

CS3214: Computer Systems

Fall 2022

Introduction

Huaicheng Li

Aug 23 2022

Resources

- ❑ Course website: <https://courses.cs.vt.edu/cs3214/fall2022>
- ❑ Syllabus: <https://courses.cs.vt.edu/cs3214/fall2022/documents/CS3214-Syllabus-Fall22.pdf>
- ❑ Course Forum: <https://cs3214.cs.vt.edu>
 - Use your @vt.edu email to sign up (ASAP)
- ❑ Discord
- ❑ Email: cs3214-staff@cs.vt.edu
- ❑ Staff: <https://courses.cs.vt.edu/cs3214/fall2022/staff>
 - Office hours (in person or zoom)
- ❑ **Canvas not used**

Format

❑ Lectures

- TR 2:00pm-3:15pm Surge 104C
- Attendance (optional), class meetings not recorded

❑ Midterm + Final

❑ Projects and exercises

❑ Grading

14%	Midterm Exam
16%	Final Exam
40%	Projects
24%	Exercises
4%	Participation
2%	Syllabus Quiz

Project Groups

- ❑ 2 students per group
- ❑ Contribute equally
 - We won't micromanage you, up to you to split the work
- ❑ Changing group between projects allowed
- ❑ Team up across class sections allowed
- ❑ Late policies:
 - 4 late days without penalty (for projects and exercises)
 - Accommodations, case by case (University accommodations / sickness)
- ❑ Best practices
 - Start early!
 - Allocate enough time
 - Hard but rewarding!

Honor Code

- ❑ “As a Hokie, I will conduct myself with honor and integrity at all times. I will not lie, cheat, or steal, nor will I accept the actions of those who do.”
- ❑ Will be strictly enforced
 - Case will be directly filed to the university without warning you
- ❑ Do NOT cheat: details in syllabus, always acknowledge the sources
- ❑ We have **very** powerful cheating detection tools
- ❑ Default penalty
 - *1st offense: F**
 - *2nd offense: Expulsion*
- ❑ Ask if in doubt!

Prerequisites

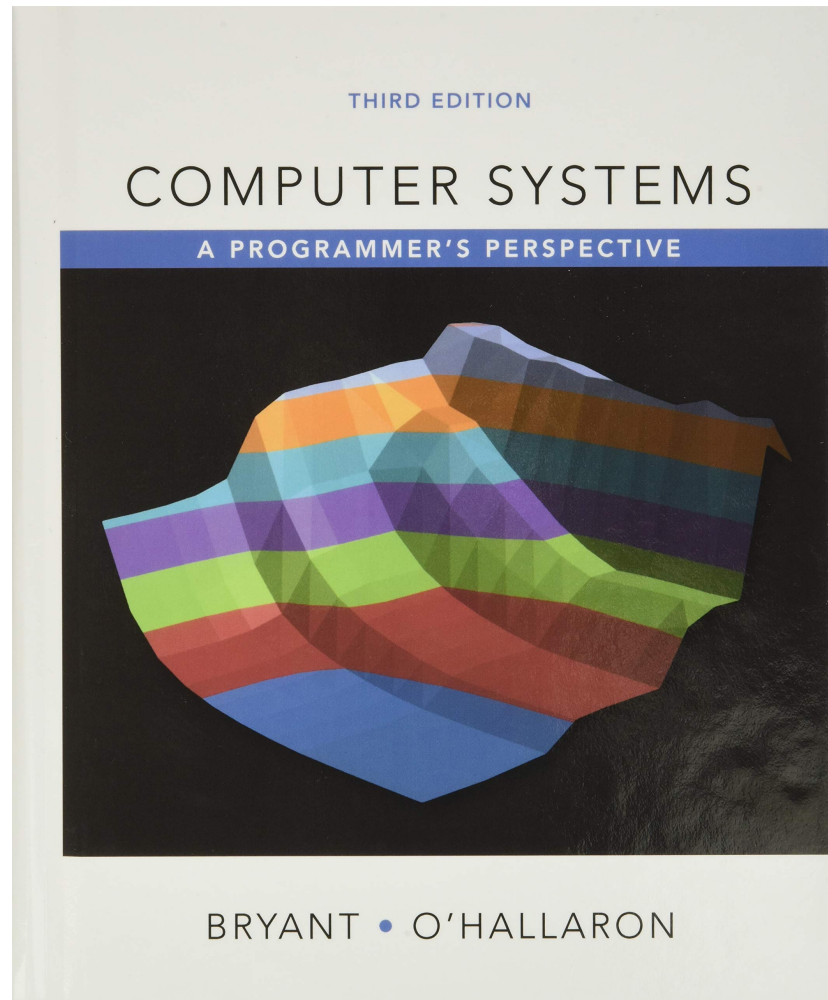
- ❑ CS 2506: Introduction to Computer Organization II (*Grade C or better*)
- ❑ CS 2114: Software Design and Data Structures (*Grade C or better*)

