Computer Science 221
Topics in Programming (Parallel Computing)
May 2017

Instructor: Dr. Chris Healy (phone: 294-2233, email: chris.healy@furman.edu)

Class meetings: Mon-Fri 9:00 to 12:00 in Rooms 203 and 204 in Riley Hall.

Office hours: Mon-Fri 12:00 to 4:00 in 200-I Riley Hall (and also by appointment). If I am not in the office, then I may be in one of the labs (201,202,203) helping another student.

Textbook: *An Introduction to Parallel Programming (2nd* edition), by Peter Pacheco. We will cover chapters 1-3 and 6.

Course objectives:

1. Put a Raspberry Pi computer together from its off-the-shelf components. Plug in hardware components, and install an operating system and other software. Connect several machines into a cluster to support parallel applications.
2. Learn and use routine Linux commands.
3. Be able to write programs in the C programming language.
4. Use the MPI package to write a variety of parallel computer programs.

Grade calculation:

40%	quizzes
20%	lab work
40%	programming assignments

On a typical class day, we will begin with a short lesson or quiz in room 204, before going to 203 for lab work. This is a lab-intensive course. For the most part, we will learn by doing. Therefore, it is essential that you come to every class and stay for its entire duration. Each unexcused absence will reduce your course grade by five percentage points. Also, before you leave the lab at the end of the class period, you must have your work checked by the instructor to show that your work is up to date.

During this course, you will work on your own cluster of computers with a partner. You will keep the same partner and cluster throughout the course. During lab work, feel free to help and seek help from each other (anyone in the room) during class. If you are not finished with lab work by 12:00, you may continue working or return at any time during the afternoon or evening to finish. No one else is using the computer lab during this term. Please leave the doors locked for security.

In addition to routine lab work, there will be programming assignments for you to complete outside of class hours. Of course, if your team finishes the daily lab work early, you may begin working on the assignment during class. They are designed to be done separately by each pair of students. On an assignment, you may not take or copy code or an algorithm from another team or from outside people. No cheating! If you have any question regarding what is and what is not allowed, please don't hesitate to ask me. For students involved in cheating, penalties will be substantial, up to and including an automatic failing grade in the course. "I didn't know it was wrong" is no excuse.

The purpose of the quizzes is to make sure that you individually are acquiring sufficient knowledge and understanding of the content of the course. If you are absent from a quiz, you will earn a score of zero, unless your absence is excused. If you know in advance that you cannot take a quiz, please let me know as soon as possible so that you can take it early. Otherwise, if you are absent from a quiz due to an excused absence, then you will be given an oral examination or individual homework assignment to replace the quiz grade.

Don't forget the most important thing – I am here for you, and I want you to succeed. Please come to my office anytime for help or advice in this course.

Some tips for success:

1. Be organized. Take notes and keep a log of what you have accomplished each day.

2. Be patient with yourself. There is no need to rush. And don't worry if your first attempt at a solution is wrong. The types of programs we will write will take anywhere from 15 minutes to several hours to complete. For the longer assignments you will need to work methodically, comment your code as you go, and realize that you don't need to finish everything in one sitting. Get into the habit of breaking a problem into small manageable pieces or steps.

3. Remember that there is more than one way to solve a problem. That is especially key in this class. For example, you may write a computer in a serial fashion as you are already accustomed to doing, and then parallelizing your algorithm.

4. Be curious. Always ask questions. When finishing up a problem, think about whether this problem or its solution lends itself to other problems.

5. Have fun. Live in the moment (i.e. don't dwell too much on the difficulties of yesterday or tomorrow). Enjoy the intellectual journey and feast. Help your partner and others as appropriate. Be enthusiastic about what you are doing.