CS 344: Operating Systems I

01.09: Introduction to Operating Systems I

Mon/Wed 12:00 – 1:50 PM (LINC 200)

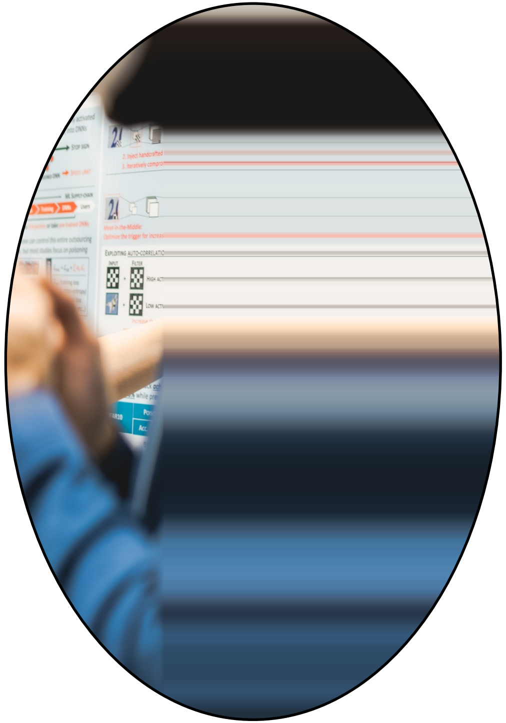
Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab

INSTRUCTOR: SANGHYUN HONG



Who am I?

- **2021 - Now:** Assistant Professor of Computer Science at OSU
- **2021:** Ph.D. from the University of Maryland, College Park
- **2015:** B.S. from Seoul National University, South Korea

What I do?

- **Formal:** I work at the intersection of security, privacy, and machine learning
- **Informal:** I "hack" machine learning, expose security threats and defeat them

What do I teach?

- CS 344: Operating Systems I
- CS 370: Introduction to Security
- CS 499: Machine Learning Security (> Trustworthy Machine Learning)

Where can you find me?

- **Office:** #4103 Kelley Engineering Center (KEC)
- **Email:** sanghyun.hong (at) oregonstate.edu | Discord Server

Topics for today

- Course overview
  - Prerequisites
  - Course information (time, location, teams, office hours, …)
  - Course structure
  - Tips: how to be successful
- Introduction
  - What is an OS?
  - Why do we study OS?
  - Why do we think studying OS difficult?
  - How has OS been developed?
  - What are the functionalities of OS?
  - What are the course topics?

Prerequisites

- Courses:
  - CS 261: Data Structures (or similar)
- Skills:
  - Good “problem-solving skills”
  - Some familiarity in C and Bash shell script languages
- Others:
  - PC or a laptop where you can access the OS1 Server for the assignments

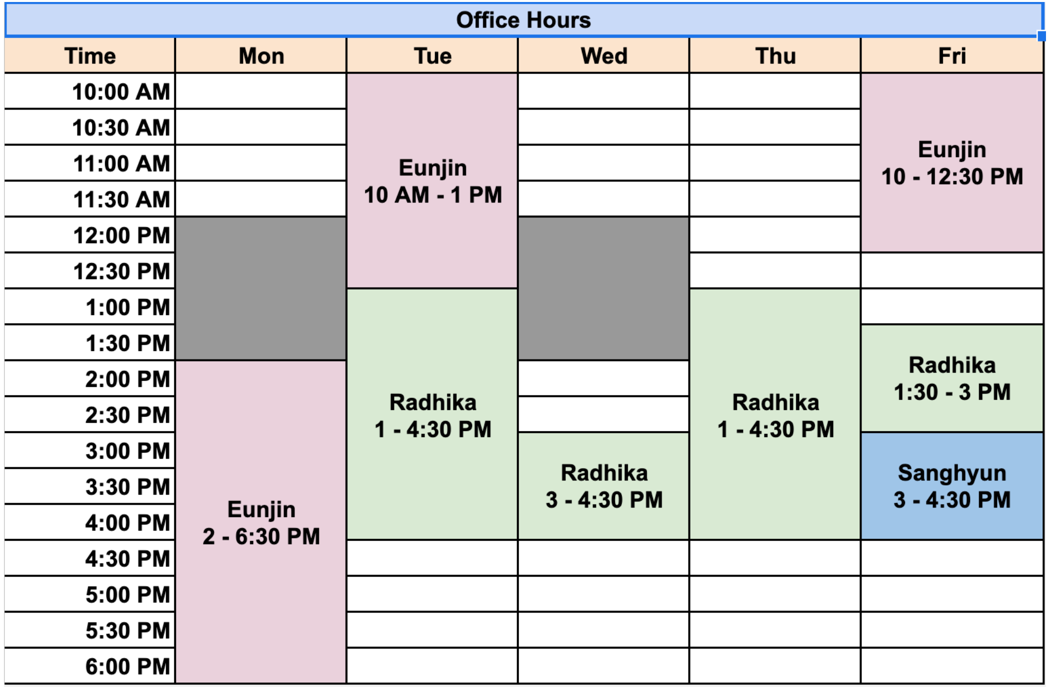
Course information

- Time and location
  - **Time:** 12– 1:50 PM PST (M/W)
  - **Classroom:** #200 LINC or Zoom (No recordings)

