the initial number of processors, and P can be as large as N, the size of the input. Each iteration of the algorithm computes monotonically tighter (over) estimate of the number of remaining processors, but the enumeration phases in subsequent iterations of the main loop do not take advantage of this fact.

1. If R is the estimate of the number of active processors (where R ≤ P) that is available prior to the start of the enumeration phase, revise the enumeration phase to use a processor counting tree so that its size depends on R. (If padding is necessary, be explicit about it.) Analyze the (worst case) time and work complexities of the revised enumeration phase.

2. Integrate this analysis within the overall analysis of the work for algorithm W leading to a revised Theorem 3.3.5. You must consider each of the Lemmas 3.3.1 - 3.3.4 and decide whether or not they need to be revised. If no changes are required, explain why. If changes are required, document the specific changes in the analysis. (You do not need to rewrite the steps in the analysis that are not changed.)

3. What is the resulting work complexity? Discuss theoretical and practical implications of your revised algorithm and analysis.

Problem 2.2
Analyze the work and time complexities of algorithm W for all fail-stop failure patterns F, for which |F| < cP, where c is a constant such that 0 < c < 1. Such an adversary is called linearly bound. Note that in the presence of this adversary at least a linear number of processors is non-faulty. This makes it possible to analyze the time complexity of the algorithm. Hint: Consider how time is used in assessing work.

Problem 2.3
Develop a fault-tolerant algorithm for summation in the fail-stop (crash) model. Given the input x[1..N], the algorithm has to compute Σ1≤i≤N x[i]. Use algorithm W as the basis for your solution. Show correctness of your solution and analyze its work. Can your algorithm be made optimal using the parameterized algorithm W as the basis? What is its range of optimality?