Textbook

- CSAPP3e
 - **User/programmer perspective**
 - Basic understand about how systems work
 - Serve as a primer for deeper Systems course

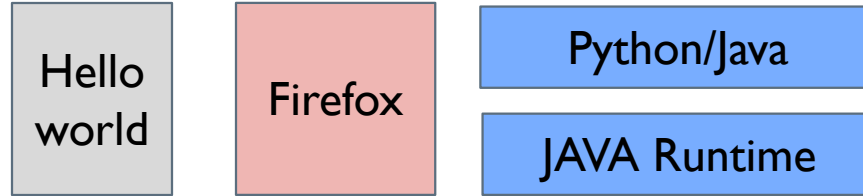
- Learning objectives
 - How does OS work?
 - How to interact with an OS via shell cmd?
 - How to write better programs by leveraging OS APIs?

- *Optional (free) textbook:*
 - <https://pages.cs.wisc.edu/~remzi/OSTEP/>



What is an Operating System?

Applications



- ❑ Software *layer(s)* sitting between applications and hardware

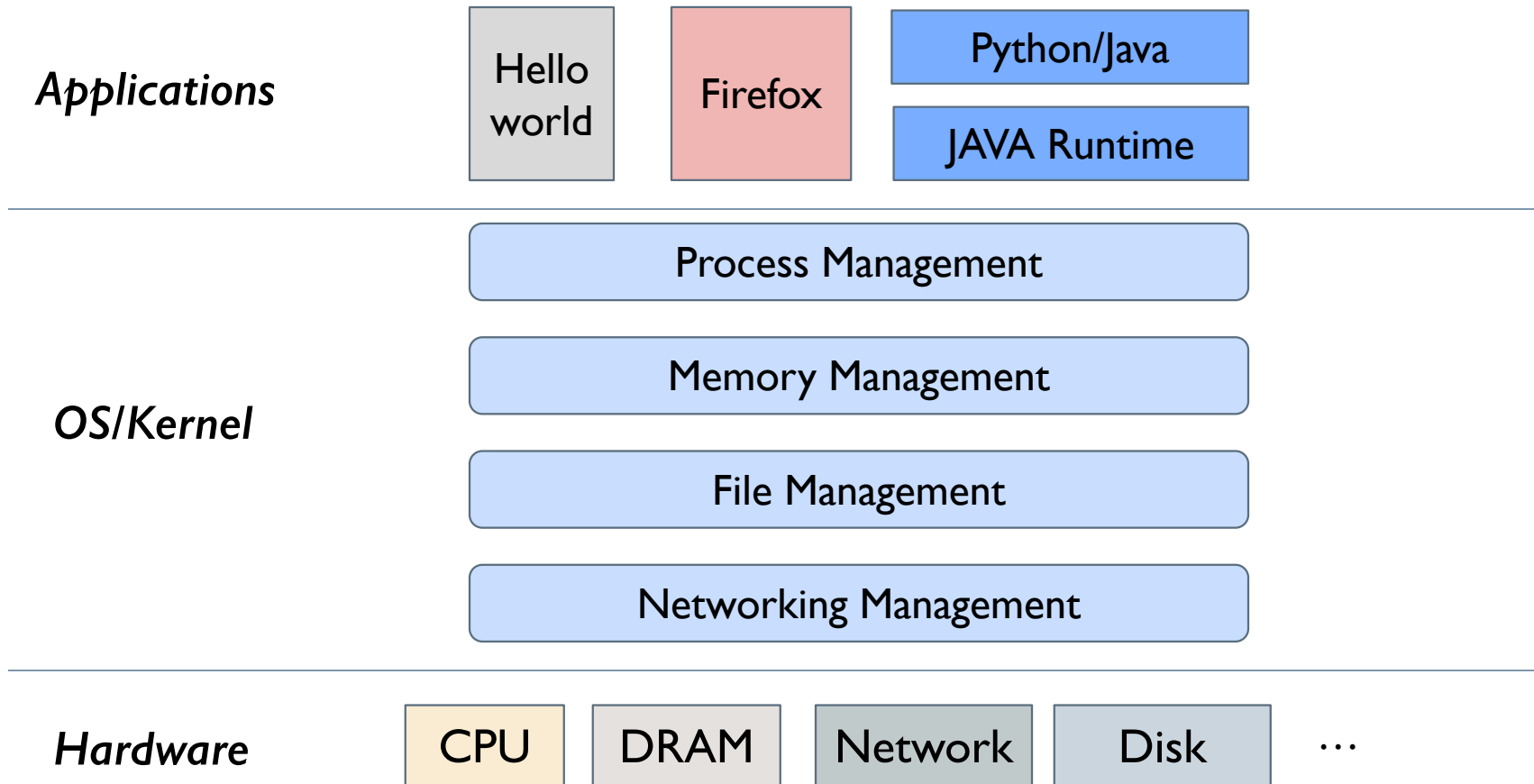
- ❑ Resource management
 - Virtualization / scheduling