- Contacts
  - **SH:** sanghyun.hong@oregonstate.edu
  - **TA:** Radhika Gupte (gupter@oregonstate.edu)
    Eunjin Roh (rohe@oregonstate.edu)
  - **Online discussion:** Discord server (see Canvas for the joining link)

Course information

- Office hours
  - **Location:** on Zoom (see Canvas for the links)
  - **Time:** More than 20 hrs / week
  - **Best practice:**
    - Q's for the assignments - TA
    - Q's for the others – SH



Office Hours

Time	Mon	Tue	Wed	Thu	Fri
10:00 AM		Eunjin 10 AM - 1 PM			Eunjin 10 - 12:30 PM
10:30 AM					
11:00 AM					
11:30 AM					
12:00 PM					
12:30 PM					
1:00 PM		Radhika 1 - 4:30 PM		Radhika 1 - 4:30 PM	
1:30 PM					Radhika 1:30 - 3 PM
2:00 PM	Eunjin 2 - 6:30 PM				
2:30 PM					
3:00 PM			Radhika 3 - 4:30 PM		Sanghyun 3 - 4:30 PM
3:30 PM					
4:00 PM					
4:30 PM					
5:00 PM					
5:30 PM					
6:00 PM					

Course structure

- Tasks
  - 4 Midterm quizzes
  - 5 Programming assignments (reduced from 6 > 5)
  - 9+ Extra credit opportunities

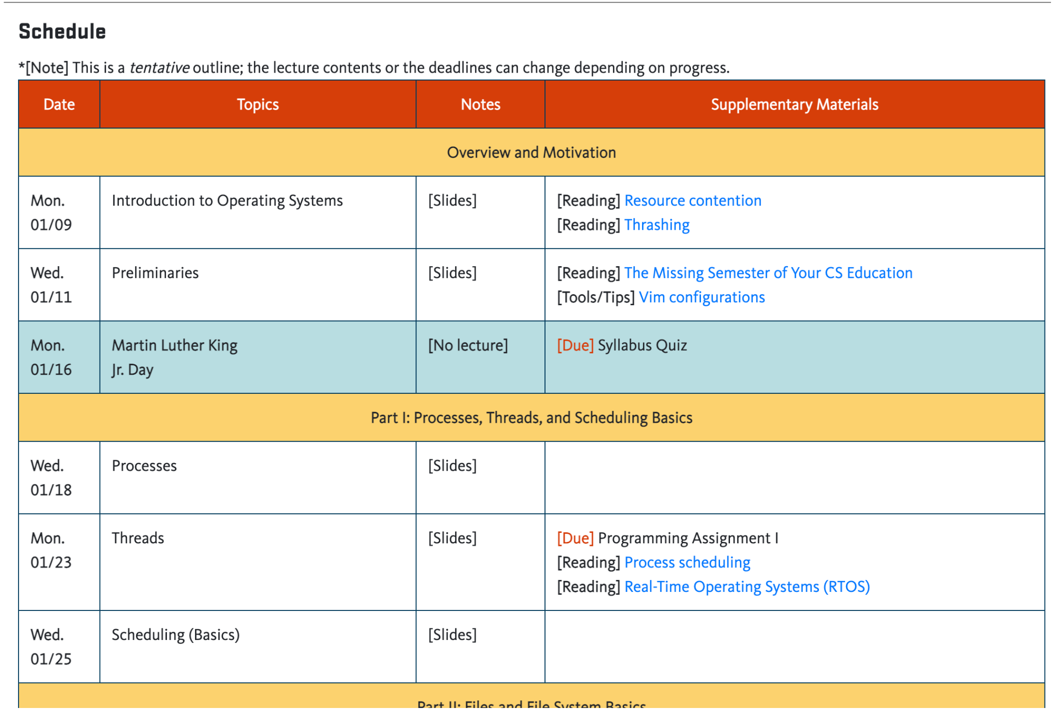
- Grading
  - 60%: 5 Programming assignments
  - 40%: 4 Midterm quizzes
  - 20%+: Extra credit opportunities (on top of the 100% from the above two)

Course structure

- Grading policy
  - 4 Midterm quizzes
    - **Period:** in every 2-3 weeks, you will have an online quiz (on Canvas)
    - **Method:** You will have 3 times to take each quiz
    - **Timed exam:** 80-120 min
  - 5 Programming assignments
    - **Period:** in every 1-3 weeks, you will have a programming assignment (on Canvas)
    - **Submission penalty:**
      - 0% penalty if you submit the assignment on time
      - 5% penalty for every 24 hours
      - 50% maximum penalty if you submit until the 22nd of March
      - 100% penalty if you miss these deadline without any note
  - 9+ Extra credit opportunities
    - **Ad-hoc opportunities:** up to SH

Course structure

- Tentative schedule
  - See: https://secure-ai.systems/courses/OS1/W23/syllabus.html



Schedule

*[Note] This is a *tentative* outline; the lecture contents or the deadlines can change depending on progress.

