Instructor

Craig E. Wills, FL-236, cew@cs.wpi.edu. Office hours: as needed by students. Any time for short questions. Electronic mail is an effective method to contact me.

Course Web Page

Copies of all handouts, assignments and notes will be posted as appropriate on the course Web page. The address for it is http://www.cs.wpi.edu/~cs535/f05/.

Purpose

This is a graduate-level course in the design of advanced systems. It focuses on the issues of advanced operating systems and distributed systems, which have evolved from general-purpose multiprogramming systems covered in previous courses. The goals are 1) to familiarize the student with current literature in the area, 2) for students to be able to place work in its context in terms of its relative importance and relationship with other work, 3) to give students experience in making public presentations of technical work, and 4) to develop students’ ability for critical thinking and discussions concerning design choices, tradeoffs, and their consequences.

Prerequisites

A first course in operating systems, such as CS 502. Background in architecture, networking, compilers and programming languages would also be helpful for issues related to these topics. An interest in reading, thinking about and discussing issues in advanced system design.

Text Book

There is no required text book for the course. Relevant text books are:

They provide reasonably up-to-date discussions on issues in advanced and distributed systems. Some material from these sources will be introduced in class.

Much of the course will also be based on seminal and current literature in the field. Literature will be selected both by the instructor and the students. Access to this material will be discussed in class.

## Grading Policy

Final grades will be computed as follows:
- Homework: 35%
- Final Project and Presentation: 30%
- Final Exam: 25%
- Class participation: 10%

Final grades will reflect the extent to which you have demonstrated understanding and insight for the material. No incomplete grades will be assigned unless there exist exceptional, extenuating circumstances. There will be one final exam. There will be one final project that may or may not involve programming.

Class meetings will consist of discussions. Each student should be prepared to pose and answer questions on the day a paper or topic is discussed. When reading papers you are expected to take notes on the main points of each article, questions concerning its contents and suggestions on how the described research can be followed up. You should try to discern the strengths and weaknesses of the paper. After reading the paper, you should write a summary of the paper along with questions and suggestions for followup (roughly a page or so). It is important to include your personal thoughts on the paper concerning what you liked or disliked about the paper. An excellent review will include a summary containing the key points of the paper along with personal observations of the paper.

For your own benefit you should maintain an online bibliography of the papers you read. The bibliography should contain papers you read for the class and any other references of interest that you come across as you read or peruse the literature. These entries will provide ideas for further reading for the course.
Late Policy

You are expected to keep up with the pace of papers for the course. Entries turned in late will be penalized 5% of total assignment value per day or partial day. All entries are due at the start of class on the due date. Those turned in after the start of class will be counted late. Exceptions to these rules can be made only a priori. Finally, no entries will be accepted after Friday, December 16 to allow sufficient time for grading.

Cheating

Unless explicitly noted, all work is to be done on an individual basis. You are encouraged to talk with others about ideas and material in the course, particularly in preparing for exams. However all work, in the form of reviews, code or answers to problems, you submit for grading must be your work. Misrepresentation of the work of another as one’s own submitted work is a violation of academic honesty. Aiding someone else to commit an act of academic dishonesty is also a violation. Submitting individually-assigned work that was jointly done with another person is a violation of academic honesty.

Any violation of the WPI’s guidelines for academic honesty will result in no credit for the course and referral to the Student Affairs Office. More information on definitions, responsibilities and procedures regarding the WPI academic honesty policy can be found at http://www.wpi.edu/Pubs/Policies/Judicial/sect5.html.
Schedule

The following is a tentative outline of the material that will be covered in this course.

week 1: 9-12. Course outline, contents, grading policy, introduction, and overview.
Papers to read: [102, 66].

Papers to read: [11, 12].
Other papers of interest: [106, 50, 7, 46, 54, 40, 18, 69, 32, 65, 114].

week 3: 9-26. Performance evaluation/Benchmarking
Papers to read: [24, 45].
Other papers of interest: [89, 43, 73, 35, 25, 59, 92].

