CS 6/73201 **Project #2** **Advanced OS**

Due via email by 11:59pm on Sunday 22 April 2001

The Problem

In this assignment, you are to (i) use MPI to implement **Garcia-Molina's Bully Algorithm for Elections** (described at the end of Lecture 13), and (ii) come up with appropriate **test code or methods** to demonstrate that your algorithm works for at least 3 machines. Your algorithm should be designed for an arbitrary number of machines, but since we only have MPI on aegis, intrepid, and trident, you only need to test it on 3 machines. Take part (ii) seriously — think about how you can have these machines run, "fail", notice that others have failed, etc., and then make sure that your implementation and test cases demonstrate this effectively. For simplicity, just have the "failed" process send back an "I've crashed" message, instead of trying something more complex.

This project will comprise 15% of your final course grade.

What to Turn In

(30 points) In a file called **p2.overview**, write a brief overview of your project. Describe what you implemented, any design decisions that you made, and why you made those decisions. Describe your test cases, and how they demonstrate that the project work. Since the TA will have a large number of different projects to grade, supply instructions to the TA as to how he should test your project to demonstrate that it works. Do a good job on this writeup — convince the TA and me that you implemented something interesting, that you learned something from doing it, and that you actually had some tough choices to make in your implementation. A superficial 10-line writeup will definitely not be worth 30 points!

(70 points) The implementation of your project. If it doesn't quite work as desired, clearly describe in the file **p2.overview** what you have done, what is not working, and how you would go about finishing the project if you had more time. The better you describe the status, the better your chances will be for getting partial credit for what you've done. Also, if you have something that works, and you can document that status with test cases, etc., you'll probably get more credit than if you have a lot of code written, but nothing that compiles (since in the latter situation it's harder for the TA to evaluate the status of your work).

Where to Get Help

Help is available from Prof. Walker and from Mr. Yisong Jiang, the course:

- For questions on what the assignment is asking, please contact Prof. Walker.
- For questions on MPI, or for help with your code or debugging, please contact Mr. Jiang (***not*** Prof. Walker)

Our contact information and office hours are on the class web page, and may be extended if necessary as the project deadline approaches; see the class web page for any announcements of extended office hours.

Cooperation versus Cheating

See the class syllabus, and contact me if you have any questions. You are allowed to discuss the problems with your friends, and to study MPI programming with your friends, but you are not allowed to write pseudo-code to solve the problems with your friends, and you are certainly not allowed to copy anyone else's code.

Submitting Your Project

When you finish, submit all files that you modified to the TA, just as you did in Project 1.

Important warning — once you submit your files, **DON'T TOUCH THEM AGAIN** — if your email didn't reach the TA, or something happens, the TA may need to ask you to resubmit your files. However, before he lets you do so, he will ask you to log on in his presence, and he will check the modification dates on your files to make sure that they haven't been modified after the due date (if they have been, you will be assessed the appropriate late penalties).