Date	Topics	Notes	Supplementary Materials
Overview and Motivation			
Mon. 01/09	Introduction to Operating Systems	[Slides]	[Reading] Resource contention [Reading] Thrashing
Wed. 01/11	Preliminaries	[Slides]	[Reading] The Missing Semester of Your CS Education [Tools/Tips] Vim configurations
Mon. 01/16	Martin Luther King Jr. Day	[No lecture]	[Due] Syllabus Quiz
Part I: Processes, Threads, and Scheduling Basics			
Wed. 01/18	Processes	[Slides]	
Mon. 01/23	Threads	[Slides]	[Due] Programming Assignment I [Reading] Process scheduling [Reading] Real-Time Operating Systems (RTOS)
Wed. 01/25	Scheduling (Basics)	[Slides]	
Part II: Files and File System Basics			

Tips: how to be successful

- Rules:
  - **Dont’s**
    - Do not share your code with others
    - Do not copy and paste someone else’s code in yours
    - Do not cheat in online quizzes
    - Do not ask for the solutions on the Internet, *e.g.*, StackOverflow
  - **Do’s**
    - Brainstorm ideas
    - Discuss basic concepts on Discord
    - Help someone else debug if they run into a technical wall
- Tips:
  - **No. 1:** Start programming assignment early
  - **No. 2:** Come to classes and office hours

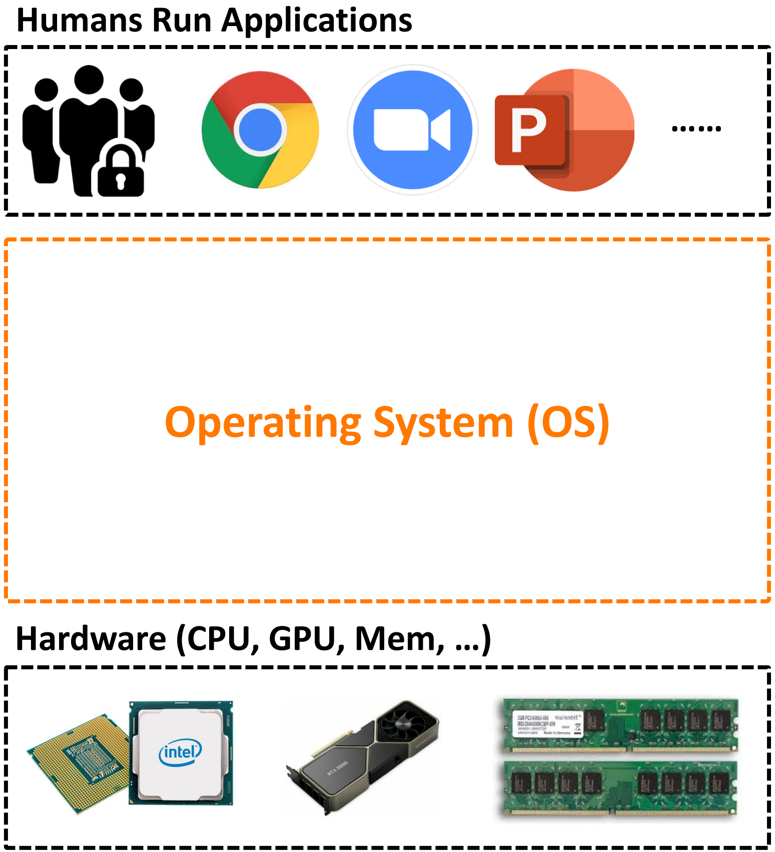
Topics for today

- Course overview
  - Prerequisites
  - Course information (time, location, teams, office hours, …)
  - Course structure
  - Tips: how to be successful

- Introduction
  - What is an OS?
  - Why do we study OS?
  - Why do we think studying OS difficult?
  - How has OS been developed?
  - What are the functionalities of OS?
  - What are the course topics?

What is an operating system?

- **Definition**
  - Computer software that lies between hardware and applications



Humans Run Applications

……

Operating System (OS)

Hardware (CPU, GPU, Mem, …)

Why do we study OS?

- Many reasons:
  - To graduate / to get a job
  - To make new hardware and use it
  - To make existing OS (*e.g.*, iOS and Linux) better
  - To understand computer systems better
  - To make your software more secure and private (me…?!)
  - Just for fun!

- **Mine:**
  - OS is an exciting field of study
  - OS brings many areas in computer science altogether
    - Data structure, algorithms
    - Programming languages, compilers
    - Computer hardware, architecture

Why do we think OS is difficult to study?

