
Due via email by 11:59pm on Friday 16 February 2001

Introduction

In this assignment, you are given MPI code showing how to construct a system with a manager and four workers. Your job is to use that code as a basis for a small client / server system. This project will comprise 10% of your final course grade.

Getting Started

To begin, review the material presented in Lecture 06 on Monday 5 February and in Lecture 07 on Wednesday 7 February, and then read the file “Using LAM/MPI in the KSU MCS Dept.” on the class web page. Then set up your account for running MPI, and test the example provided to make sure it runs and produces the expected output.

Once you have MPI set up, read the sample solution “average.c” to the problem posed on slide 13 of the ND MPI Tutorial, Part 2. For convenience, this solution is also available on the class web page.

The Problem

In this assignment, you are to (i) use MPI and the “average.c” program as a basis for implementing a small client / server system, and (ii) come up with appropriate test code or methods to demonstrate that your implementation works for at least 2 clients and for 1 server. The “server” can perform any action of your choice, although for simplicity you might just want it to do some computation — for example, compute the average of a set of values sent to it.

What to Turn In

- (30 points) In a file called **p1.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 how well you could implement a client / server system in MPI — is this a good programming environment for a client / server system? 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 choices to make in your implementation. A superficial half-page writeup will definitely not be worth 30 points!
- (70 points) The implementation of your project and test code / methods. If it doesn't quite work as desired, clearly describe in the file **p1.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 the project overview file and all of your code to the TA. Assuming you have the overview file **p1.overview** and your code in the file **cliserv.c**, you can submit those files to the TA for grading by typing the following commands in the directory that contains those files:

```
shar p1.overview cliserv.c >shar.out           (type all this on one line)
elm -s "P1 for Your Name" jyisong@mcs.kent.edu <shar.out
                                                (type all this on one line)

rm shar.out
```

The first line puts your files into a single file called **shar.out**, and the second line emails that file to the TA (replace "Your Name" with your own name).

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).

The project is due at 11:59pm on Friday 18 February 2001. For a discussion of my late policy, see the class syllabus. However, you should probably plan on starting early, ending on time, and then spending the weekend resting or working on something else, instead of trying to perfect a late project.