Links

- Overview
- Reading List
- Schedule

Overview

This is a first-year graduate class in operating systems. As such, this class will give you a deeper understanding of OS subsystems for process management, memory management, and so on, and expose you to classic and contemporary literature on the subject. The class also has a hands-on implementation project which will give you insight into what it takes to build real networking sub-systems.

The class has the following components:

- **Lab Assignments (50%)**. During the course of the semester, you will complete six lab assignments, filling in various pieces of an educational operating system. Starting from a bare-bones operating system, this sequence of lab exercises asks you to fill in important pieces of process management, memory management, filesystems and network communication. At the end of the course, you will have a functional Web server running on top of your operating system! All lab assignments are to be done individually. For the lab assignments, ***you will need to program in C***.

- **In-class quizzes (50%).** There will be four quizzes. Each quiz will test not only the reading materials, but also your understanding of the lab exercises. All quizzes will be closed book (but you may bring one sheet of 8.5x11 paper with anything written on it) and will contain a mixture of multiple-choice and short answer questions.

Student Responsibilities

By enrolling this class, here is what you're agreeing to:

- All work turned in on programming assignments and exams must be that of the individual student. If the instructor finds any evidence of cheating on an exam, he reserves the right to discipline the student appropriately (see below).

- This policy also holds for programming assignments. In this class, we will use sophisticated automated program checkers to detect cheating. Be aware that the program checkers have demonstrated very good results and are widely used within the academic community. Any student caught cheating ***will be given zero*** for the assignment and ***will be reported to SJACS***.

- You must keep up with the assigned readings. If you come to class without having read the papers for the corresponding lecture, you're unlikely to learn anything at all from the lecture. In particular:
  - You are expected to read all the papers in detail. Not all details will be covered in class. I will assume knowledge of material covered in CS 402 and EE/CS 450, a C/C++ programming proficiency from CS402 or its equivalent, and Python programming profiency. If you covered the introductory material at some other school it is YOUR responsibility to fill in any missing background. Feel free to ask me for advice on appropriate introductory readings if you feel your background is insufficient.
  - I expect you to attend every class meeting. If you do happen to miss a session, you are responsible for finding out what material was covered and if any administrative announcements were made.

News and Discussion Forum

All interaction with the instructor and with other students for the class will be through its Piazza site. On the first day of classes, I will announce more details about Piazza.

You are expected to check Piazza periodically for important announcements.

If you wish to contact me, please use Piazza (rather than my regular email), and I will try to respond to your question quickly. While you should always feel free to contact me, I strongly encourage you to post your question to Piazza; you will usually get much faster responses from your fellow students.

Reading List

Early Operating Systems

- **[Ritchie:Unix]** Ritchie, Dennis M., Thompson, Ken, The UNIX Time-sharing System, *Commun. ACM*, Vol. **17**, No. 7, pp. 365--375, ACM, New York, NY, USA, July 1974. **[HTML]**
- **[Ritchie:Evolution]** Dennis M. Ritchie, The Evolution of the Unix Time-sharing System, *Communications of the ACM*, Vol. **17**, pp. 365--375, 1984. **[PDF]**

Alternative OS Structures

- **[Engler:Exokernel]** Engler, D. R., Kaashoek, M. F., O'Toole,Jr., J., Exokernel: An Operating System Architecture for Application-level Resource Management, *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*, pp.

251--266, ACM, New York, NY, USA, 1995. [HTML]

- **[Barham:Xen]** Barham, Paul, Dragovic, Boris, Fraser, Keir, Hand, Steven, Harris, Tim, Ho, Alex, Neugebauer, Rolf, Pratt, Ian, Warfield, Andrew, Xen and the Art of Virtualization, *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 164--177, ACM, New York, NY, USA, 2003. [HTML]
- **[Hunt:Singularity]** Larus, James, Hunt, Galen, The Singularity System, *Commun. ACM*, Vol. **53**, No. 8, pp. 72--79, ACM, New York, NY, USA, August 2010. [HTML]

Process Management and Concurrency Control I

- **[Anderson:Activations]** Anderson, Thomas E., Bershad, Brian N., Lazowska, Edward D., Levy, Henry M., Scheduler Activations: Effective Kernel Support for the User-level Management of Parallelism, *Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles*, pp. 95--109, ACM, New York, NY, USA, 1991. [HTML]
- **[Waldspurger:Lottery]** Waldspurger, Carl A., Weihl, William E., Lottery Scheduling: Flexible Proportional-share Resource Management, *Proceedings of the 1st USENIX Conference on Operating Systems Design and Implementation*, USENIX Association, Berkeley, CA, USA, 1994. [HTML]
- **[Savage:Eraser]** Savage, Stefan, Burrows, Michael, Nelson, Greg, Sobalvarro, Patrick, Anderson, Thomas, Eraser: A Dynamic Data Race Detector for Multithreaded Programs, *ACM Trans. Comput. Syst.*, Vol. **15**, No. 4, pp. 391--411, ACM, New York, NY, USA, November 1997. [HTML]
- **[Boyd-Wickizer:Non-scalable]** Silas Boyd-wickizer, M. Frans Kaashoek, Robert Morris, Nickolai Zeldovich, Non-scalable locks are dangerous. [PDF]

