CS 346 – Operating Systems
Course Syllabus
Spring, 2018
Dr. Steven K. Andrianoff

Time and Place
Lecture: MWF 12:30 – 1:20 Walsh 203
Lab: Th 5:00 – 7:00 Walsh 101

Instructor
Dr. Andrianoff (andrianoff@sbu.edu)
Office: Walsh 113
Phone: 375-2053
Office hours: Mon 2:00 – 3:30 p.m.
        Tues 1:30 – 2:30 p.m.
        Wed 2:00 – 3:00 p.m.
        Thur 2:00 – 3:30 p.m.
        Fri 10:30 – 11:30 a.m.
(Other times by appointment)

Course web page http://www.cs.sbu.edu/andrianoff/cs346

In general, announcements, readings, and assignments for CS 346 will be given in class AND published on the course web page. Students are expected to check that page regularly for news, and are nonetheless responsible for any assignment announced in either manner.


Catalog Description

A study of modern multi-programmed operating systems including system structure, concurrency, process scheduling and control, memory management, file systems, and system performance. The course consists of three lecture hours and one two-hour laboratory per week. The laboratory component provides experience in concurrent programming using Java threads and in system level programming using C in a UNIX environment. 4 credits.

Prerequisites

CS231 - Computer Organization
Course objectives

1) To understand the underlying principles in the design of a multi-programmed operating system.

2) To gain an understanding of concurrency and be able to write concurrent (multi-threaded) programs.

3) To be exposed to the UNIX operating system as an example of a modern multi-programmed operating system. This will include the ability to use UNIX system tools and write system level code.

Course Topics

<table>
<thead>
<tr>
<th>Number Of Weeks</th>
<th>Topics</th>
<th>Textbook Readings</th>
<th>Relevant Labs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Computer System Overview: Including: Instruction execution, Interrupts, Cache, Multiprocessors, Multicore</td>
<td>Chapter 1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Operating System Overview Including: Evolution, Major Achievements, Microkernel, Virtual Machines</td>
<td>Chapter 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Processes: Process Description and Control Threads</td>
<td>Chapters 3, 4</td>
<td>UNIX Processes Introduction to Java Threads</td>
</tr>
<tr>
<td>2</td>
<td>Concurrency: Mutual Exclusion, Semaphores, Monitors, Inter-Process Communication Deadlock and Starvation</td>
<td>Chapters 5, 6</td>
<td>Racing Threads Producer/Consumer Problem with a Bounded Buffer Semaphore Solutions to the Bounded Buffer and Readers-and-Writers Problem Monitor Solution to the Readers-and-Writers Problem Monitors via Java’s Locks and Condition Variables</td>
</tr>
<tr>
<td>2</td>
<td>Memory: Memory Management, Paging Virtual Memory</td>
<td>Chapters 7, 8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Scheduling</td>
<td>Chapter 9</td>
<td>Scheduling Simulator</td>
</tr>
<tr>
<td>1</td>
<td>Input/Output: I/O Devices, Buffering, Disk Scheduling</td>
<td>Chapter 11</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>File Management</td>
<td>Chapter 12</td>
<td>File IO in C</td>
</tr>
<tr>
<td>1</td>
<td>Computer Security: Threats, Attacks Access Control, Intrusion Detection, Buffer Overflow Attacks</td>
<td>Chapters 14, 15</td>
<td></td>
</tr>
</tbody>
</table>