- Many reasons:
  - OS requires to understand many computer science areas
  - OS is large (10M+ lines of code; 1000+ man-years of work)
  - OS is complex (many different behaviors; hardware supports; goals)
  - OS is poorly understood (only few completely understood Linux internals)
  - … **A false sense of difficulty!**

My Guess: We Are Not Familiar to How Computers Think!

Why do we think OS is difficult to study?

- Study tips for CS 344:
  - Focus on problems: OS is a history of “problem-solving”
  - Memorize definitions: we name things with meanings
  - Understand solutions and their limits
  - Do hands-on exercises: write your own *small* programs

How has OS been evolved?

- Why do we care about this history?
  - Defining OS *precisely* is difficult
  - Motivation:
    - **To run computer programs**
    - OS has been evolved to solve “a set of problems”
    - Problems: new hardware, faster execution, multi-program supports, etc…

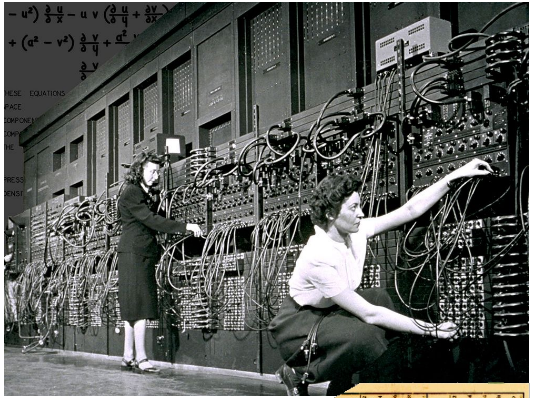
- Three Phases of OS History
  - Phase I: early 50s – mid 60s
  - Phase II: mid 60s – mid 90s
  - Phase III: mid 90s – Present
  - Phase IV: It’s your job!

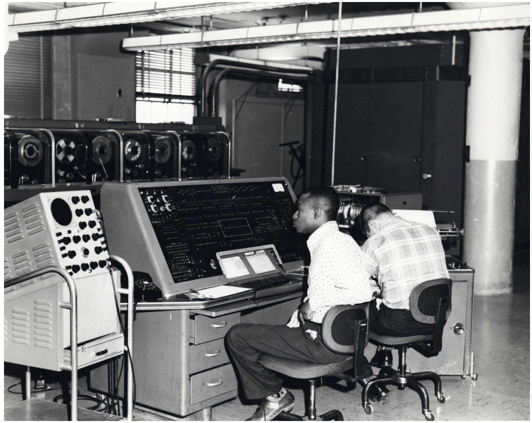
Phase I: Early 50s – Mid 60s

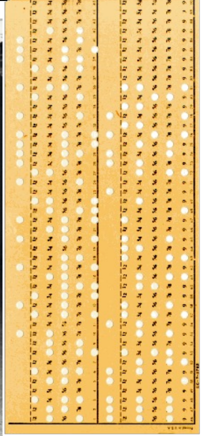
- Motivation
  - Hardware was expensive; humans are cheap

- Design objectives
  - Make efficient use of the hardware
  - Increase CPU utilization (no idle time)

- Phase I-I: Human Operator as OS
  - OS (Human) is a shared subroutine (function)
  - Only one user can run a job at a time
  - Job to job transition is slow (human…)