OS/Kernel

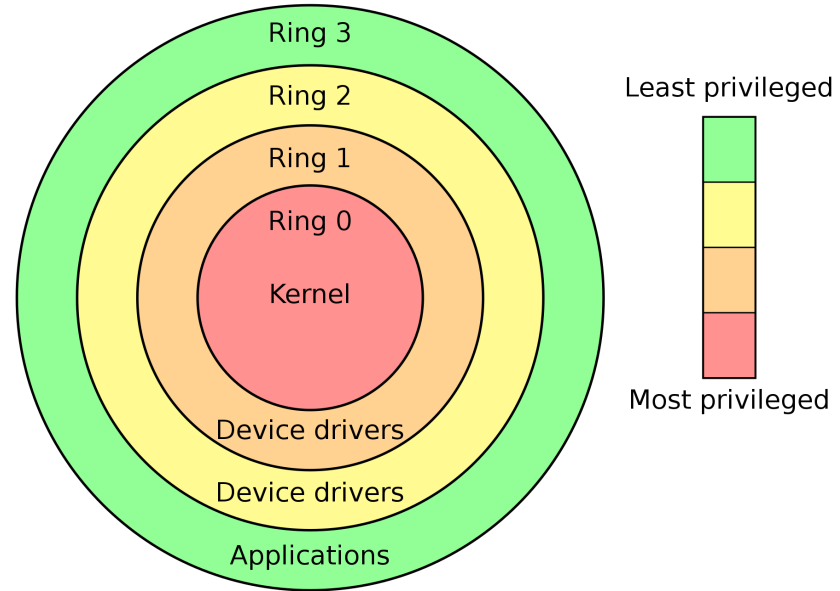
- ❑ Protection
 - Preemption (arbitration who has the right to use hardware resources)
 - Privilege levels (user mode vs. kernel mode)
- ❑ *Abstraction* (processes, memory manager, socket/file, etc.)

