K22 – Operating Systems: Design Principles and Internals

Fall 2025 @dit

Vaggelis Atlidakis

Lecture 02

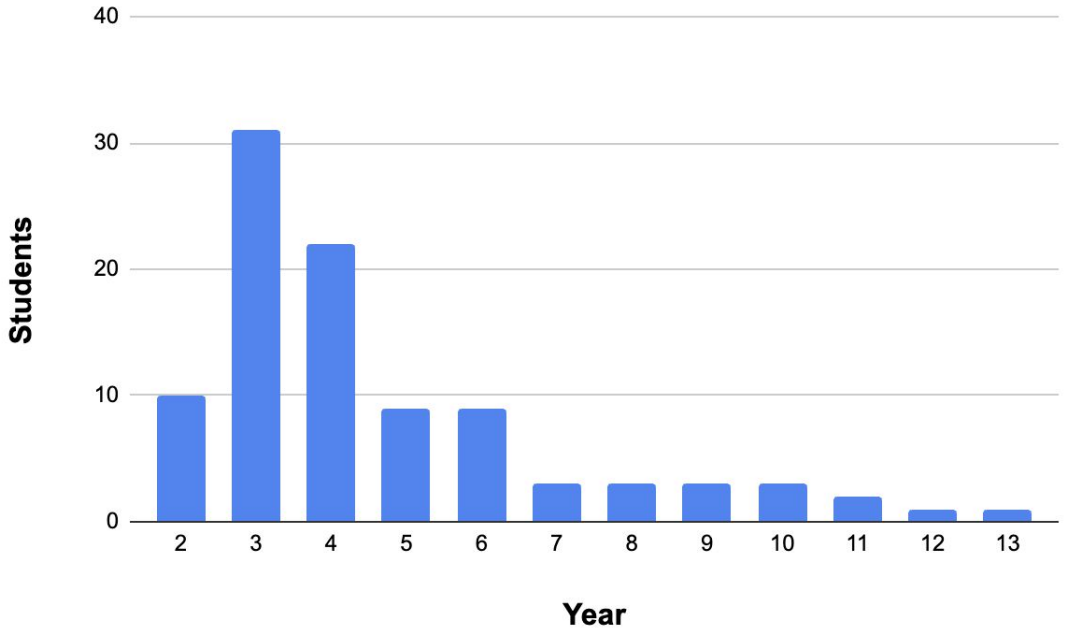
References: Similar OS courses @Columbia, @Stanford, @UC San Diego, @Brown, @di (previous years); and textbooks: Operating Systems: Three Easy Pieces, Operating Systems: Principles and Practice, Operating System Concepts, Linux Kernel Development, Understanding the Linux Kernel

Administrivia

- Register to eclass ASAP
  - Use your sdixxx@di.uoa.gr email (not gmail)
  - The registration will **not** remain open forever

Administrivia

- Where we are today? ~100 ppl.



Administrivia

- **Homework bonus (+0.5 / 10)**
  - +5%, if all homeworks submitted on time
  - +3.75%, if three homeworks submitted on time
  - +2.5%, if two homeworks submitted on time
  - +1.25%, if one homework submitted on time
- **Class attendance (+0.5 / 10)**
  - We' ll see how you get that…
- **Minimum grade requirements**
  - At least the base at final exam

Summary so far

- **Introduction (7/10)**
  - Q1: What is a OS?
  - Q2: Why do we need an OS?
  - Q3: Desirable properties for an OS?
  - Q4: Hardware model?
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: The basic hardware components so far?

Summary so far

- **Introduction (7/10)**
  - Q1: What is a OS?
  - Q2: Why do we need an OS?
  - Q3: Desirable properties for an OS?
  - Q4: Hardware model?
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: The basic hardware components so far?

Summary so far

- **Introduction (7/10)**
  - **Q1: What is a OS?**
  - **Q2: Why do we need an OS?**
  - **Q3: Desirable properties for an OS?**
  - Q4: Hardware model?
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: The basic hardware components so far?

What is an OS?