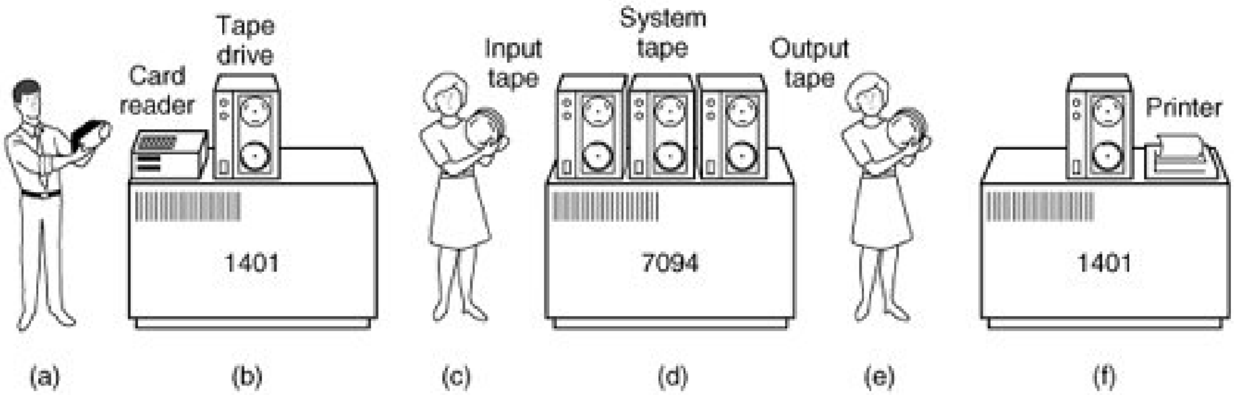






Phase I: Early 50s – Mid 60s

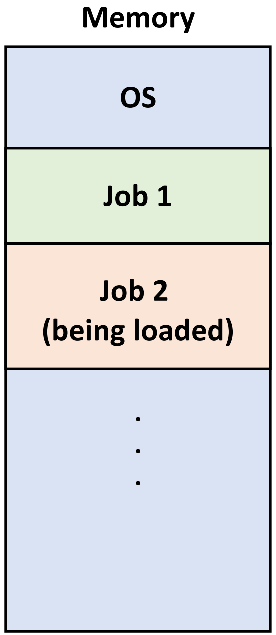
- Phase I-II: *Simple* Batch Monitor
  - OS loaded: run user jobs and take dumps
  - IBM machines
    - IBM 1401 (I/O Machine) reads cards in batch onto tape
    - IBM 7904 (Main Machine) computes and dumps results back to the tape
    - I/O Machine prints outputs from the tape



Phase I: Early 50s – Mid 60s

- Problem of Simple Batch Monitors
  - CPU in idle when the machines do I/O
  - CPU stops if a job in a tape has errors

- Phase I-III: Batch Monitor
  - **Solution:** let's make CPUs run while the machine is on I/O (overlap!)
  - **Interrupt (asynchronous)** mechanism:
    - Machine first reads a tape
    - Machine runs jobs in the tape and starts reading another tape
    - If the tape reading is done, the machine sets "ready" flag.
    - Do iteratively…


Memory

OS

Job 1

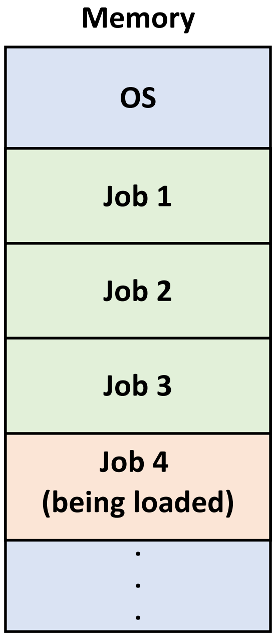
Job 2
(being loaded)

.
.
.

Phase I: Early 50s – Mid 60s

- Problem of Batch Monitors
  - Run only a single job at a time
  - What if a job does some I/O? All other jobs need to wait…

- Phase I-IV: *Multi-programmed* Batch Monitor
  - **Solution:**
    - Let’s make an OS run/manages multiple jobs
    - *e.g.*, load a new job to CPUs until the I/O is done
  - OS became a focus of study
    - How can we *protect* multiple jobs in mem?
    - How can we manage multiple jobs in mem with a *smart* way?
    - How can we *write* programs that maximize the CPU usage?
  - Memory protection; relocation; concurrency…



Memory

OS

Job 1

Job 2

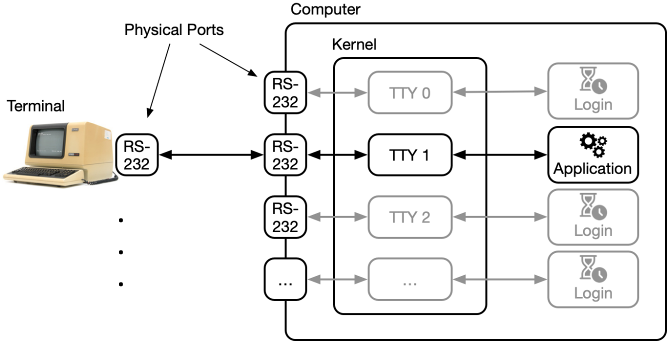
Job 3

Job 4
(being loaded)

.
.
.

Phase II: Mid 60s – Mid 90s

- Motivation
  - Hardware became cheap; humans are expensive
- Design objectives
  - Make efficient use of human’s time
  - Reduce the idle time of humans
- Phase II-I: Interactive Time-sharing OS
  - **Solution:** Give terminals (cheap) to users
  - **Problems:**
    - CPU Time should be sliced …
    - Not all data can be accessed by everyone …
    - New metrics for evaluating OS …



