Course Number & Name: CSC 228 Operating Systems
Credit Hours: 4.0  Contact Hours: 4.0  Lecture: 4.0  Lab: N/A  Other: N/A
Prerequisites: Grade of “C” or better in CSC 225
Co-requisites: None  Concurrent Courses: None
Course Outline Revision Date: Fall 2010

Course Description: This course examines the concepts, designs, and operations of modern real-time, general-purpose operating systems. The course covers fundamental operating system technology as well as contemporary design principles such as real-time systems, multiprocessor scheduling, memory management, file management, and security and network processing. Students are required to complete a selected series of programming projects that illustrate operating system design principles.

Course Goals: Upon successful completion of this course, students should be able to do the following:

1. apply process scheduling and dispatching;
2. describe and implement distributed and real-time systems;
3. explain and use multi-tasking, pre-emptive scheduling and time sharing; and
4. describe system security.

Measurable Course Performance Objectives (MPOs): Upon successful completion of this course, students should specifically be able to do the following:

1. Apply process scheduling and dispatching:
   1.1 discuss how and why process scheduling and dispatching are used; and
   1.2 implement process scheduling and dispatching in programming projects
2. Describe and implement distributed and real-time systems:
   2.1 describe distributed and real-time systems;
   2.2 implement distributed and real-time system problems in programming projects; and
   2.3 document solutions to distributed and real-time system problems in programming projects
Measurable Course Performance Objectives (MPOs) (continued):

3. Explain and use multi-tasking, pre-emptive scheduling and time sharing:
   3.1 define multi-tasking, pre-emptive scheduling, and time sharing;
   3.2 discuss the problems associated with multi-tasking, pre-emptive scheduling, and time sharing;
   3.3 discuss the solutions used by various operating systems to address the problems associated with multi-tasking, pre-emptive scheduling, and time sharing; and
   3.4 implement the solutions used by various operating systems to address the problems associated with multi-tasking, pre-emptive scheduling, and time sharing in programming projects.

4. Describe system security:
   4.1 describe issues of system security; and
   4.2 suggest ways to address issues of system security.

Methods of Instruction: Instruction will consist of lectures, laboratory demonstrations and assignments, and programming examples.

Outcomes Assessment: Exam questions are blueprinted to course objectives. Checklist rubrics are used to evaluate the programming projects for the presence of course objectives. Data is collected and analyzed to determine the level of student performance on these assessment instruments in regards to meeting course objectives. The results of this data analysis are used to guide necessary pedagogical and/or curricular revisions.

Course Requirements: All students are required to:

1. Maintain regular attendance and take part in class discussions.
2. Complete assigned homework and programming projects on time.
3. Take all exams as scheduled.

Methods of Evaluation: Final course grades will be computed as follows:

<table>
<thead>
<tr>
<th>Grading Components</th>
<th>% of final course grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework, class participation and attendance</td>
<td>10%</td>
</tr>
<tr>
<td>Students must practice skills on their own by doing homework to be able to master course objectives. Homework assignments relate to these objectives. Attendance and class participation are necessary for students to benefit from the guidance of the instructor.</td>
<td></td>
</tr>
<tr>
<td>3 or more programming projects</td>
<td>35%</td>
</tr>
<tr>
<td>Programming projects will show evidence of the extent to which students meet course objectives. Students should show that they have synthesized a combination of concepts.</td>
<td></td>
</tr>
</tbody>
</table>
Methods of Evaluation (continued):

<table>
<thead>
<tr>
<th>Grading Components</th>
<th>% of final course grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm Exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>30%</td>
</tr>
</tbody>
</table>

The Midterm Exam will provide evidence of the extent to which students have mastered course objectives and synthesize material taught in the first half of the course.

The Final Exam will provide evidence of the extent to which students have mastered course objectives and synthesize material taught in the second half of the course.

Academic Integrity: Dishonesty disrupts the search for truth that is inherent in the learning process and so devalues the purpose and the mission of the College. Academic dishonesty includes, but is not limited to, the following:

- plagiarism – the failure to acknowledge another writer’s words or ideas or to give proper credit to sources of information;
- cheating – knowingly obtaining or giving unauthorized information on any test/exam or any other academic assignment;
- interference – any interruption of the academic process that prevents others from the proper engagement in learning or teaching; and
- fraud – any act or instance of willful deceit or trickery.

Violations of academic integrity will be dealt with by imposing appropriate sanctions. Sanctions for acts of academic dishonesty could include the resubmission of an assignment, failure of the test/exam, failure in the course, probation, suspension from the College, and even expulsion from the College.

Student Code of Conduct: All students are expected to conduct themselves as responsible and considerate adults who respect the rights of others. Disruptive behavior will not be tolerated. All students are also expected to attend and be on time for all class meetings. No cell phones or similar electronic devices are permitted in class. Please refer to the Essex County College student handbook, *Lifeline*, for more specific information about the College’s Code of Conduct and attendance requirements.
**Course Content Outline:** based on the text *Operating Systems*, 3rd edition, by Deitel, Deitel and Choffnes; published by Pearson/Prentice Hall, 2004