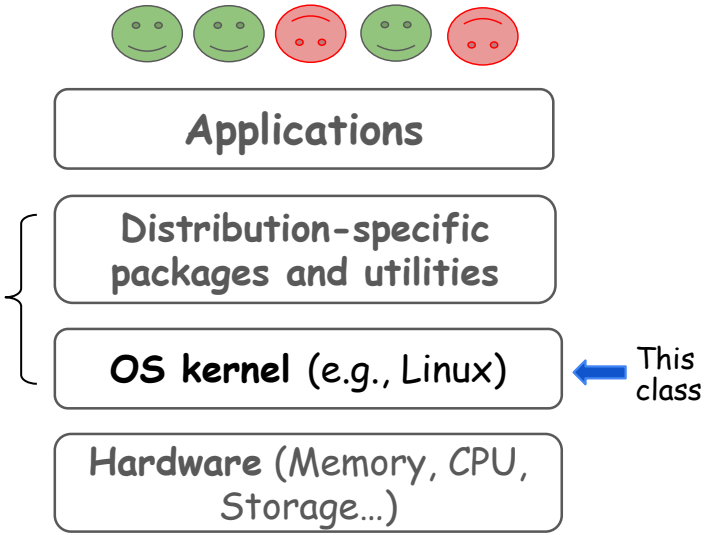
- A layer of abstraction that translates the hardware of a machine into (standardized) software concepts that applications use

Why need an OS?

- Multiple users not necessarily sane, build and run different applications, at the same time on shared hardware (design assumptions)

Desirable properties?

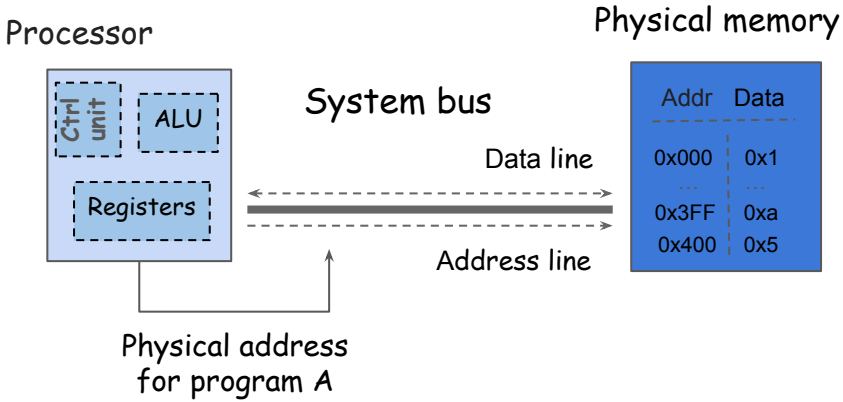
- ***Security***: Does the system behave as expected in adversarial contexts?
- ***Reliability***: Does the system behave as expected given benign failures?
- ***Portability***: Is it easy for developers to build and maintain apps?
- ***Fairness***: Is the system "pleasant" to use?



Summary so far

- **Introduction (7/10)**
  - **Q1: What is a OS?**
  - **Q2: Why do we need an OS?**
  - **Q3: Desirable properties for an OS?**
  - **Q4: Hardware model?**
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: The basic hardware components so far?

What is the hardware we build OSes for?



- **Physical Memory (PM)**
  - Stores data addressable on a byte granularity
- **Processor**
  - Reads data from memory to its registers
  - Performs computations
  - Writes the data from its registers to memory
- **System bus**
  - Memory and processor communication channel
- Is this model satisfactory?
- Reminds you of something?

Summary so far

- **Introduction (7/10)**
  - **Q1: What is a OS?**
  - **Q2: Why do we need an OS?**
  - **Q3: Desirable properties for an OS?**
  - **Q4: Hardware model?**
  - **Q5: Design principles?**
  - Q6: Hardware support to enforce design principles?
  - Q7: The basic hardware components so far?