Phase III: Mid 90s – Present

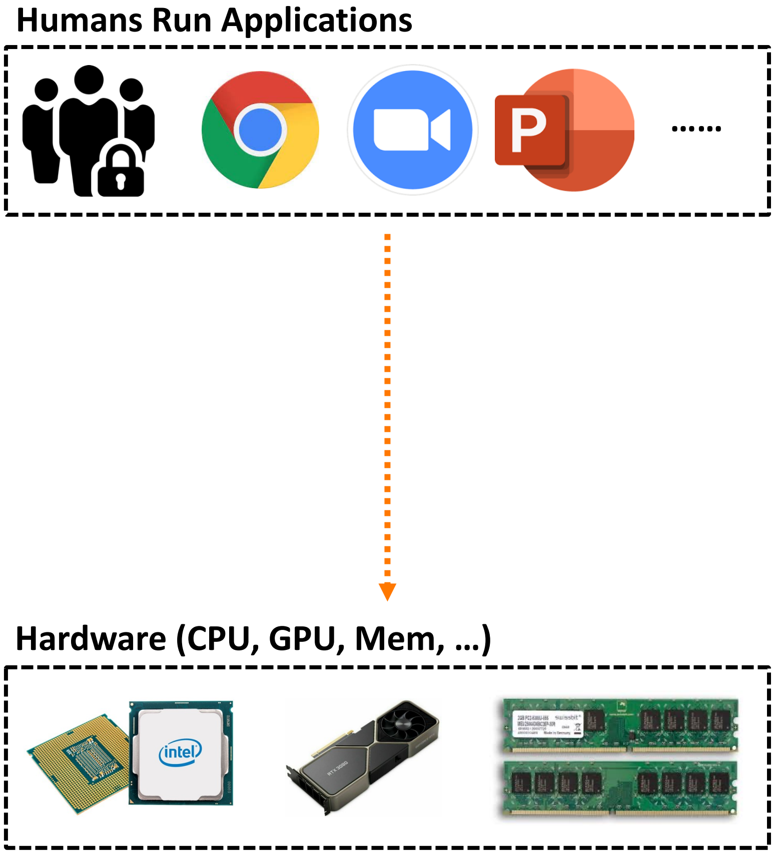
- Motivation
  - Computers became connected to each other
- Design objectives
  - Offer connected multimedia services for users
- Phase III-I: OS Built with Connectivity
  - Internet *protocols* added to PC OS
  - Internet (network) programming became important (Web, Python, …)
  - Multi-tasking became much more important

Phase III: Mid 90s – Present (cont'd)

- Phase III-II: Complex PC OS
  - Computers became extremely cheap
  - Computers were equipped with sophisticated H/W architecture
  - Modern OS should be complex too to support H/W…
- Phase III-III: OS with Multimedia Support
  - Increasing demands of computer and network resources
  - Human perception became the center of universe
    - QoS (Quality of Services)
    - RTOS (Real-time OS)
  - Home appliances and computers became merged (IoT devices…)

What does OS do?

- Functionalities of Modern OS
  - Manage resources
  - Provide abstractions
  - Offer standard interfaces

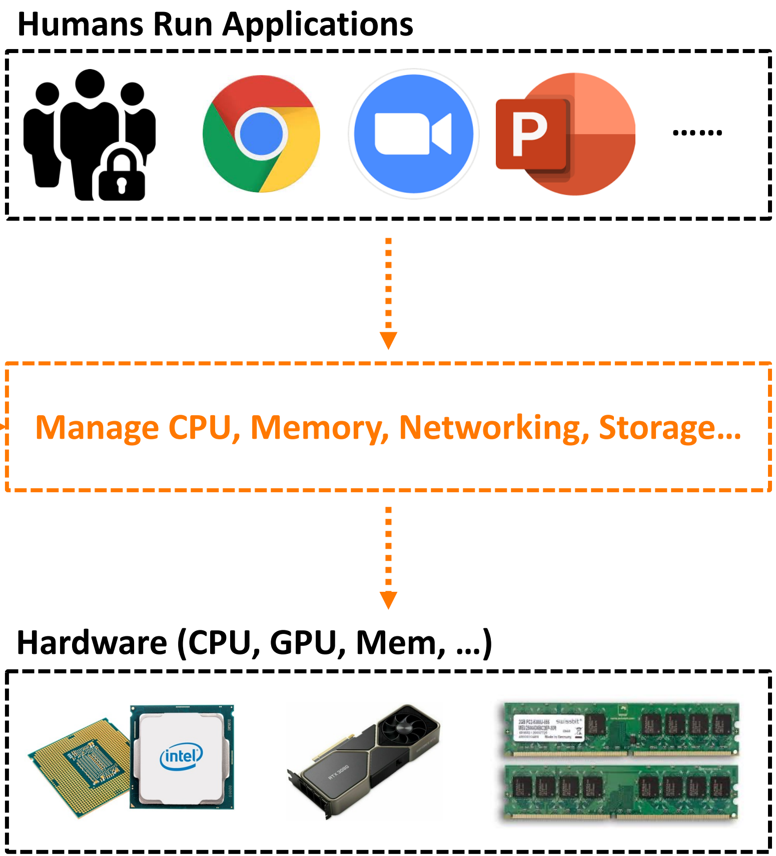


Humans Run Applications

Hardware (CPU, GPU, Mem, …)

What does OS do?