Memory Management

- **[Navarro:Superpages]** Navarro, Juan, Iyer, Sitararn, Druschel, Peter, Cox, Alan, Practical, Transparent Operating System Support for Superpages, *SIGOPS Oper. Syst. Rev.*, Vol. **36**, No. SI, pp. 89--104, ACM, New York, NY, USA, December 2002. [HTML]
- **[Waldspurger:VMWare]** Waldspurger, Carl A., Memory Resource Management in VMware ESX Server, *SIGOPS Oper. Syst. Rev.*, Vol. **36**, No. SI, pp. 181--194, ACM, New York, NY, USA, December 2002. [HTML]
- **[Clements:RadixVM]** A. Clements, M. Frans Kaashoek, N. Zeldovich, RadixVM: Scalable Address Spaces for Multithreaded Applications, *Proc. Eurosys*, 2013. [PDF]

File Systems

- **[Patterson:RAID]** Patterson, David A., Gibson, Garth, Katz, Randy H., A Case for Redundant Arrays of Inexpensive Disks (RAID), *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, pp. 109--116, ACM, New York, NY, USA, 1988. [HTML]
- **[Rosenblum:LFS]** Rosenblum, Mendel, Ousterhout, John K., The Design and Implementation of a Log-structured File System, *ACM Trans. Comput. Syst.*, Vol. **10**,

No. 1, pp. 26--52, ACM, New York, NY, USA, February 1992. **[HTML]**

- **[Tweedie:Journaling]** Stephen Tweedie, Journaling the Linux ext2fs Filesystem, *Linux Expo*, 1998. **[PDF]**
- **[Ghemawat:GFS]** Ghemawat, Sanjay, Gobioff, Howard, Leung, Shun-Tak, The Google File System, *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 29--43, ACM, New York, NY, USA, 2003. **[HTML]**

Process Management and Concurrency Control II

- **[Baumann:Barrelfish]** Baumann, Andrew, Barham, Paul, Dagand, Pierre-Evariste, Harris, Tim, Isaacs, Rebecca, Peter, Simon, Roscoe, Timothy, Sch\"upbach, Adrian, Singhania, Akhilesh, The Multikernel: A New OS Architecture for Scalable Multicore Systems, *Proceedings of the ACM SIGOPS 22Nd Symposium on Operating Systems Principles*, pp. 29--44, ACM, New York, NY, USA, 2009. **[HTML]**
- **[Clements:Commutativity]** Clements, Austin T., Kaashoek, M. Frans, Zeldovich, Nickolai, Morris, Robert T., Kohler, Eddie, The Scalable Commutativity Rule: Designing Scalable Software for Multicore Processors, *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pp. 1--17, ACM, New York, NY, USA, 2013. **[HTML]**
- **[Belay:Dune]** Belay, Adam, Bittau, Andrea, Mashtizadeh, Ali, Terei, David, Mazi\`eres, David, Kozyrakis, Christos, Dune: Safe User-level Access to Privileged CPU Features, *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, pp. 335--348, USENIX Association, Berkeley, CA, USA, 2012. **[PDF]**

Communication

- **[Bershad:LRPC]** Bershad, Brian N., Anderson, Thomas E., Lazowska, Edward D., Levy, Henry M., Lightweight Remote Procedure Call, *ACM Trans. Comput. Syst.*, Vol. **8**, No. 1, pp. 37--55, ACM, New York, NY, USA, February 1990. **[HTML]**
- **[Liedtke:IPC]** Liedtke, Jochen, Improving IPC by Kernel Design, *Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles*, pp. 175--188, ACM, New York, NY, USA, 1993. **[HTML]**
- **[Belay:IX]** Adam Belay, George Prekas, Ana Klimovic, Samuel Grossman, Christos Kozyrakis, Edouard Bugnion, IX: A Protected Dataplane Operating System for High Throughput and Low Latency, *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pp. 49--65, USENIX Association, Broomfield, CO, October 2014. **[HTML]**

Reliability, Fault Isolation, and Debugging

- **[Gray:Stop]** Jim Gray, Why Do Computers Stop And What Can Be Done About It?, 1985. **[PDF]**
- **[Wahbe:SFI]** Wahbe, Robert, Lucco, Steven, Anderson, Thomas E., Graham, Susan L., Efficient Software-based Fault Isolation, *SIGOPS Oper. Syst. Rev.*, Vol. **27**, No. 5, pp. 203--216, ACM, New York, NY, USA, December 1993. **[HTML]**

