

“Self-Healing” Programs

Objective:

To understand how computer programs can recover from faults. You will play the role of a network of computers and will simulate a program that can heal itself.



Introductory Remarks:

Sometimes, faults are inevitable in a computer system. Examples of faults are programming errors, hard disk crashes, network failures, and power failures. When these faults occur, programs can be designed to heal themselves! These programs are called “fault tolerant”.

For example, on the space shuttle, fault tolerance is used to guarantee that the computers calculate correct values needed for flight control. In this case, several computers are used for critical operations. That way, the accuracy of each machine is checked by the other computers. If three computers are used, it is called triple modular redundancy.



Materials:

- 14 sets of binary (numbers 0,1) “number” cards
- 14 sets of “number” cards, numbered 0,1,...,13
- 14 hand chimes, of appropriate tones to form a melody
- 14 participants to be “processors”, arranged in a circle, facing out and each holding one set of binary cards, one set of number cards and a hand chime
- 1 volunteer to be the “demon”, capable of changing anyone’s number cards

Activity:

A) *Fault-Tolerance in sequential programs: “Triple Modular Redundancy”*

1. Each person is given the same math problem.
2. Each person holds up their answer using the number cards.
3. The demon can change at most 3 cards.
4. Can the rest of the group figure out what the right answer is?



B) *Fault Tolerance in distributed programs: “Token Rings”*

1. Begin with everyone’s number equal to 0
2. Run mode for *Basic Program: Token Ring*
 - All of the processors, except one, do the following, continuously:
If my left neighbor’s number is ever DIFFERENT than mine:
Play my note and change my number to MATCH that neighbor’s
 - The remaining processor does the following, continuously:
If my left neighbor’s number is ever the SAME as mine:
Play my note and change my number to be ONE BIGGER
3. While the program runs, the demon is allowed to change at most 3 cards
4. Run Basic Program in *Brittle Version: “Binary Token Ring”*: Use cards numbered 0,1

5. Run Basic Program in *Fault-Tolerant Version: "k-State Token Ring"*: Use cards numbered $0, 1, \dots, 13$

Challenges:

1. Let everyone pick a starting number in the brittle version. What happens?
2. Let everyone pick a starting number Fault-Tolerant Version. What happens?