- Functionalities of Modern OS
  - **Manage resources**
  - Provide abstractions
  - Offer standard interfaces



Humans Run Applications

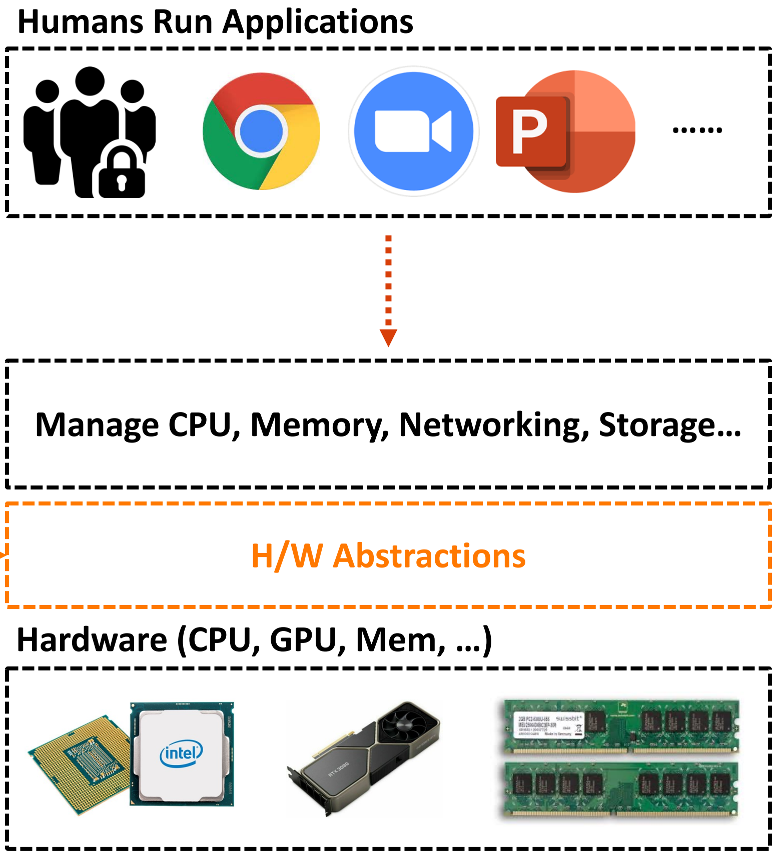
......

Manage CPU, Memory, Networking, Storage…

Hardware (CPU, GPU, Mem, …)

What Do OS Do?

- Functionalities of Modern OS
  - Manage resources
  - **Provide abstractions**
  - Offer standard interfaces



Humans Run Applications

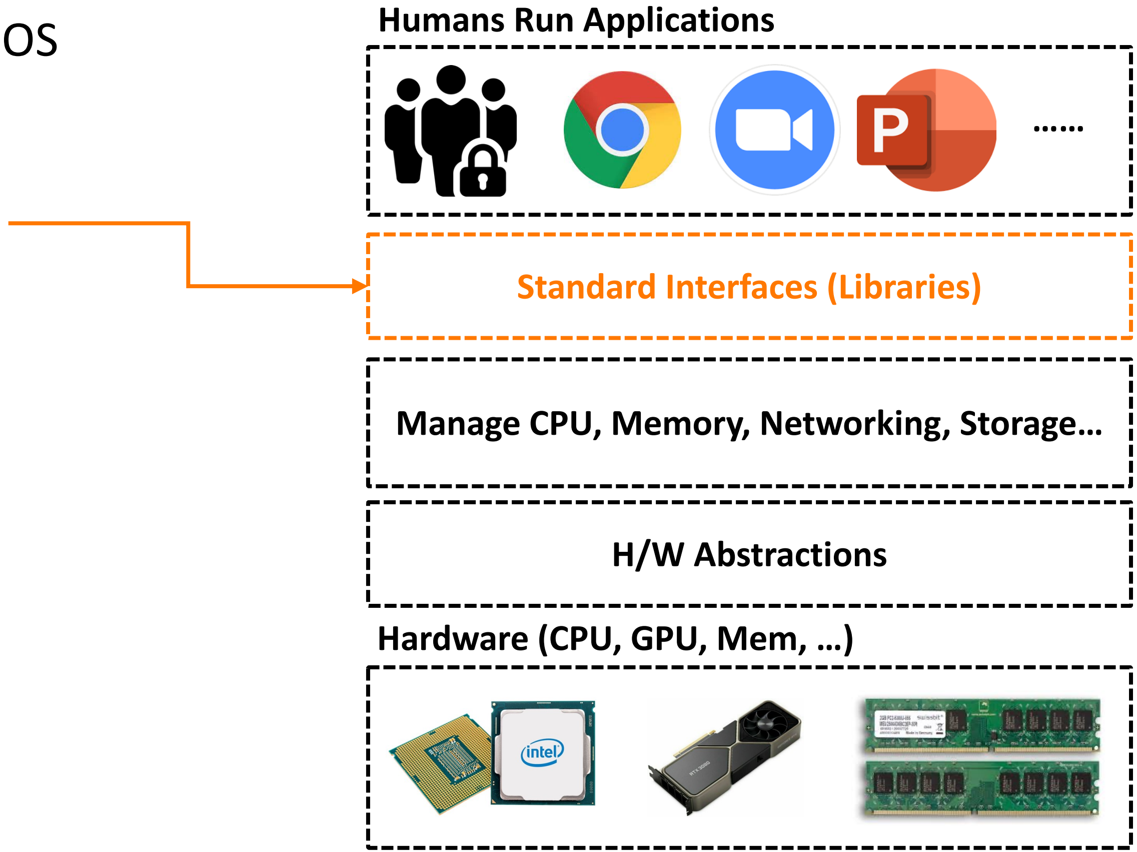
Manage CPU, Memory, Networking, Storage…