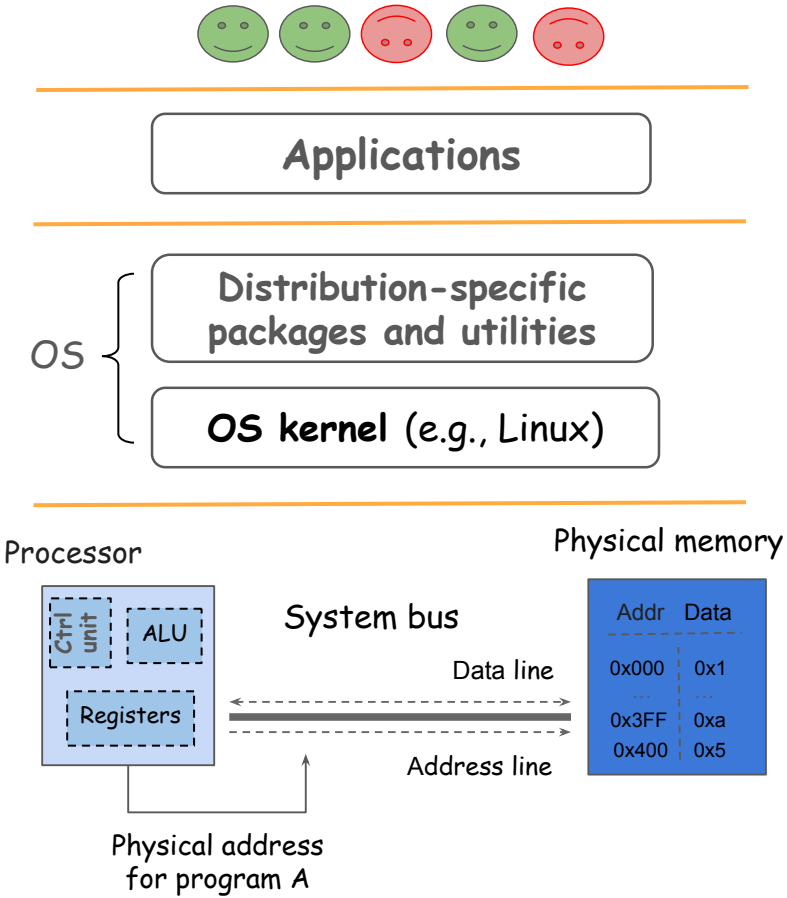
Designing an OS kernel

Desirable properties

1) Security, 2) Reliability, 3) Portability, 4) Fairness

From desirable properties to design principles

- ***Fault Isolation***: Errors in one running program do not affect any other program
- ***Principle of Least Privilege (PoLP)***: Any program has the minimum privileges necessary to perform its function
- ***Preemption***: The OS is always able to take control of the processor, regardless of what programs are running

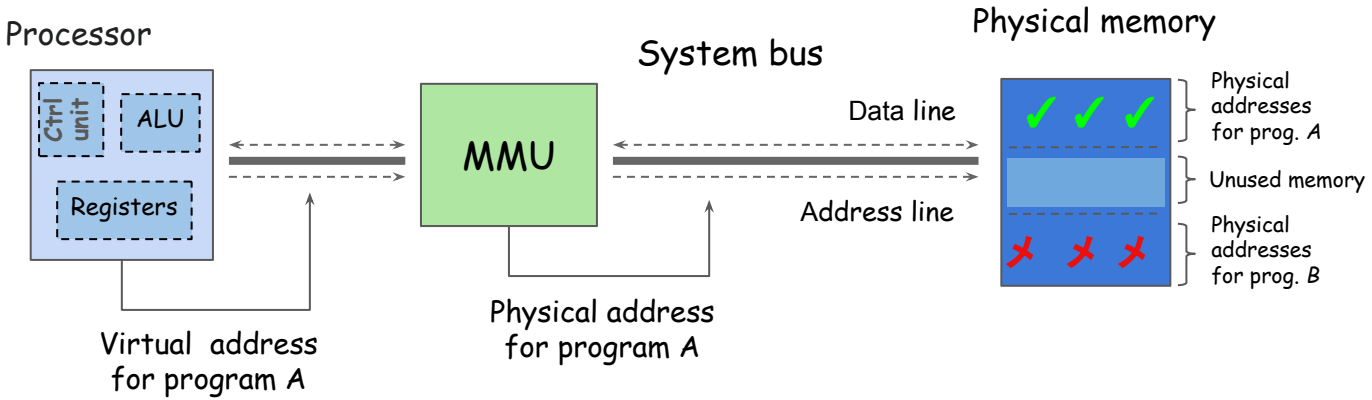


Summary so far

- Introduction (7/10)
  - Q1: What is a OS?
  - Q2: Why do we need an OS?
  - Q3: Desirable properties for an OS?
  - Q4: Hardware model?
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: The basic hardware components so far?

