Introduction

CSE 506: Graduate Operating Systems

Pradipta De
pradipta.de@sunykorea.ac.kr
What is CSE 506 about?

• Graduate Operating Systems
  – To understand OS beyond theory
  – e.g. how memory paging works? … how does it work in Linux, or other OS

• At the end of the course,
  – Good understanding of how OS works
  – Going beyond concepts to delve into OS code
Outline

• Operating System Overview
• OS kernel
  – Booting
  – Program execution
  – How devices work
• Admin and Logistics
  – Lab assignments
  – Exams, Grading, etc.
Operating System: What is it

• Most common view: a software program that manages (and hides complexity) resources and devices from the user
  – In a desktop: manages processor, memory, disk space, network, etc
  – In cellphone: manages much more … battery, screen, dialer and so on
  – In embedded systems: could be managing a specialized piece of hardware

• Collection of code to manage the resources, and libraries to provide interfaces
Types of OS

- Batch OS, Interactive OS

- From perspective of users and tasks:
  - single user, single task: DOS, PalmOS
  - single user, multiple task: Windows Personal Editions
  - Multiple user, multiple task: Linux, Windows Servers

- Real Time OS: RT Linux
<table>
<thead>
<tr>
<th>User Mode</th>
<th>Kernel Mode</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Applications</strong></td>
<td><strong>System call interface</strong></td>
<td><strong>Devices</strong> (keyboard, mouse, memory Controllers)</td>
</tr>
<tr>
<td><strong>Standard Libs</strong></td>
<td><strong>OS Kernel</strong> (memory manager, process manager, filesystem, networking stack)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Kernel interface to hardware (device drivers)</strong></td>
<td></td>
</tr>
</tbody>
</table>
Extend to Virtualization
Assembler, Linker, Loader

- C program (*.c) → Compiler → Assembly lang prog (*.S) → Assembler → Machine lang prog (*.o)
- Library routine/shared libs
- Executable machine lang program (ELF, exe) → Linker → Loader
- Loaded into memory

Program(shell) → execve() → sys_execve() → sys_execve() → do_execve() → search_binary_handler() → load_elf_binary() → start_thread()
Keyboard to Console

- Key pressed and released (inside keyboard)
  - Generates a scancode, and sends a (stream of) byte to the keyboard controller chip
- Scancodes assembled into keycodes
  - Keyboard controller managed by kbd driver
  - Conversion of scan code to keycode
- Keycodes converted to TTY input chars
  - Uses kernel maps (fonts are resolved)
Admin and Logistics
Lab Assignments: JOS

- JOS: A bare-bone OS kernel developed at MIT to help teaching
  - We will be using the 64-bit port of JOS implemented at SBU (Don Porter’s initiative)
  - Each lab hands out skeleton code and you implement one OS block
  - Last lab ends up with a (rudimentary) working kernel, which can boot on a real PC
Important Instructions for Lab

- Lab assignments will be emailed to you
  - Details on the webpage
- Get familiar with GIT
  - Checkpointing your VM can be helpful
    - Play with VMWare Player to understand the features
    - [https://www.vmware.com/products/player](https://www.vmware.com/products/player)
  - MUST NOT post any of the assignments, or the solutions handed out, online or share with others
Exams, Grading

- Midterm: 20
- Final: 30
- Lab1: 10 (use this grade as a bonus)
- Lab2: 10
- Lab3: 10
- Lab4: 10
- Lab5: 10
- Lab6: 10

- Late submission:
  - Check with instructor

- Course Schedule
  - Reading list: more pointers to what is presented in class
  - Lecture slides: what is relevant for this course
Other resource

• Please send emails with a subject prefix
  – [CSE 506] <subject>
  – Copy everyone in class if it is a common question
  – Treat this as our discussion group