ORT Braude College
Department of Electrical and Electronic Engineering

31261 Operating Systems

Lecturer: Dr. Samy Zafrany
Credits: 3.0
Hours: 2 lecture, 2 laboratory
Grade Composition: 30% - laboratory project, 70% - final exam
Prerequisites: 31230 and 31616

Course Description

Design, implementation, and architecture of operating systems. Historical overview of mainstream operating systems: Unix, Windows, Android. Process synchronization, interprocess communication, processor scheduling, memory management, virtual memory, interrupt handling, device management, I/O, file systems. Concurrent programming, time-sharing systems, threads, deadlocks, semaphors. Hands-on study of Windows and Linux operating system design and usage.

Course Contents

1. Operating System overview: what operating systems do? Computer-System organization & architecture. Operating-system structure and operations. Process management. Memory management. Storage management. Protection and security. Distributed systems.
2. Operating system services. User interfaces. System calls. System programs. Operating system design, implementation, and structure. System boot.
3. Process concept. Process Scheduling. Operations on processes. Cooperating processes. Interprocess communication. Time-shared systems. Communication in client-server systems.
4. Threads Overview. Multithreading models. Threading issues. Linux & Windows threads. Java threads
5. CPU scheduling. Scheduling criteria. Scheduling algorithms. Multiple-Processor scheduling. Real-time scheduling. Thread scheduling. Operating systems examples.
6. Process Synchronization. The critical section problem. Peterson’s solution. Synchronization Hardware. Semaphores. Classic Problems of Synchronization. Monitors. Synchronization Examples. Atomic Transactions. Deadlocks.
7. Memory management. Swapping. Contiguous allocation. Paging. Segmentation.
8. Virtual memory. Demand paging. Process creation. Page replacement. Allocation of frames. Thrashing. Demand segmentation. Operating system examples.
9. File system interface. File concept. Access methods. Directory structure. File-system mounting. File sharing. Protection.
10. File system implementation. File-system structure. Directory implementation. Allocation methods. Free-space management. Efficiency and performance. Recovery. NFS. Mass storage systems.
11. I/O systems. I/O Hardware. Application I/O interfaces. Kernel I/O Subsystem. Transforming I/O Requests to hardware operations. Streams. Performance.

12. The security problem. Program threats. System and network threats. Cryptography as a security tool. User authentication. Implementing security defenses. Firewalling to protect systems and networks. Computer-Security Classifications.
13. Distributed system structures. Network structure. Network topology. Communication structure. Communication protocols. Robustness. Design issues. An example: Networking.
14. Course summary. Real time systems.

Bibliography

1. *Silberschatz and Galvin*. Operating Systems Concepts. 8th edition, 2008, John Wiley & Sons, Inc.
2. *Andrew S. Tanenbaum*. Modern Operating Systems, 3/e. Prentice-Hall 2007
3. *Bovet and Cesati*. Understanding the Linux Kernel. 3rd edition, 2005, O'Reilly.
4. *Robert Love.* Linux Kernel Development. Third Edition , 2010, Addison-Wesley Professional.

Software: Windows 7 and Linux operating systems

Expected Learning Outcomes

Students will be able to understand how modern operating systems work, and how to interact with them efficiently and utilize the available resources optimally.

Last Update: September 08, 2013.