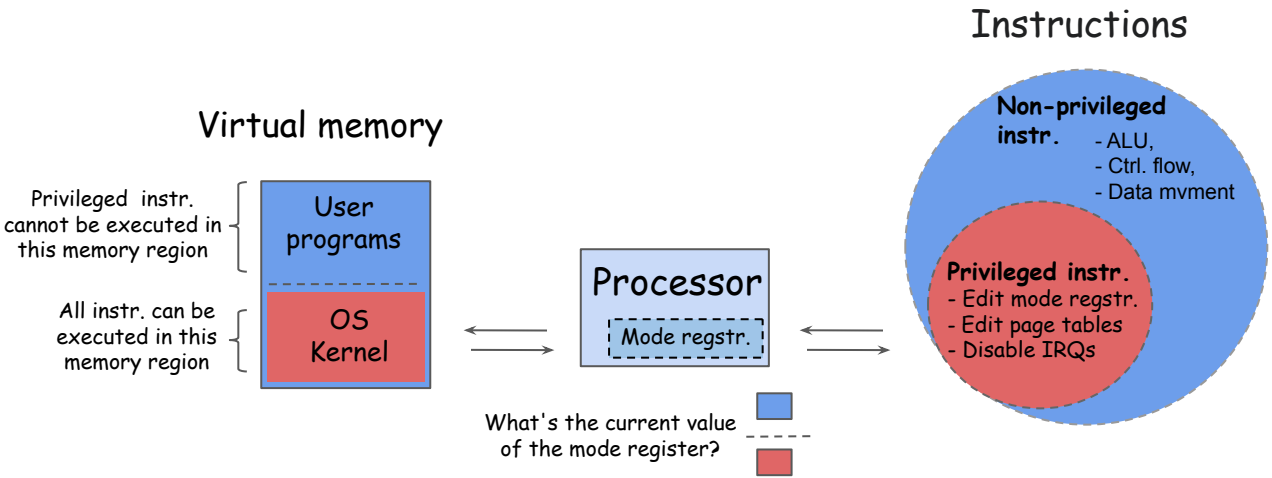
Implementing fault isolation

- In simple words: load/ store/ jmp instructions of a program cannot read, write, or jump to another program's memory



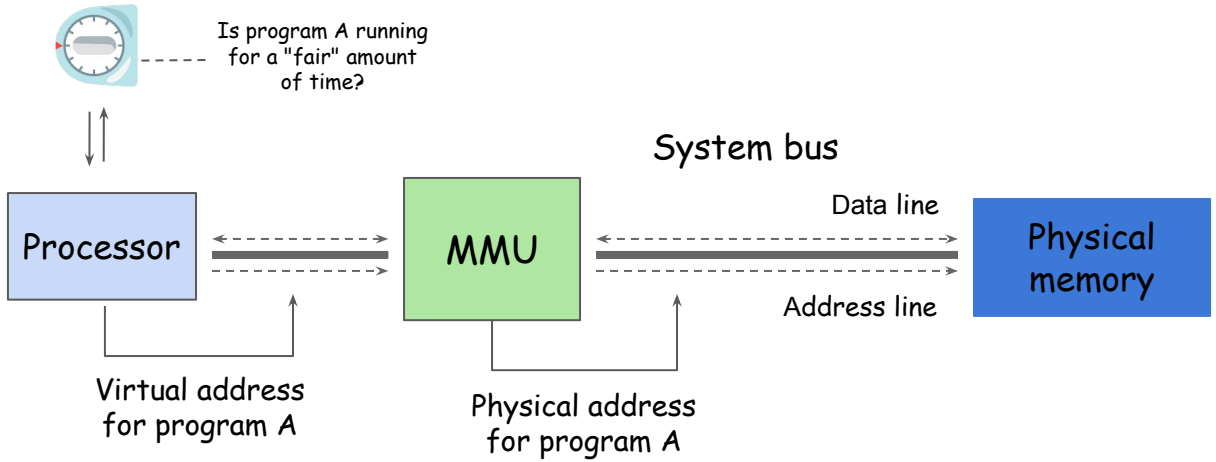
Implementing dual-mode execution

- In simple words: A trusted portion of the code (the OS) must have full control of the hardware