Hardware



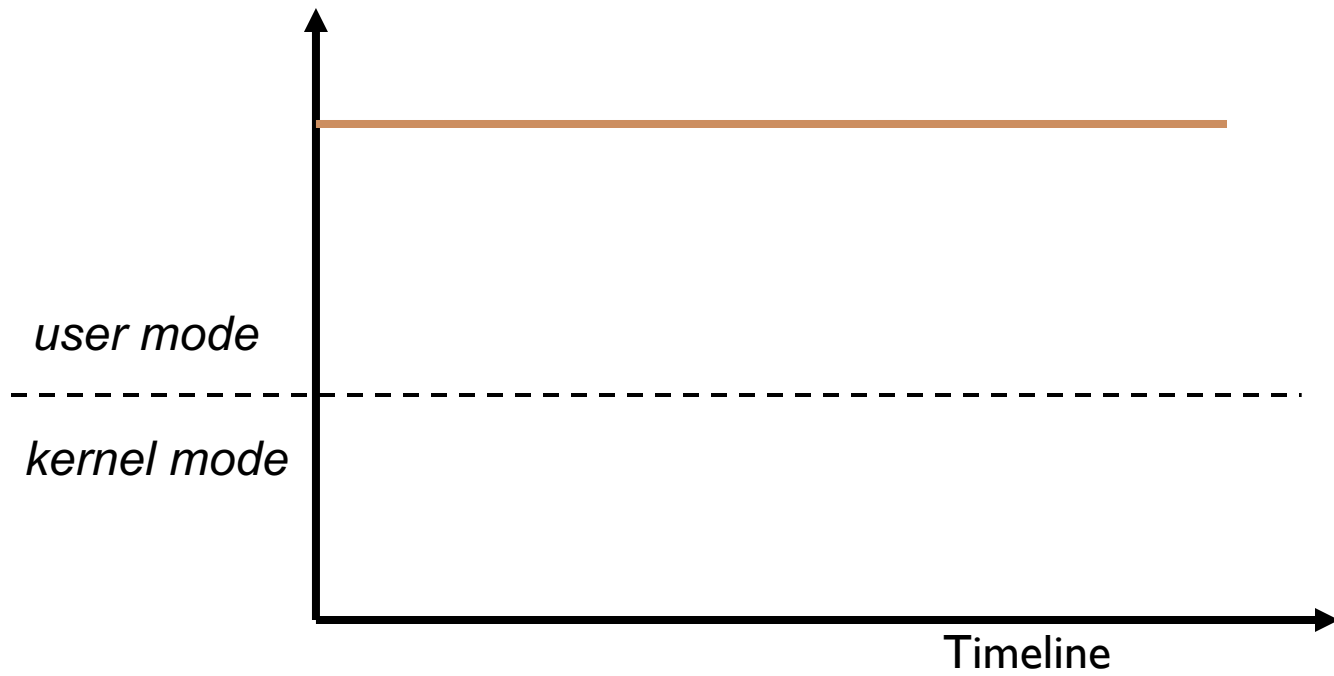


- ❑ Instruction set
- ❑ CPU privilege levels
- ❑ OS
 - User mode
 - Kernel mode
- ❑ **Context switch:**
 - User mode <-> Kernel mode
 - System calls



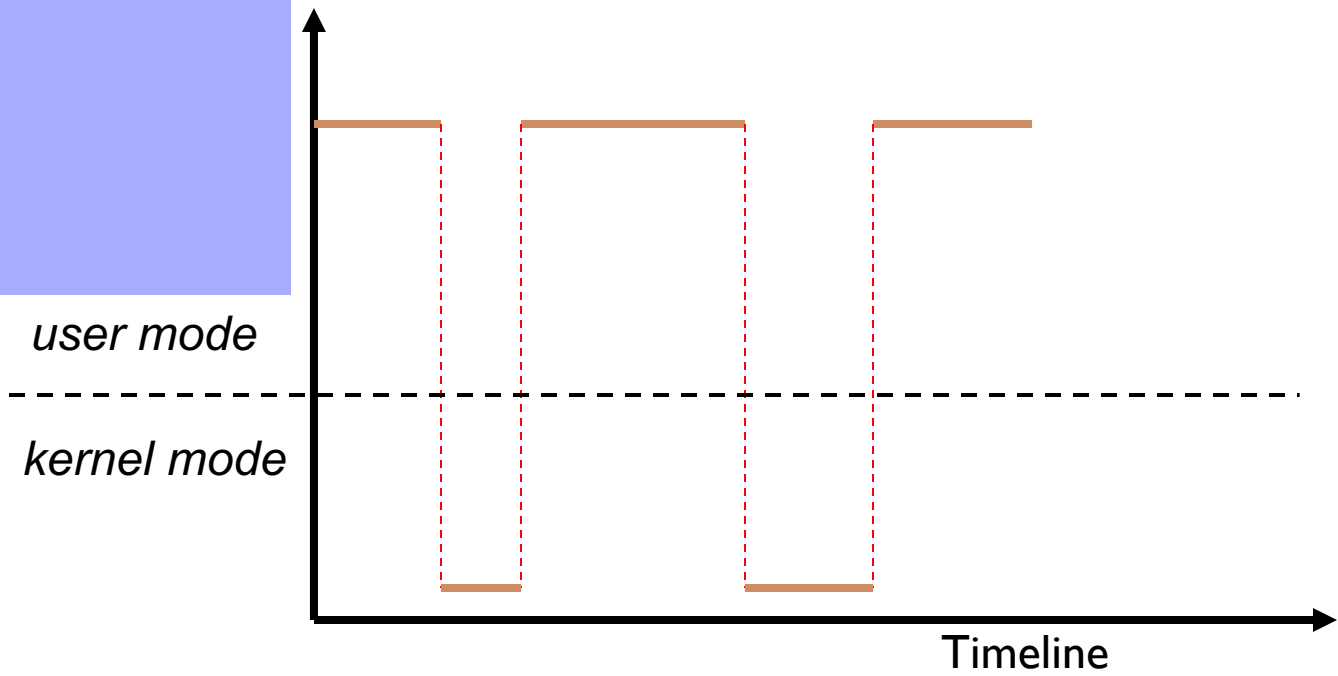
A Simplified Example

```
int main()  
{  
  while (1) {  
    ;  
  }  
}
```



A Simplified Example

```
int main()
{
    char buf[1024];
    int fd = open("/foo/bar", O_RDWR);
    while (1) {
        ;
        read(fd, buf, 1024);
    }
}
```



Topics of this Course at a High Level

- ❑ Processes
 - Multi-core / Multi-processing
 - Process lifecycle management
 - Process communication (signals, inter-process communication)
- ❑ Threads
 - Concurrency / Synchronization concepts and APIs
- ❑ Memory
 - Virtual memory management
 - Linking and loading process
 - Binary file
 - Shared memory
- ❑ Network
 - Socket APIs / HTTP protocol / event-driven programming

More on Processes on Thursday!