

## ► What is Systems Programming?

First an Analogy: Driving a car

### ■ End user

- Able to use application w/ little to no understanding about how computer works
  - ↳ Ex: driver of a car

### ■ Application developer

- Deeper understanding of how computer works to create application ← for end users
  - ↳ Ex: car mechanic

### ■ Computer Science / Engineer <sup>our</sup> #focus

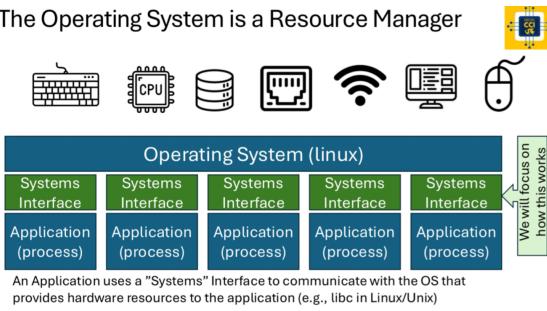
- The CS/CE designs hardware and software needed to support modern computing services ← created by application dev.
  - ↳ Ex: automotive engineer

## ► Operating Systems

### ■ Operating System

- A program that manages a computer's hardware and software resources
  - e.g. CPU, network, disk, etc...
  - \* Systems program is a piece of software intended to be used by other software

The Operating System is a Resource Manager



## ► Our Focus + Tools

### ■ An application uses a "systems" interface to:

- communicate w/ OS that provides hardware resources to app.
- ↳ we will focus on this

### ■ Hardware (Microcontroller) → Board Provided SDK → Systems Programming

### ■ C

- Best language to study systems programming
- Small stable vendor over the machine
- portable across hardware architectures

### ■ C++

- widely used for systems programming
- has evolved in inconsistent ways (Brian's opinion)

### Why not Python / Java?

- "interpreted" languages → but doesn't mean they are slow/inefficient!
  - ↳ however Java is technically "compiled" to bytecode that is executed by Java VM
- unpredictable for memory + runtime
- Abstract important concepts/interactions

### ■ Rust

- Great Language
- Hides some abstractions

### ■ Assembler

- will not use
- Hardware specific (CPU)
- low level

## ► Processor Modes

### ■ Machine Mode

- Most modern processors offer machine mode
  - ↳ highest level of privilege
- can perform operations that have full control over hardware

### ■ Hypervisor Mode

- includes capabilities required to support hardware virtualization
  - ↳ VMware

### ■ Supervisor Mode / Kernel Mode

- sometimes referred to as Kernel mode
- Provides the operating system w/ full access to manage hardware resources

### ■ User Mode

- provides a restricted set of access to machine resources
- when running in user mode
  - ↳ the machine needs to switch to kernel mode to interface w/ hardware resources that are managed by the operating system that runs in supervisor mode

# ► Computer Architecture

## design/organization of a computer's components

- different design decisions have a big influence on certain factors ← eg speed, energy, performance, etc

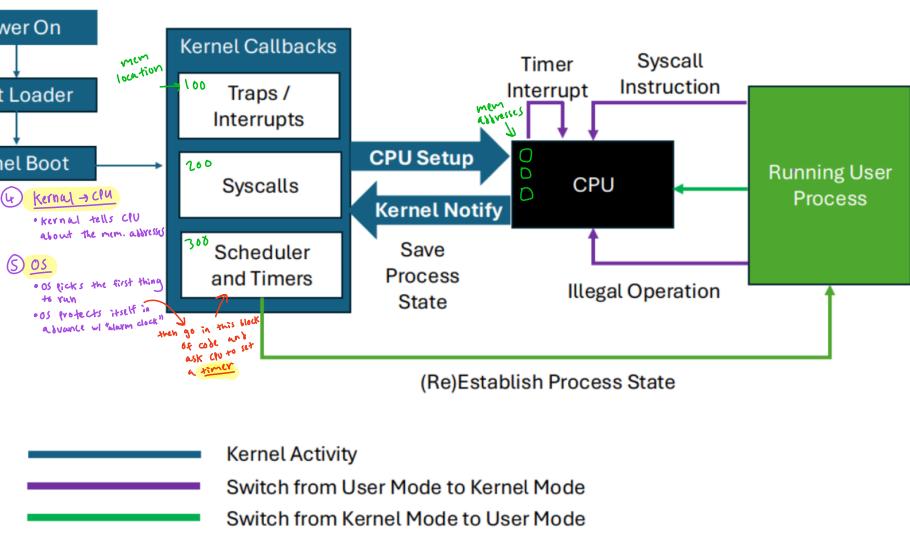
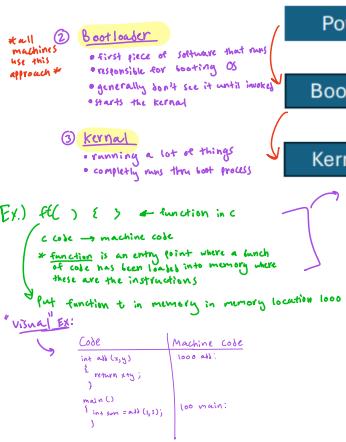
## CPU - Central Processing Unit - "Brain"

- primary component of any device
- responsible for carrying out all instructions to make a computer work → takes instructions ↓ provides output
- single core CPU - can run one thing at a time → either application process or operating sys.

Ex: say we want to open a file  
 f.open ← part of C library  
 has no ability to open a file on its own  
 asks OS to open file which is done through a syscall

## Conceptual Model of Operating System Execution

### ① Power on machine



### Syscall

- syscall-ish
- A fixed function that the OS knows how to interpret
- Every syscall has a number and exactly six arguments