Implementing preemption

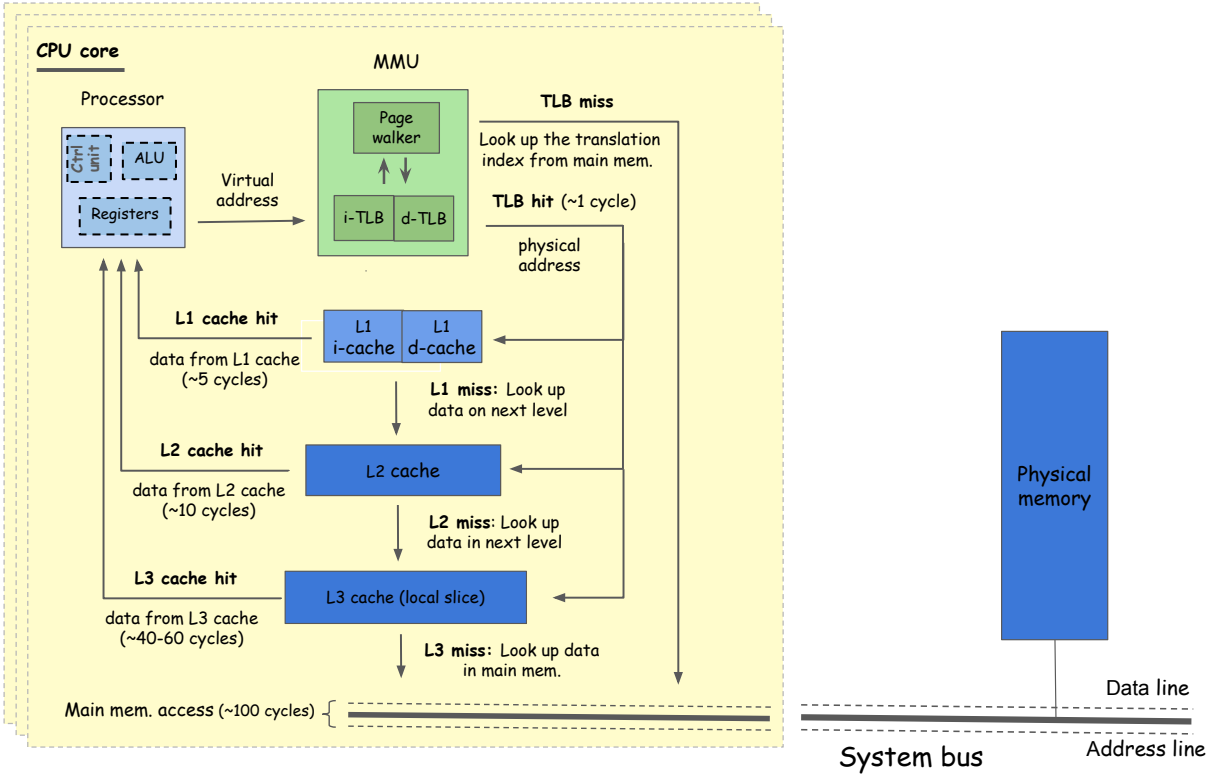
- In simple words: The OS should be able to periodically gain control of the processor regardless of what programs are executing



