CIS 5512 - Operating Systems

Introduction

Professor Qiang Zeng
Fall 2015
About me

• PhD in CSE, Penn State
• Enjoy hacking kernels and compilers
• Industry experiences:
  – IBM T.J. Watson Research Cent
  – NEC Lab America
  – Yahoo
  – Symantec
• Office hours: 3-5pm Thur, SERC 328
• Encourage questions, feedbacks, and comments
You need to know

• Architectures and systems basics
  – CIS 3207 or CIS 5012

• Data structures
  – CIS 3223 or CIS 5011

• C programming
About this course

• *How* the subsystems of an OS are built
  – The beautiful designs behind them
• *Why* they have been built this way
  – What are the trade-offs?
  – Can the ideas be generalized to your research?
• E.g., caching, virtualization, resource allocation
Job hunting

Reference
(Not reference letters)

Online coding (maybe)

Phone interview:
mainly coding

My suggestions:
• Contact schoolmates and friends
• Coding: leetcode, topcoder, careercup
• Stay in the Bay Area
• Learn this course well

Onsite Interviews:
• 3 to 4 Coding interviews
• 1 Behavior Interview
• 1 Systems Design

My suggestions:
• Contact schoolmates and friends
• Coding: leetcode, topcoder, careercup
• Stay in the Bay Area
• Learn this course well
Course materials

- Please check this website frequently for updates of assignments, readings, and slides
- Students are **required** to do the readings ahead of classes
Textbooks

• Required

• Recommended
Grading

• Midterm (35%), Final (35%), Projects (30%)
• **Pair programming** is encouraged
• Cheating will lead to “F”
• Slip days
  – 5 slip days are allowed totally across the projects
  – The README file in your submission should specify

Slip days used: ___ days (this project), ___ days (total)
Why is an operating system needed?

• Services
  – System calls
• Resource allocation
  – Processor, memory, disk
• Protection
  – Isolation and access control
• Inter-process communication (IPC)
  – Signals, pipeline, shared memory, etc.

Think about a bank:
Bank – computer
Staff – operating system
Customers – user programs
  • Services: deposit and withdraw
  • Queues
  • Protect accounts
  • Notify auto dealers of loan approval

Note: Some slides in the following are courtesy of Kwatny, Silberschatz, Galvin, Gagne, Anderson, Dahlin, and Stallings
Three major subsystems

- Process management
  - Processes, threads, synchronization
- Memory management
  - Paging and swapping
- Device management
  - File systems, networks, display
Protection rings

- Introduced by Multics, the predecessor of Unix

- Unix and Windows families on X86 use
  - Ring 0 for kernel; ring 3 for user code

- **Essential difference between user and kernel modes**

- **Why is it needed?**
  - Fault isolation: the crash of a user process can be captured and handled by a lower ring
  - Privileged instructions can only be issued in ring 0, which makes resource management possible; e.g.,
    - I/O: read/write disks, etc.
    - Context switch: change CR3 (to switch page tables)
Questions

• If read/write disks are privileged instructions, how does a user program read/write?
  – Via system calls, e.g.,
    – `fprintf libc call -> write system call -> I/O`

• What if a kernel process/thread crashes?
  – The whole system crashes. Why?
    – Device drivers

• When is Ring 1 used?
  – Some OSes (e.g., OS/2) use Ring 1 for certain I/O
    – In Xen, the hypervisor runs in Ring 0, while the guest OS kernel runs in Ring 1
Virtual Machine

• System-level virtual machine
  – IBM CP-40 (first; 60’)
  – Hardware-level virtualization:
    • needs a VM Monitor (hypervisor)
    • Type 1 hypervisor (runs directly on h/w): Xen, KVM
    • Type 2 hypervisor (runs as a user program): VMWare Fusion
  – OS-level (or., Container): share the kernel but do not share user spaces (e.g., Process IDs and file systems)
  – Emulation: QEMU, Cygwin

• Process-level virtual machine (or, Managed Runtime Environment)
  – Each Java program runs in one JVM; .NET is similar
  – Question: Will the crash of a Java program affect others?
    • No, each Java program runs in its own JVM instance
Hardware-level vs. OS-level virtualization
Questions

• Why is hardware-level virtualization useful?
  – Consolidation: applications previously running on different physical machines now run on different virtual machines (within a single physical machine);
  – Meanwhile, provide acceptable isolation/security
  – Facilitate management: migration, snapshot, recovery

• Pros and cons of Containers?
  – Pros: kernel is shared, so smaller overhead
  – Cons: security is a main concern; applications in all containers are limited to the same type of OS