Operating Systems
Spring 2008

Chien Chin Chen

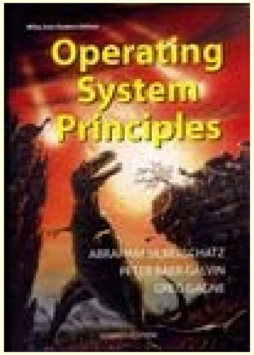
Department of Information Management
National Taiwan University

Glass Information (1/2)

- Instructor:
  - 陳建錦 (Chien Chin Chen)
  - Office: Mgmt II 502.
  - E-mail: paton@im.ntu.edu.tw
- Teaching assistants:
  - 陳政彥, r96725036@ntu.edu.tw
- Class time: 2:20pm ~ 5:20pm, every Tue.
- Location: Mgmt II Room 205.
- Class notes:
  - http://www.im.ntu.edu.tw/~paton/courses/OS_2008spring

Glass Information (2/2)

- Text book:



  - Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne,"Operating System Principles," John Wiley & Sons, 7th edition, 2006.
  - 新月圖書, tel: 2331-7856

- Grading:
  - Attendance: 10%.
  - Homework (at least 4 times): 30%.
  - Midterm: 30%, on 4/15/2008.
  - Final: 30%, on 6/17/2008.

Homework Policy

- Please submit your homework via **hardcopy before the class on the due date**.
- Late homework would not get any credit.
- Students who copy (or copied) other's homework will not get any credit.

Preface (1/8)

- Operating systems are an essential part of any computer system.
  - As an **intermediary** between the computer **user** and the computer **hardware**.
  - Prevent and changing rapidly.
    - Mainframes, PCs, handheld devices, cell phone, …

- This course presents an introduction of operating systems.
  - We will cover the **concepts** and **algorithms** that underlie operating systems.
  - The concepts and algorithms covered in the book are often based on those used in existing commercial operating systems.
  - Particular attention is paid to the Microsoft family of operating systems and various versions of UNIX/Linux.

Preface (2/8)

- The content of the text correspond to the suggestions from *computing curricula 2001* for teaching operating system.
  - Published by the Joint Task Force of the IEEE Computing Society and Association for Computing Machinery (ACM).

Preface (3/8)

- The course will cover the following parts:
  - **Overview**: chapters 1 (introduction) and 2 (system structures) explain:
    - What operating systems are,
    - What they *do*,
    - How they are *designed* and *constructed*.
    - The common features.

    - The presentation is motivational and explanatory to avoid getting into the details of the internal algorithms.

Preface (4/8)

- **Process management**:
  - A **process** is the unit of work in a system.
  - Chapters 3 through 5 (process-concept, multithreaded programming, process scheduling) describe the *process concept* and *concurrency* as the heart of modern operating systems.
  - A computer system consists of a collection of concurrently executing processes.
    - Operating-system process (execute system code).
    - User processes (execute user code).
  - The text also cover methods for *process scheduling* and *interprocess communication*.

Preface (5/8)

- **Process coordination**:
  - Concurrent access to shared data may result in data **inconsistency**.
  - Chapters 6 (synchronization) and 7 (deadlocks) describe various mechanisms to ensure the orderly execution processes so that data consistency is maintained.

- **Memory management**:
  - To improve both the utilization of the CPU and the speed of its response to its users, the computer must keep several processes in memory.
  - Memory size is relative smaller than programs on the disks.
  - Chapters 8 and 9 (memory-management strategies and virtual-memory management) deal with main memory management during the execution of a process.

Preface (6/8)

- **Storage management**:
  - Chapters 10 through 13 (file system, implementing file systems, secondary-storage structure, I/O systems) describe how the *file systems*, *mass storage*, and *I/O* are handled in a modern compute system.
  - The *file system* of operating systems provides the mechanism for on-line *storage* of and *access* to both data and programs residing on the disks.
  - I/O devices are general the slowest major components of the computer.
  - Since the I/O devices that attach to a computer vary widely, the operating system needs to provide a wide range of functionality to applications to allow them to control all aspects of the devices.

Preface (7/8)

- Most program examples are written in C (with some Java).
  - In addition, students should be familiar with basic *data structure*, *computer organization*.

- The example programs require the following programming environments:
  - Windows **Win32 API** (application programming interface), which provides a set of functions for managing processes, threads, memory, and peripheral devices.

  - **POSIX** (Portable Operating System Interface), a set of standards implemented primarily for UNIX-based operating systems.
    - POSIX-compliant operating systems must implement the POSIX core standard (POSIX.1).
      - Linux, Solaris, and Mac OS X.
    - Windows XP and Windows 2000 systems can also run certain POSIX programs.
    - POSIX also defines several extensions to the standards:
      - POSIX1.b – real-time extensions.
      - POSIX1.c – an extension for a thread library.

Preface (8/8)

- **Java**, a widely used programming language with a rich API.
  - Java programs run on any operating system supporting a Java virtual machine (JVM).