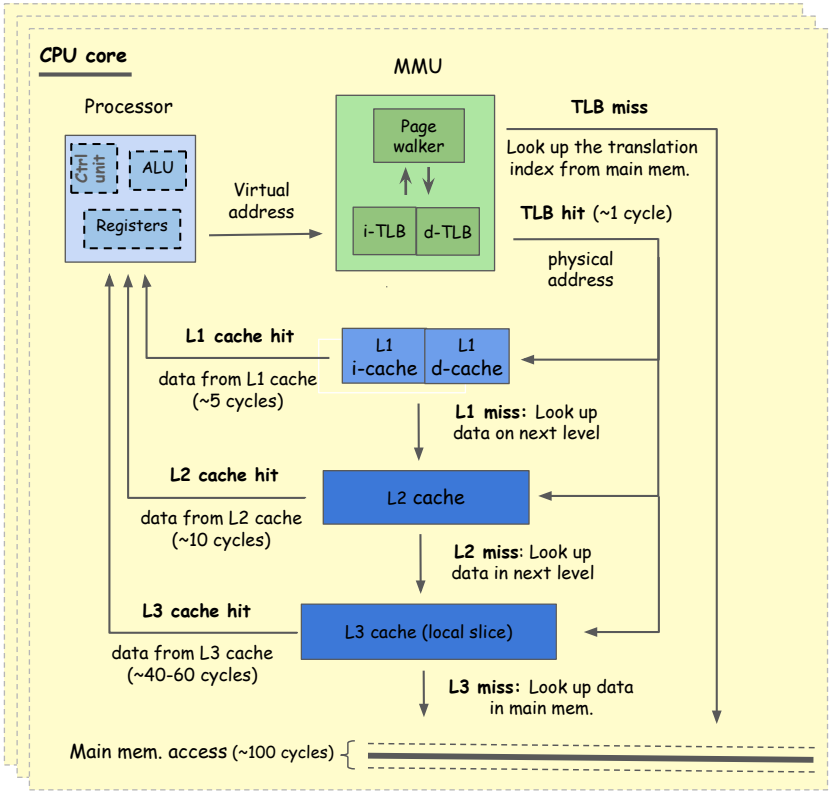
Summary so far

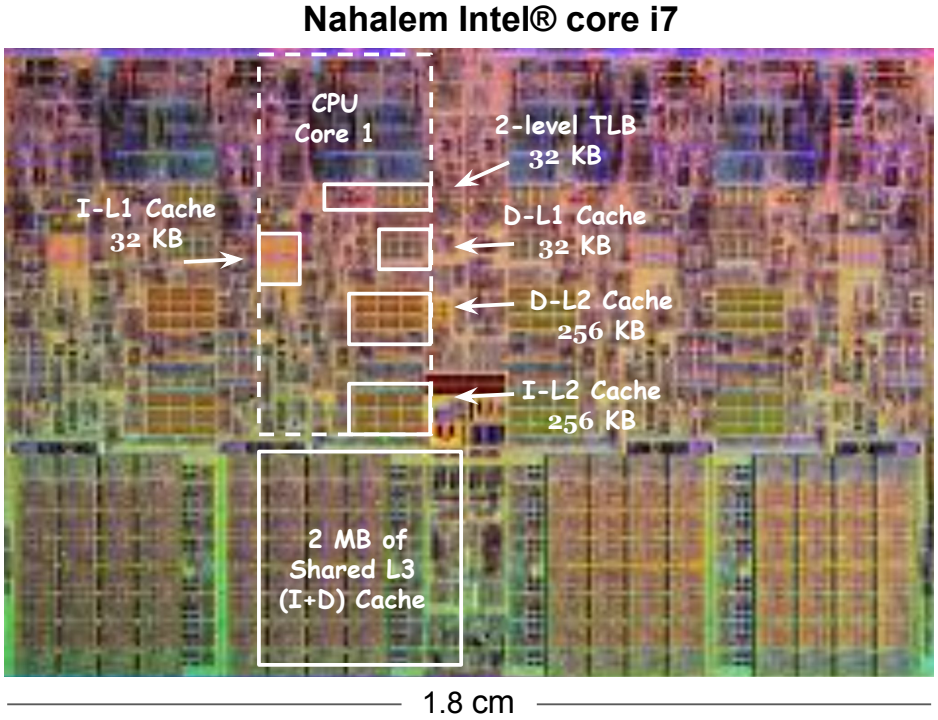
- Introduction (7/10)
  - Q1: What is a OS?
  - Q2: Why need an OS?
  - Q3: Desirable properties for an OS?
  - Q4: Abstract hardware model?
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: Basic hardware components so far?