Learning Objectives

1. Students will be able to explain the steps taken by a processor to execute an instruction.
2. Students will understand the concept of interrupts and how and why a processor uses interrupts.
3. Students will be able to discuss the concept of locality and analyze the performance of a multilevel memory hierarchy.
4. Students will be able to discuss the evolution of operating systems from early simple batch systems to modern complex systems.
5. Students will be able to define and discuss virtual machines and virtualization.
6. Students will understand the OS design issues raised by the introduction of multiprocessor and multicore organization.
7. Students will be able to define the term process and explain the relationship between processes and process control blocks.
8. Students will be able to explain the concept of a process state and discuss the state transitions the processes undergo.
9. Students will be able to list and describe the purpose of the data structures used by an OS to manage processes.
10. Students will be able to understand the issues involved in the execution of OS code and assess the key security issues that relate to operating systems.
11. Students will understand the distinction between process and thread.
12. Students will be able to explain the difference between user-level threads and kernel-level threads.
13. Students will be able to discuss basic concepts related to concurrency, such as race conditions, OS concerns, and mutual exclusion requirements.
14. Students will be able to define and explain semaphores, monitors, and message-passing mechanisms.
15. Students will be able to list and explain the conditions for deadlock.
16. Students will be able to explain the difference between deadlock prevention and deadlock avoidance.
17. Students will be able to explain the fundamental difference in approach between deadlock detection and deadlock prevention or avoidance.
18. Students will understand the reason for memory partitioning and explain the various techniques that are used.
19. Students will understand and explain the concepts of paging and segmentation and assess the relative advantages of each.
20. Students will be able to summarize key security issues related to memory management.
21. Students will be able to define virtual memory and describe the hardware and control structures that support virtual memory.
22. Students will be able to explain the differences among long-, medium-, and short-term scheduling.
23. Students will be able to compare the performance of different scheduling policies and algorithms.
24. Students will be able to discuss the organization of the I/O function and explain some of the key issues in the design of OS support for I/O.
25. Students will be able to explain the concept of RAID and describe the various levels.
26. Students will be able to describe the basic concepts of files and file systems and understand the principal techniques for file organization and access.
27. Students will understand the design issues for file system security.
28. Students will understand the spectrum of computer security attacks.
29. Students will be able to define and compare three methods of user authentication.
30. Students will be able to explain the basic principles and techniques of intrusion detection and malware defense.

Grading

Grades will be based on:
1. Homework & laboratory assignments 1/3
2. Quizzes (8) 1/3
3. Final exam 1/3

Homework assignments

Written homework problems will be periodically assigned, collected, and graded. A 10% per day penalty will be assessed for late assignments up to the time the graded work is returned by the instructor. No late assignments will be permitted after the graded work is returned.
Laboratory assignments

The labs are scheduled for Thursday, 5:00 – 7:00 p.m. This time will be devoted to laboratory assignments which will involve operating system level (UNIX) programming in C and multi-threaded programming in Java. Linux will be used in the lab assignments. Each lab assignment will require a write-up that is written using a word processor. Late lab assignments will be penalized 10% per day late.

Quizzes

Approximately eight quizzes will be given. Only the five highest quiz scores will be kept. There is generally no make-ups for missed quizzes.

Final exam The final exam is comprehensive and is scheduled for Tuesday, May 8 at 10:35 a.m.

Attendance policy

There is no attendance requirement, however students are expected to attend all of the classes and will be responsible for all assignments. Attendance will be monitored. More than three absences is considered excessive.

Absences from class, excused or not, do not exempt the student from completing the assigned work. If you miss a class it is your responsibility to arrange with your instructor for the work to be made up in a timely manner.

Academic integrity policy

Academic dishonesty is inconsistent with the moral character expected of students in a university committed to the spiritual and intellectual growth of the whole person. It also subverts the academic process by distorting all measurements. It is a serious matter and will be dealt with accordingly. A list of unacceptable practices, penalties to be assigned, and procedures to be followed in prosecuting cases of alleged academic dishonesty may be found in the Student Handbook.

Students are expected to read and abide by the department’s Academic Practices and Policies, a copy of which will be distributed with the course syllabus. Unless other instructions are explicitly stated all graded work will be subject to the policy

“Individual Project With Limited Collaboration: In particular, you may receive help from the following persons, in addition to an instructor in this course: any St. Bonaventure University student enrolled in CS 331, and any other person specifically approved by your instructor. You may use the following materials produced by other students: NONE.”

In addition, if you do collaborate with anyone other than the instructor, there must be a note to that effect at the top of the solution you turn in.

Academic dishonesty in any form will not be tolerated. Typically the first offense will result in a zero on the assignment. Repeated offenses will likely result in a failing grade for the course. Any offense deemed punishable will also be referred to the Dean of Arts and Sciences.
Services for Students with Disabilities

Students with disabilities who believe that they may need accommodations in this class are encouraged to contact the Disability Support Services Office, Plassmann 100D, at 375-2065 as soon as possible to better ensure that such accommodations are implemented in a timely fashion.