Purpose: The purpose of an Operating Systems project is to allow you to gain hands-on experience on a topic related to our course in greater depth.

Goal: To present a poster paper on School of Mathematics, Engineering, and Business Scholarship Day, summarizing the results of your investigations.

If you have a creative idea that you are not sure is related enough to our course, see me. Plan ahead in case you need inter-library loans, or need to gain access to computer resources, whether local like WinXP or Linux, or remote.

Since Operating Systems is a broad topic, you may do a project jointly with work you are doing in another course. Then the instructor in the other course must approve it too. It will be more substantial than a project for either course separately.

Teamwork: If different people suggest similar topics, your must either collaborate or your projects must complement each other. Team projects are permitted and encouraged, but are not required. If you work in a team the result should represent more depth than an individual’s project.

Format: Treat the project writeup the way you have written up laboratories for me in the past. If you have not had me before, see a very brief example at <http://home.messiah.edu/~chase/csc/416/labReportSample.pdf>. In this larger lab, you have the following sections: Problem (Purpose), Discussion (Results), Limitations, Conclusions, Suggestions for Further Work, Works Cited, and an Appendix with any computer code. General guidelines for paper format in my Computer Science courses are at <www.messiah.edu/acdept/dept/home/mathsci/courses/paperformat.htm>. Printing those guidelines including the guidelines linked from there about writing clear code is a good idea if you think that you need help in formatting a research paper. You are going to present your work as a “poster paper” this year. More details on that later.

Your typed project report should be in Times Roman 12, double-spaced, with pages numbered, and any code should be in an appendix in a single-spaced monospaced font. Use MLA format for in-line citations and for Works Cited. Aim for about 8 to 10 pages in length including your program. You may do the same topic as in another course such as Senior Seminar, but then the length must be the sum of the lengths required of both papers, and prior approval must be granted by the instructors in both courses. I will grade the paper, but I will also grade a version of it which includes just the highlights pasted in much larger print onto a poster, to show off what your project does. That's where you sell the idea of your project.

Grading criteria: Creativity, comprehensiveness (breadth, depth), clarity (focus, organization, liveliness in writing, use of figures and diagrams and demonstrations), and correctness (relevance to our course, good choice of current periodicals and books and web sites as resources, technical accuracy). The project is worth 15% of your grade, 5% awarded for timely and thoughtful milestones along the way, and 10% for the final product.
Milestones:
• First milestone, due Friday, February 10, 8:00 a.m.:
  Proposed topic area and decision about whether to work together or separately.
  This should be one or two paragraphs, typewritten, 8.5”x11” paper.
• Second milestone, due Friday, February 17, 8:00 a.m.:
  Narrow topic down and supply references.
  You should list at least five resources that you expect to use beyond your textbook, with
  one source at least not a web site and one source a journal article (whether found via the
  Library's on-line full-text journal articles in for example Academic Search Premier or J-STORE,
  or on the Library's open stacks). Be sure to reference your sources fully and in correct MLA
  style. You will be graded both on the content and on the format. Present it as professionally as
  you know how.
• Third milestone, sometime during week of Monday, February 20:
  Consultation with instructor and request for resources.
• Additional milestones to be assigned, culminating in a poster and a paper on Friday, April 28.

Suggested topics:
Here are a few examples of topics that are only covered minimally in the text or are not covered at all,
but are interesting and current and lend themselves to this project. Depth is of more value than breadth.
All of the following have been done successfully in this course in previous years.

• Run and discuss a specific modern operating system, or some aspect of it, such as Mach, Amoeba,
  Oberon, QNX, or other. Several of these are available for free downloading. Frey 366 is
  available for your use so you need not mess up your own computer.
• Use CORBA for inter-process communication or another method not the one that you use in the
  “Candy Factory” homework.
• Create a Beowulf cluster by using Condor with Windows XP.
• Simulate a job queue with arrival times (if you know some Statistics).
• Compare disk scheduling algorithms (or memory management algorithms) by simulation.
• Simulate the hardware Data Encryption Standard (DES) in software.
• Write an Ebay or Amazon or Google Earth client.
• Write a program to illustrate load sharing, perhaps modifying something that you find on the web.
• Tune an operating system parameter in Linux and demonstrate the difference in behavior between the
default value and the value you set. (For example, change the time quantum.)
• Discuss cache coherency in multiprocessor environments.
• Discuss and illustrate process scheduling in Linux.
• Identify and propose solutions for several security issues with Windows XP.
• Discuss how Java's JavaSpaces implements the Linda parallelizing extension to programming
  languages [The library has two articles in Byte magazine, from Nov 1988, and from May 1990; I
  have three articles in my How, How to write parallel programs; Applications experience with
  Linda; and Linda in Context, from Communications of the ACM, Apr 1989. The last three
  papers are by Carriger and Gelernter. IBM’s Java offering for Linda is called T-spaces. It is also
  available at no cost.]
• Reliability. Subtopics include techniques to minimize system failure, correctness of operating system
• Network operating systems [If you have done or are doing a paper for Networking, then be careful to
  work on something different from that.]. For example, demonstrate OpenAFS and contrast it
  with NFS.