- **[Engler:Bugs]** Engler, Dawson, Chen, David Yu, Hallem, Seth, Chou, Andy, Chelf, Benjamin, Bugs As Deviant Behavior: A General Approach to Inferring Errors in Systems Code, *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, pp. 57--72, ACM, New York, NY, USA, 2001. **[HTML]**
- **[Bessey:Billion]** Bessey, Al, Block, Ken, Chelf, Ben, Chou, Andy, Fulton, Bryan, Hallem, Seth, Henri-Gros, Charles, Kamsky, Asya, McPeak, Scott, Engler, Dawson, A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World, *Commun. ACM*, Vol. **53**, No. 2, pp. 66--75, ACM, New York, NY, USA, February 2010. **[HTML]**

Security

- **[Saltzer:Multics]** Saltzer, Jerome H., Protection and the Control of Information Sharing in Multics, *Commun. ACM*, Vol. **17**, No. 7, pp. 388--402, ACM, New York, NY, USA, July 1974.**[HTML]**
- **[Morris:Password]** Morris, Robert, Thompson, Ken, Password Security: A Case History, *Commun. ACM*, Vol. **22**, No. 11, pp. 594--597, ACM, New York, NY, USA, November 1979. **[HTML]**
- **[Steiner:Kerberos]** Jennifer G. Steiner, Clifford Neuman, Jeffrey I. Schiller, Kerberos: An Authentication Service for Open Network Systems, *IN USENIX CONFERENCE PROCEEDINGS*, pp. 191--202, 1988. **[PDF]**
- **[Garfinkel:Terra]** Garfinkel, Tal, Pfaff, Ben, Chow, Jim, Rosenblum, Mendel, Boneh, Dan, Terra: A Virtual Machine-based Platform for Trusted Computing, *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 193--206, ACM, New York, NY, USA, 2003. **[HTML]**
- **[Anderson:Economic]** Ross Anderson, Why Information Security is Hard - An Economic Perspective, *Proceedings of the Seventeenth Computer Security Applications Conference*, 2001.**[PDF]**

Lessons

- **[Waldo:System]** Waldo, Jim, On System Design, *Proceedings of the 21st Annual ACM SIGPLAN Conference on Object-oriented Programming Systems, Languages, and Applications*, pp. 467--480, ACM, New York, NY, USA, 2006. **[HTML]**
- **[Lampson:Hints]** Lampson, Butler W., Hints for Computer System Design, *SIGOPS Oper. Syst. Rev.*, Vol. **17**, No. 5, pp. 33--48, ACM, New York, NY, USA, October 1983. **[HTML]**
- **[Levin:SOSP]** Levin, Roy, Redell, David D., An Evaluation of the Ninth SOSP Submissions or How (and How Not) to Write a Good Systems Paper, *SIGGRAPH Comput. Graph.*, Vol. **22**, No. 5, pp. 264--266, ACM, New York, NY, USA, October 1988. **[HTML]**

Schedule

Approximate Weekly Schedule

Topics	Date
Introduction, Early OSes	Aug 20
Alternative OS Structures	Aug 27
Alternative OS Structures	Sep 5
Proc. Mgmt. I	Sep 12
Proc. Mgmt I	Sep 19
Memory Management	Sep 26
Memory Management	Oct 3
File Systems	Oct 10
File Systems	Oct 17
No Class: Spring Break	Oct 24
File Systems	Oct 31
Proc. Management II	Nov 7
Proc. Management II, Communication	Nov 14
Communication	Nov 21
Communication, Wrap-Up	Nov 28

Schedule of Exams

All quizzes will be during the quiz session. Please bring a laptop or mobile device to class on the day of the quiz.

Exam	Date
Quiz 1	Sep 17
Quiz 2	Oct 8
Quiz 3	Nov 5
Quiz 4	Nov 28

Lab Submission Deadlines

For each lab, the TA will present, during the Quiz section on the dates indicated below, an outline of the lab, and steps that you should take to do the lab exercise.

Part	Due Date
Lab 1	Sep 4, 5pm
Lab 2	Sep 19, 5pm

Lab 3, Part A	Sep 28, 5pm
Lab 3, Part B	Oct 5, 5pm
Lab 4, Part A	Oct 12, 5pm
Lab 4, Part B	Oct 19, 5pm
Lab 4, Part C	Nov 2, 5pm
Lab 5	Nov 16, 5pm
Lab 6	Nov 30, 5pm

Grading

Final letter grades will be determined using a "modified" curve. I will assign grades of C and BELOW to individuals who do not perform satisfactorily in the above areas (i.e., you should not assume that because this is a graduate class you will get a B or even B- if you perform unsatisfactorily.).

We will not assign incompletes unless it is for a documented medical reason (in accordance with USC policy).

Statement for Students with Disabilities

Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 am - 5:00 pm, Monday through Friday. The phone number for DSP is (213) 740-0776.

Statement on Academic Integrity

USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles. Scampus, the Student Guidebook, contains the Student Conduct Code in Section 11.00, while the recommended sanctions are located in Appendix A. Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty. The review process can be found at this link.