CSC 343 – Operating Systems, Spring 2020 First Day Handout

MonWed 3-4:20 PM, http://faculty.kutztown.edu/parson
Dr. Dale E. Parson, Old Main 260, parson@kutztown.edu, 484-646-4296
Office Hours: Tu 2:30-4:30, Wed 12:00-2:00, Fri 1:30-2:30, or by appointment.

This course is an introduction to the basic software components of an operating system. Topics include process management, storage management, processor management, auxiliary storage management, evaluation of an operating system's performance, networks, operating system security, case studies of particular operating systems.

Prerequisite: CSC 237 or CSC 253 AND completion of 18 credits of CSC courses numbered 125 or higher AND a GPA in the CSC courses of 2.25.

Textbook: *Operating System Concepts*, 10th Edition, Silberschatz, et. al. (9th edition is OK)

OR *Operating System Concepts Essentials*, 2nd Edition, Silberschatz, et. al.

(9th edition) & *Essentials Edition* on **1-day reserve in Rohrbach Library**. Chapter numbers vary.

Grading (A = 92:100, A- = 90:91, B+ = 87:89, B = 82:86, B- = 80:81, C+ = 77:79, C = 70:76, D = 60:69, F = 0:59)

Midterm exam	20% of grade
Final exam	20% of grade (non-cumulative exam)
Projects	60% divided equally among the project assignments.

Programming project assignment grading criteria

5% Concise, accurate documentation using CSC Department requirements spec, **see**: http://cs.kutztown.edu/pdfs/Documentation_Standard.pdf
10% Clean separation of specification of interface from module implementations
5% Documentation of client modules (including tests) that use ADT modules
30% Correct implementation of central data structure or algorithm
20% Adherence to written specification of assignment, including coding standards
5% Willingness to ask questions and clarify any uncertainties about assignments
5% Concise, structured, and accurate documentation of implementation
10% Inclusion of test code, data & results showing coverage of common and corner cases
10% Successful execution against my private evaluation test cases

The academic integrity policy is at http://cs.kutztown.edu/pdfs/AcademicIntegrityPolicy.pdf

Your first reading assignment is to read the above policy statement.
You may openly discuss ideas, algorithms, pitfalls, and the use of programming tools.
You may not share code, test drivers or test data except within groups for group projects.
Group projects, when assigned, have documented partitioning of student responsibilities.

Class attendance is not graded, but I will be teaching using data sources and concepts both inside and outside the scope of the textbook. You are responsible for all material covered in class, including technical information, coding standards and conventions, verbal specification of assignments, and your questions about topics that are not clear to you. Please, there should be no classroom conversations, cell phones, text messaging, eating, sleeping, obscenities, listening to music or other disruptions of the class.

If you have already disclosed a disability to the Disability Services Office (215 Stratton Administration Building) and are seeking accommodations, please feel free to speak with me privately so that I may assist you. If you have an injury sustained during military service including PTSD or TBI, you are also eligible for accommodations under the ADA and should contact the Disability Services Office.

Week	Text chapters	(Essentials Chapter / 10th Edition Chapter) Lecture Topics
1	1	Introduction, computer architecture, hardware / software interface, history
2	2, P[1]	Operating system structures, multiprocessor hardware / software architecture
3	3	Processes, inter-process communication (IPC), modeling using state diagrams
4	4, 5 (10th 4, 6)	Threads, thread libraries, queuing, related state machine models
5	5,6(10th 5,6) P[2]	Process synchronization, CPU scheduling
6	6 (10th 7,8ish)	CPU scheduling, deadlocks
7	7 (10th 9), P[3]	Main memory, memory hierarchies, swapping, paging
8	8 (10th 10)	Review, **Midterm exam, -- 20% of grade**
9	8 (10th 10)	Virtual memory, scheduling memory
10	9,10(10th 11,13)P[4]	Mass storage systems, File system interface
11	11 (10th 14)	File system implementation, directories, allocation / free space management, NFS
12	12 (10th 12), P[5]	Disk scheduling, swap space management, RAID
13	12 (10th 12)	Input-output (I/O) systems, buses and multiprocessor interconnectivity
14		Remaining topics, perhaps one additional chapter from the textbook
15		**Final exam, 20% of grade**

There will be 4 or 5 project assignments, depending on timing. Last offering we completed 4.

[1] Project 1 is a practice project for learning how to construct, compile and simulate state machines based on the UML (Unified Modeling language) State Diagram notation.
[2] Project 2 involves building state machine models to simulate various CPU scheduling algorithms.
[3] Project 3 involves building interacting state machines to share resources and avoid / risk deadlock.
[4] Project 4 involves building a simulation model to simulate memory mapping, swapping and page scheduling.
[5] Project 5 involves building a simulation model to simulate file system space management or I/O scheduling.