H/W Abstractions

Hardware (CPU, GPU, Mem, …)

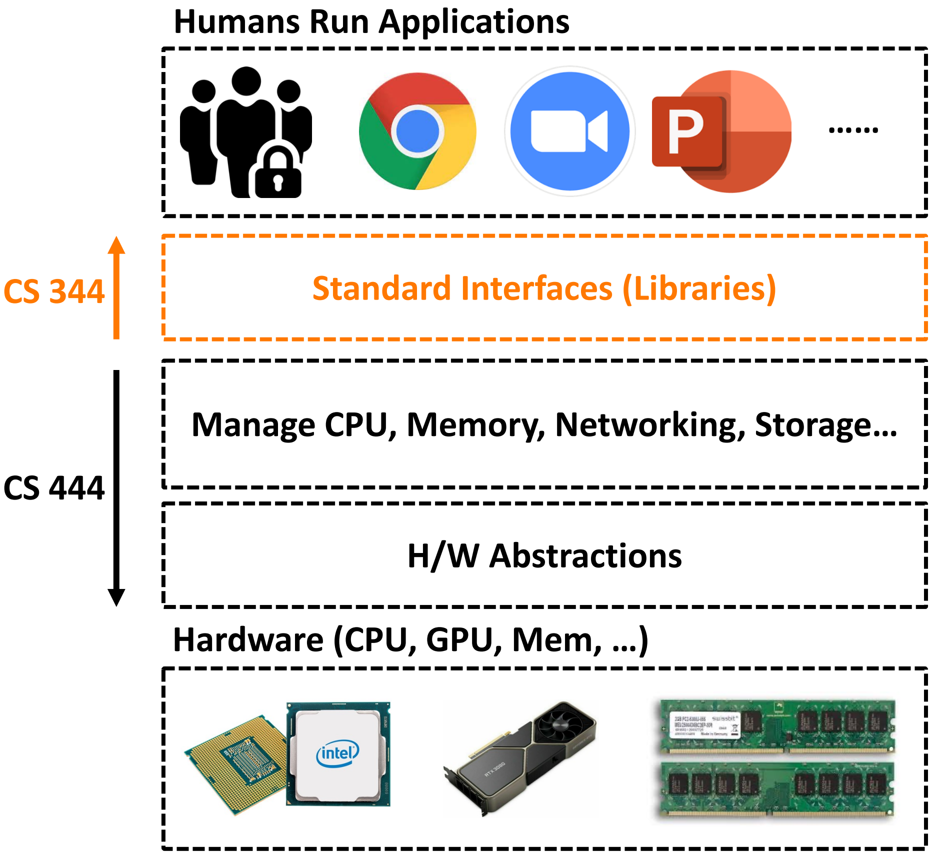
What Do OS Do?

- Functionalities of Modern OS
  - Manage resources
  - Provide abstractions
  - **Offer standard interfaces**


Humans Run Applications
……
Standard Interfaces (Libraries)
Manage CPU, Memory, Networking, Storage…
H/W Abstractions
Hardware (CPU, GPU, Mem, …)
intel

What are the Course Topics?

- Functionalities of Modern OS
  - Manage resources
  - Provide abstractions
  - Offer standard interfaces
- What do we learn in CS 344?
  - Standard interfaces
    - Shell (Bash), Terminal
    - C language
    - Files and directories
    - Processes (and threads)
    - Inter-process communication (IPC)
    - Networking
    - (Secure OS) Rust language



Humans Run Applications

CS 344

Standard Interfaces (Libraries)

CS 444

Manage CPU, Memory, Networking, Storage…

H/W Abstractions

Hardware (CPU, GPU, Mem, …)

Topics for today

- Course overview
  - Prerequisites
  - Course information (time, location, teams, office hours, …)
  - Course structure
  - Tips: how to be successful

- Introduction
  - What is an OS?
  - Why do we study OS?
  - Why do we think studying OS difficult?
  - How has OS been developed?
  - What are the functionalities of OS?
  - What are the course topics?

Thank You!

Mon/Wed 12:00 – 1:50 PM (LINC 200)

Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab