CTE214 Operating Systems
Syllabus

Course Details
Course Name: CTE214 – Operating Systems
Course Credits: 4
ECTS Credits: 6
Prerequisite: CTE112 (Introduction to Programming II)
Semester: 2011-2012 Fall
Instructor: Serhat M. Azgur
  E-mail: azgur@bilkent.edu.tr
  Room, Phone: GZ-14A, x-3361
Office Hours: Weekdays, 08:30 – 18:00
Teaching Assistant: Engin Z. Kıraçbedel
  E-mail: engink@ctp.bilkent.edu.tr.
  Room, Phone: GZ-13B, x-3225
Lecture hours and place: Wed: 08:40 – 10:30, GB52 - Fri: 10:40 – 12:30, GZ54
Lab hours and place: Wed: 10:40 – 12:30, GB52 (Lab-4)
URL (web pages): http://2011-2012-fall.moodle.bilkent.edu.tr/

Course Description
This course is designed to provide the fundamentals of operating systems and introduction to the
internal operations of modern operating systems as well as practical system-administration tasks
that an ICT employee may need in his/her career.
In particular;
  • Basic operating system concepts and structures.
  • Particular emphasis will be given to the following major OS subsystems:
    o Process management; processes, threads, CPU scheduling, process
      synchronization & deadlocks.
    o Memory management; main memory, virtual memory, paging, swapping.
  • Protection & Security.
  • UNIX environment. Creating Shell scripts with AWK and Perl programming languages.
And, whenever time permits, we briefly examine networking and case studies concerning the
special applications (like MS Windows, Sun Solaris, IBM, SCO, HP/DEC OpenVMS, etc.).

Aim
This course is designed as four hours of lecture, two hours of computer lab applications.
Lab applications are done at the computer lab using desktop computers that are connected to our
department’s server system and thus access to the University network and Internet.
This syllabus (hence, the course) is designed for a typical vocational school student where the goal
is to introduce the basic principles of modern operating systems, the operating system as a control
program and as a resource allocator. What is aimed at:
  • provide coverage of basic computer system organization.
  • describe the services and operating system components, which an operating system
    provides to users, programmers, and other information systems.
  • provide enough details for a software programmer to understand how operating systems
    affect programming efficiency and effectiveness.
• understand and practice lab applications using Linux environment. Successfully write, run and debug shell scripts that can manipulate text-based data, either in files or data streams.

Learning Outcomes

On successful completion of this course:

• Students should apply key operating system design concepts (e.g. processes, threads, paging, etc.) to develop successful software systems.
• Students should understand and discriminate the strengths and weaknesses of scheduling policies, interprocess communication methods and memory management issues in time-sharing systems.
• Students can understand and analyze process deadlocks in computer related applications.
• Students should evaluate the relevant issues that will enable them to make informed judgments about computer information protection and security.
• Students should be able to write, edit and run shell scripts in a Unix environment that can manipulate text-based data, either in files or data streams.
• Having identified the key problems and issues of processing in a multiprocessing environment, students will develop a more efficient and effective software writing style.
• Students should be able to recognize, describe, find and retrieve the professional information that is needed in professional life.

Course Outline:

• Introduction to Operating Systems
• Operating System Concepts
• Operating System Structures
• Process Management
• Interprocess Communications
• Deadlocks
• Threads
• Memory Management, Virtual Memory
• Protection and Security
• UNIX shell programming

Text Book:

Reference Books:
The following books can be consulted for different approaches or further readings (they are all available at the Bilkent University Library):

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>BLISS-Call #</th>
<th>Year &amp; ISBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Systems</td>
<td>William Stallings</td>
<td>QA76.76.O63S733</td>
<td>0-02-415493-8 0-13-180977-6</td>
</tr>
<tr>
<td>Linux İşletim Sistemi</td>
<td>Tr Linux Kull. Grubu</td>
<td>QA76.76.O63L568</td>
<td></td>
</tr>
<tr>
<td>Operating System Concepts</td>
<td>Abraham Silberschatz, Peter Baer Galvin</td>
<td>QA76.76.O63S5583</td>
<td>0-471-69466-5</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>Milenkovic, Milan</td>
<td>QA76.76.O63 M53</td>
<td>0-07-112711-9</td>
</tr>
</tbody>
</table>
Other Resources:
- Lecture notes prepared by the instructor;
- Bilkent Library, electronic databases like Safari Books Online, other web references/resources.

Instructional methods, techniques and tools:
- In class teaching,
- Lab Sessions, Labworks concerning UNIX Shell scripting using Ubuntu operating system.
- Subject related questions-answers and discussions on the web through Moodle,
- Invited guests from the Industry.

Assessments & Grading:
To receive a passing credit, students must achieve at least 50 percent on the course composite grade (= "D"). Although, the assessments and weights may vary from semester to semester, the weighting of the course composite grade for this semester is as follows:

- Term project: 15%
- Labworks: 20%
- Midterm: 25%
- Final: 30%
- In-class participation: 10%

There will be a term project, details will be announced during the semester.

Lab works are done at the computer lab with the help of the teaching assistant and there is at least one lab work that will take place every week. Lab grading will be based on the successful accomplishment of individual lab-works.

Moodle Course Management System is extensively used through out the academic semester. Questions and answers of exams can be found in Moodle together with the other related resources.

