

Distributed Project

Assignment due:

Project description: 8 pm on Monday, April 21st. Use email to {patrick@cs, rsharma@cs} to send a text description, or use turnin to provide either a text or pdf file.

Due: Project demonstrations will be scheduled for Monday to Wednesday, May 5th to 7th. You will need to turnin your project before your demonstration.

Description:

You are to design and implement a concurrent program that makes creative use of several processes running on a distributed set of computers. You may use MPD, C with MPI, C with sockets, Java with sockets, or Java with RMI, or combinations of these. You may substitute other flavors of C; i.e., C++, Objective-C, C#.

The choice of the program is purposely unspecified; pick something that you think would be interesting and educational.

Exercise 7.26, pages 359-360, describes several possible programming projects. Other projects are possible. For example, MPD provides an interface to the X-Window libraries and Java has tools to create a GUI. You could take a parallel solution to the first project (or some similar parallel application) and add an animation window to it. The animation code would run on one computer, the parallel computation. Games are another source of ideas (I can suggest some). Animations that require multiple machines and/or displays are another source. Consider, for example, a game or animation where objects move across multiple displays each attached to a different machine. Games that allow participation from more than one machine are another example.

Machines available for this project include: lectura, voltron, the Linux machines in GS 228 (fd01 to fd08). You can use any combination of these machines as appropriate for your project.

Unlike the previous project, there is not a requirement for performance measurements. If you want to do a project that is performance-based, talk with me about the machines needed and we will set up reservation slot(s) for timing runs.

Project teams:

You may work solo or in teams of two. Two person teams are expected to undertake a more ambitious, or more polished, project. Team membership can be the same as for the parallel project or different.

Project description: Monday, April 21st

Give me a brief (one to three paragraphs) description of what you propose to do, and the members (solo or group) of the team.

I will give you feedback by the 23rd. I can respond earlier if you provide an earlier description. You are welcome and encouraged to discuss possible topics with me (email, in person, phone) before the 21st.

Final report and demonstration: no later than Wednesday, May 2nd

You will demonstrate your project for me. You will need to prepare the demonstration. Plan a presentation that will last about 15 minutes. This will be followed by my questions. The total time for demonstration and questions will be no longer than 30 minutes. If doing a two-person project, both members of the group must be present. A sign-up sheet will be available for times on Monday, Tuesday and Wednesday, May 5th to May 7th. Note that my class schedule will result in limited times available on Tuesday, the 6th. You can also make an appointment for a demonstration time prior to the 5th. I will have a sign-up sheet available in class on Thursday, May 1st.

Demonstrations can be done in my office, in GS-228, or in GS-930. The demonstration can also involve personal laptops.

See the turnin item below. Do the turnin before your demonstration.

At the time of your demonstration, provide:

- a written description of your project — not more than 5 pages.
- Include in your report a diagram of your program that shows how it is distributed across the machines you are using the communication among the machines/processes.

Turnin:

Prior to your demonstration, use the turnin command to turnin all source files needed for your project. Include a **Makefile** and/or instructions on how to compile the project. Include a README file that explains how to run the project (i.e., what command-line options are needed, are specific machines needed, etc.).

Use the assignment name **422distributed** for this turnin.