Papers to read: [20, 17].
Other papers of interest: [10, 70, 77, 33, 15, 49, 99, 53, 78, 81].

week 5: 10-10. Synchronization—clocks, mutual exclusion, deadlocks.
Papers to read: [60].
Other papers of interest: [101].

week 6: 10-17. Processes and multithreaded processes, threads, memory management, load sharing, scheduling.
Papers to read: [34, 36].
Other papers of interest: [64, 21, 22, 52, 3, 90, 14, 100, 48, 5, 71, 109, 63, 62, 30, 82].

Papers to read: [6, 42].
Other papers of interest: [2, 75, 80, 57, 86, 94, 44, 97, 4, 104, 68, 93].

weeks 8-14: 10-31 – 12-12
Other issues:
Name servers/resource location [103, 47, 16, 98, 19].
Object-oriented systems [29].
Protection and security [37, 31, 108, 112, 113].
Experience with systems: [27, 91, 61].
Distributed operating systems [74, 111, 107, 88, 79, 26].
Distributed computing environments [83, 41, 58, 38, 28, 51].
Web as a Distributed System [95, 13, 72, 23, 8, 55, 56].
Caching/Prefetching [76].
Peer-to-Peer Systems [85, 96].
Wireless/Mobile Systems [105, 87, 39, 1].
Network Servers and Services [110, 9, 84].
Sensor Networks [67].
Systems Literature

We will be reading articles from journals and conference proceedings. Three primary publishers of journals and conference proceedings are ACM (www.acm.org), IEEE Computer Society (www.computer.org) and USENIX (www.usenix.org). Each of these has their own digital library, which is a prominent link from each of these organization’s home page. WPI is a member of each of these organizations and access from a machine on the WPI campus will allow download of all articles from ACM and IEEE. If you are off-campus, then you can use the WPI library proxy (http://www.wpi.edu/Academics/CCC/Help/Software/proxy.html) to gain access to these repositories. USENIX allows access to most of their collection without a password. If you need to access a collection that is less than a year old then a member password may be needed. Contact the instructor to obtain the WPI password.

Another source of articles is available online. It is a bibliography of distributed systems and World Wide Web systems literature. This bibliography can be found at http://www.cs.wpi.edu/~webbib/. This bibliography contains sources for systems literature and specific bibliographic references with links to papers.

Articles are also available in the journals and proceedings themselves. The following lists sources that are usually, if not always, systems related. Location of the literature is given if known to be available in the WPI library (indicated by *) or the Computer Science Department (indicated by †). Other good libraries should hold many of these journals and proceedings, although online access as described above is often easiest.


ACM Symposium on Operating Systems Principles (published as part of Operating System Review)*

ACM Transactions on Computer Systems*

HotOS: Workshop on Hot Topics in Operating Systems (USENIX)

IEEE Proceedings of the International Conference on Distributed Computing Systems*

Operating Systems Review (ACM)*

Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ACM)

USENIX General Conference†

USENIX Symposium on Operating Systems Design and Implementation†

USENIX Workshop on Hot Topics in Operating Systems†

USENIX Symposium on Networked Systems Design and Implementation†

Articles in these journals and conference proceedings are sometimes systems related.

ACM Computing Surveys*
Communications of the ACM*
Computer Communications*
Computer Networks*
IEEE Computer*
IEEE International Symposium on Computer Architecture*
International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*
ACM Transactions on Internet Technology*
IEEE Internet Computing*
IEEE Network*
IEEE/ACM Transactions on Networking*
IEEE Transactions on Software Engineering*
IEEE Transactions on Computers*
IEEE Transactions on Parallel and Distributed Systems*
International World Wide Web Conference
http://www.w3.org/pub/Conferences/Overview-WWW.html
Journal of Parallel and Distributed Computing*
Performance Evaluation*
Performance Evaluation Review (ACM)*
SIGCOMM Symposium on Communications, Architectures and Protocols (ACM) (published as part of Computer Communication Review)*
Software Practice and Experience*

Other journals and conference proceedings not listed here may also contain relevant material.

References


[28] Common object request broker architecture design specification.


[52] Michael B. Jones and John Regehr. The problems you’re having may not be the problems you think you’re having: Results from a latency study of Windows NT. In Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, February 1999.