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic/Chapter</th>
</tr>
</thead>
</table>
| 1    | Class Overview – Class syllabus, office hours, policies & requirements; Introduction to Operating Systems – history, hardware & software concepts, operating system environments, operating system components and goals, operating system architectures  
Homework – Exercises 1.1 – 1.11, p 46  
Hardware Concepts – 2.3 Hardware (mainboards, processor components, clocks, memory hierarchy, main memory, secondary storage, buses, direct memory access (DMA) & peripherals), 2.4 Hardware Support for Operating Systems (processor – user mode, kernel mode, privileged instructions, memory protection and management & interrupts and exceptions, timers and clocks, bootstrapping & plug and play), 2.5 Caching and Buffering  
Software Concepts – 2.6 Software (machine language and assembly language, interpreters and compilers, high-level languages, structured programming & object-oriented programming), 2.7 Application Programming Interfaces, 2.8 Compiling, Linking and Loading, 2.9 Firmware, 2.10 Middleware  
Homework – Exercises 2.1 – 2.17, pp 101 – 102 |
| 2    | Process Concepts – 3.1 Definition of Process, 3.2 Process States, 3.3 Process Management (process states and state transitions, process control blocks, process operations, suspend and resume & context switching)  
Process Concepts (continued) – 3.4 Interrupts (interrupt processing & interrupt classes), 3.5 Inter-process Communication (signals & message passing)  
Discuss Basic Tools to Manage Windows XP  
Homework – Exercises 3.1 – 3.16, pp 139 – 141, Windows 2000/XP Treasure Hunt (handout), Display Directory Program (handout) & Text Search Program (handout)  
Windows & Linux Overviews – 21.1 – 21.5 Windows Overview, 20.1 – 20.3 Linux Overview  
Programming Project #1 Utility assigned |
<table>
<thead>
<tr>
<th>Week (3 meetings @ 80 minutes)</th>
<th>Topic/Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Thread Concepts – 4.2 Definition of Thread, 4.4 Thread States, 4.5 Thread Operations, 4.6 User-Level Threads &amp; Kernel-Level Threads&lt;br&gt;Thread Concepts (continued) – 4.9 Linux Threads, 4.10 Windows XP Threads, 20.4 – 20.5 Linux Kernel, Process and Thread Organization, 21.6 Windows XP Process and Thread Management&lt;br&gt;Basic Tools to Manage Linux&lt;br&gt;<strong>Homework</strong> – Exercises 4.1 – 4.4 &amp; 4.9, Linux Treasure Hunt (handout, optional)</td>
</tr>
<tr>
<td>4</td>
<td>Concurrency – 5.2 Mutual Exclusion, 5.6 Semaphores, 7.1 Definition of Deadlock, 7.2 Examples of Deadlock (traffic deadlock, simple resource deadlock, spooling system deadlock &amp; dining philosophers example – handout)&lt;br&gt;Concurrency (continued) – 7.3 Indefinite Postponement, 7.4 Resource Concepts, 7.5 Four Necessary Conditions for Deadlock&lt;br&gt;<strong>Homework</strong> – Traffic Problem (handout)&lt;br&gt;Concurrency (continued) – 7.6 Deadlock Solutions, 7.7 Deadlock Prevention, 7.8 Deadlock Avoidance, 7.9 Deadlock Detection, 7.10 Deadlock Recovery&lt;br&gt;Discuss Basic Tools to Manage Linux&lt;br&gt;<strong>Homework</strong> – Exercises 7.1 – 7.4 &amp; 7.9, p 323&lt;br&gt;<strong>Programming Project #2 Dining Philosophers assigned</strong></td>
</tr>
<tr>
<td>5</td>
<td>Processor Scheduling – 8.2 Processor Scheduling (scheduling levels &amp; preemptive versus non-preemptive scheduling), 8.5 Scheduling Objectives, 8.6 Scheduling Criteria&lt;br&gt;Scheduling Algorithms – 8.7 Scheduling Algorithms (FIFO scheduling, round-robin (RR) scheduling, shortest-process-first (SPF) scheduling, highest-response-ratio-next (HRRN) scheduling &amp; shortest-remaining-time (SRT) scheduling)&lt;br&gt;Scheduling Algorithms – 8.7 Scheduling Algorithms (continued) (multi-level feedback queues &amp; fair share scheduling), 8.8 Deadline Scheduling, 8.9 Real-Time Scheduling&lt;br&gt;Discuss User Management within Windows and Linux&lt;br&gt;<strong>Programming Project #3 Barbershop assigned</strong></td>
</tr>
<tr>
<td>Week (3 meetings @ 80 minutes)</td>
<td>Topic/Chapter</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
| 6                             | Memory Organization and Management – 9.2 Memory Organization, 9.3 Memory Management, 9.4 Memory Hierarchy, 9.5 Memory Management Strategies, 9.7 Single-User Contiguous Memory Allocation, 9.8 Fixed Partition Multiprogramming, 9.9 Variable Partition Multiprogramming  
Virtual Memory Organization – 10.2 Virtual Memory: Basic Concepts, 10.3 Block Mapping, 10.4 Paging, 10.5 Segmentation  
Virtual Memory Management – 11.2 Locality, 11.3 Demand Paging, 11.4 Anticipatory Paging, 11.5 Page Replacement Strategies |
| 7 – 8                          | Midterm Exam Review (handout) |
|                               | **Midterm Exam** |
|                               | Secondary Storage, Files and Databases – 12.3 Moving Head Disk Storage, 12.4 Why Disk Scheduling is Necessary, 12.5 Disk Scheduling Strategies  
Secondary Storage, Files and Databases (continued) – 12.8 Caching and Buffering, 12.9 Disk Performance Techniques, 12.10 RAID  
| 9 – 10                         | 15.6 Multiprocessor Scheduling, 15.8 Load Balancing, 15.9 Multiprocessor Mutual Exclusion |
| 11 – 12                        | Introduction to Networking – 16.2 Network Topology, 16.3 Network Types, 16.4 TCP/IP, 16.9 Client Servers  
Networking Labs |
| 13 – 14                        | Distributed Processing – 17.2 Attributes of Distributed Systems, 17.3 Communications in Distributed Systems, 17.4 Synchronization  
Distributed Processing (continued) – 17.5 Mutual Exclusion, 17.6 Deadlock  
Client/Server – 18.2 Distributed File Systems, 18.3 Multicomputer Systems, 18.5 Peer-to-Peer Distributed Computing |
**Final Exam** |