Attendance:
According to Bilkent University's regulations attendance is mandatory. In-class Participation, which is 10% of the final grade, is going to be decided on the following factors:

- Lecture attendances (although attendance is compulsory, I take roll calls).
- Active participation in class discussions.
- Reflection letter to be written at the end of the semester, and
- Online participation in Moodle.

Lab attendance is mandatory. If you miss more than three lab sessions (all valid excuses must be endorsed by the Bilkent Health Center and/or Department Chair), you will receive an automatic zero as the Lab grade.

Make-up Policy
If you have missed any one of the assessments (whether an exam or any other type) due to a serious problem, you must inform the instructor immediately, together with the supporting documentation (e.g. hospital report, police report etc.). You may not be able to get a make-up exam or any other type of compensation if your case is not convincing.

Academic Integrity
We trust our students are honorable and honest, so that they do not copy from each other’s work to receive a grade they do not deserve. We believe, each student has a responsibility to understand, accept and comply with the university’s standards of academic conduct as set forth by
the Code of Academic Conduct, as well as policies established by the schools and colleges. Cheating, collusion, misconduct, fabrication, and plagiarism are considered serious offense. “Student Code Of Discipline” is presented in the following web page:
http://www.bilkent.edu.tr/bilkent/admin-unit/hukukm/edisiplin.html

Weekly outline: (based on a semester with 14 full weeks, with a week reserved for midterm)

<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Introduction to Operating Systems</td>
<td>Labwork # 1 Linux environment. Login and access procs. Simple Unix cmds.</td>
</tr>
<tr>
<td></td>
<td><em>Introduction to Operating Systems – 3 Lectures</em></td>
<td>General course outline. Overview of fundamental concepts. Purpose of an OS as a system. How does a computer start?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>Operating System Concepts</td>
<td>Labwork # 2 Linux environment. Linux admin commands</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>Operating System Concepts (cont'd)</td>
<td>Labwork # 3. UNIX shell programming with AWK.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>Operating System Concepts (cont'd)</td>
<td>Labwork # 4 UNIX shell programming with AWK.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>Operating System Structures</td>
<td>Labwork # 5 UNIX shell programming with AWK.</td>
</tr>
<tr>
<td></td>
<td><em>Design and Implementation - 3 Lectures</em></td>
<td>Operating System Services. User-operating system Interface Batch, CLI, GUI, System Calls (types of system calls), System Programs (what are they?), Operating System Structure (Simple, Layered, Modular Approaches), Virtual Machines.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>Process Management</td>
<td>Labwork # 6. UNIX shell programming with AWK.</td>
</tr>
<tr>
<td></td>
<td><em>Processes - 2 Lectures</em></td>
<td>Process concept, process control block (PCB), operations on processes, process scheduling diagrams.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Context Switching - 1 Lecture</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>MIDTERM EXAM</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>Process Management (cont'd)</td>
<td>Labwork # 7 UNIX shell programming with AWK.</td>
</tr>
<tr>
<td></td>
<td><em>Process Scheduling - 3 Lectures</em></td>
<td>Process concept, job and process scheduling. Process scheduling algorithms (FCFS, RR, SJF, SRT, priority scheduling), Queues (multilevel and feedback queues).</td>
</tr>
<tr>
<td>Week</td>
<td>Topic</td>
<td>Lectures</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| 09    | Interprocess Management  
*IPC* - 2 Lectures  
*Process Synchronization* - 1 Lecture  
Semaphores; definition, init, wait, signal operations, use of semaphores to implement process synchronization. Client-Server communication; Sockets. |          | Labwork # 8 UNIX shell programming with Perl |
| 10    | Deadlocks  
*Deadlock Concepts* - 1 Lecture  
Resource deadlock vs. communication deadlock. What is livelock? Shareable, non-shareable vs. Usable, consumable resources. Deadlock conditions and detection. Coffmann Conditions.  
*Methods of Handling Deadlocks* - 2 Lectures  
Deadlock Detection, avoidance, prevention and recovery. Resource Allocation Graph. Banker's Algorithm (Single Source, Multiple Sources). Ostrich algorithm. |          | Labwork # 9 UNIX shell programming with Perl |
| 11    | Threads  
*Threads* - 3 Lectures  
What is a thread? How to create threads? Differences between threads and processes (advantages, disadvantages)? Thread scheduling. Thread Control Block (TCB). Where do we use threads? |          | Labwork # 10 UNIX shell programming with Perl |
| 12    | Memory Management  
*Main Memory* - 3 Lectures  
Memory organization and management, secondary & tertiary memories, storage allocation, fixed/dynamic partitioning, Fragmentation (internal/external fragmentations), swapping, paging. Contiguous memory allocation & reallocation. Memory allocation algorithms (buddy system, first-fit, next-fit, best-fit and worst-fit). |          | Labwork # 11 UNIX shell programming with Perl |
| 13    | Memory Management (cont’d)  
*Virtual Memory* - 3 Lectures  
| 14    | Protection And Security  
*Protection* – 1 Lecture  
*Security* – 2 Lectures  
Security policies. Types of Malware; user, program, system and network threats. Implementing security. Depth of defense. Firewallsing to protect systems and networks. Intrusion detection. |          | Labwork # 13 UNIX shell programming with Perl |