Hardware components leveraged by OS mechanisms



Reality check on hardware's scale and access times

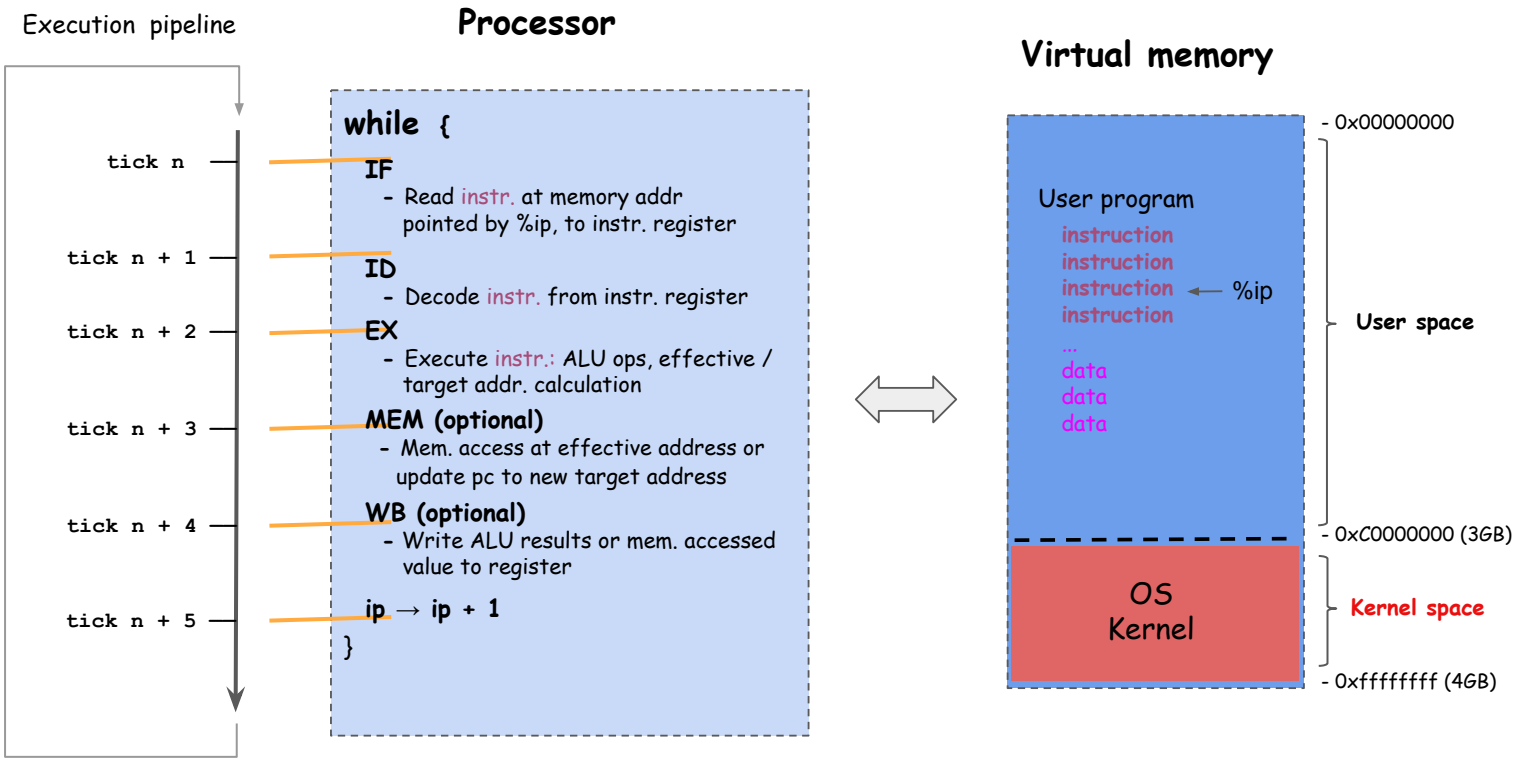




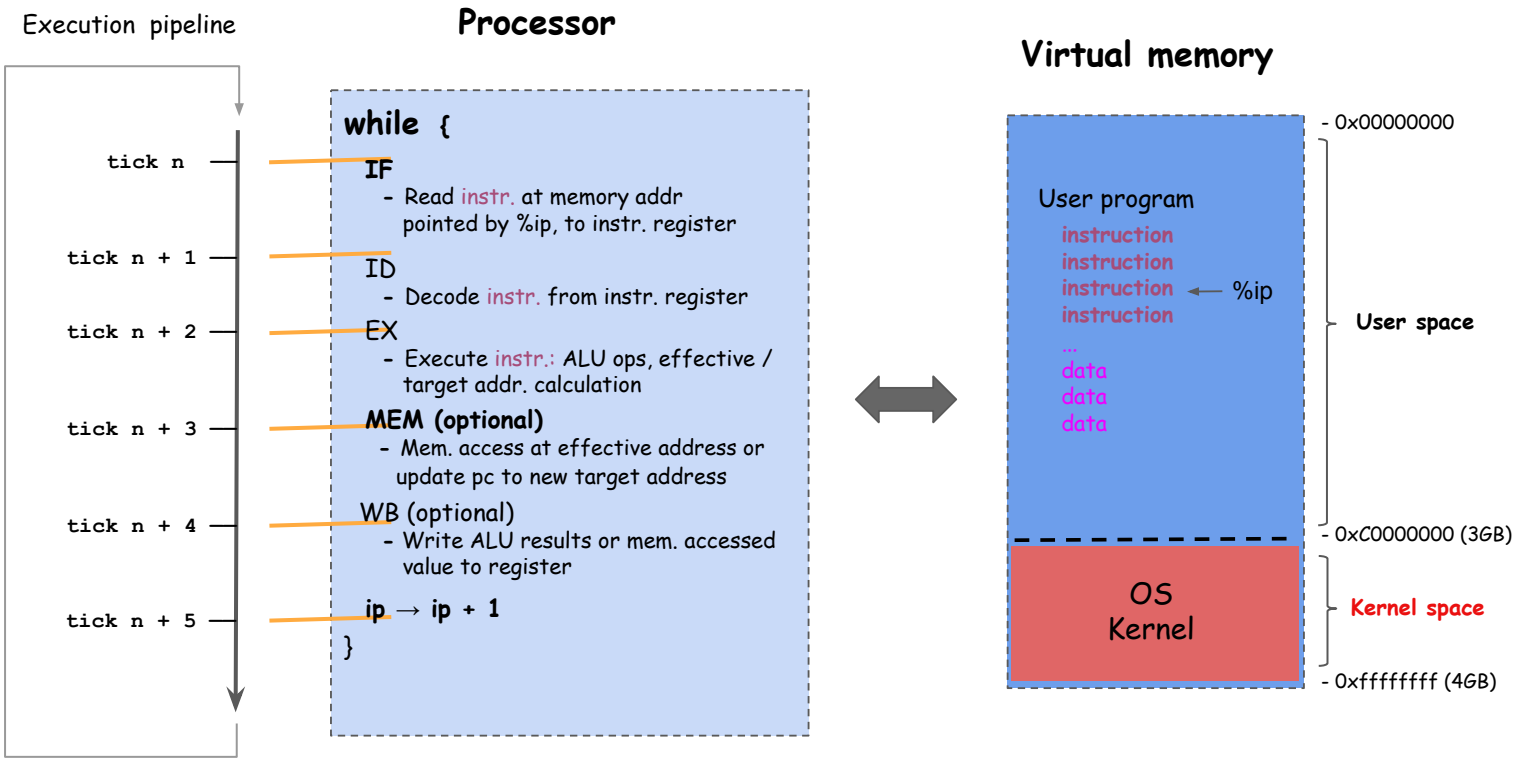
Summary so far

- Introduction (7/10)
  - Q1: What is a OS?
  - Q2: Why need an OS?
  - Q3: Desirable properties for an OS?
  - Q4: Abstract hardware model?
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: Basic hardware components so far?

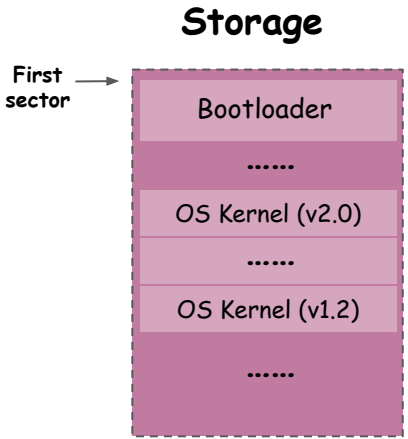
How does the processor execute programs?



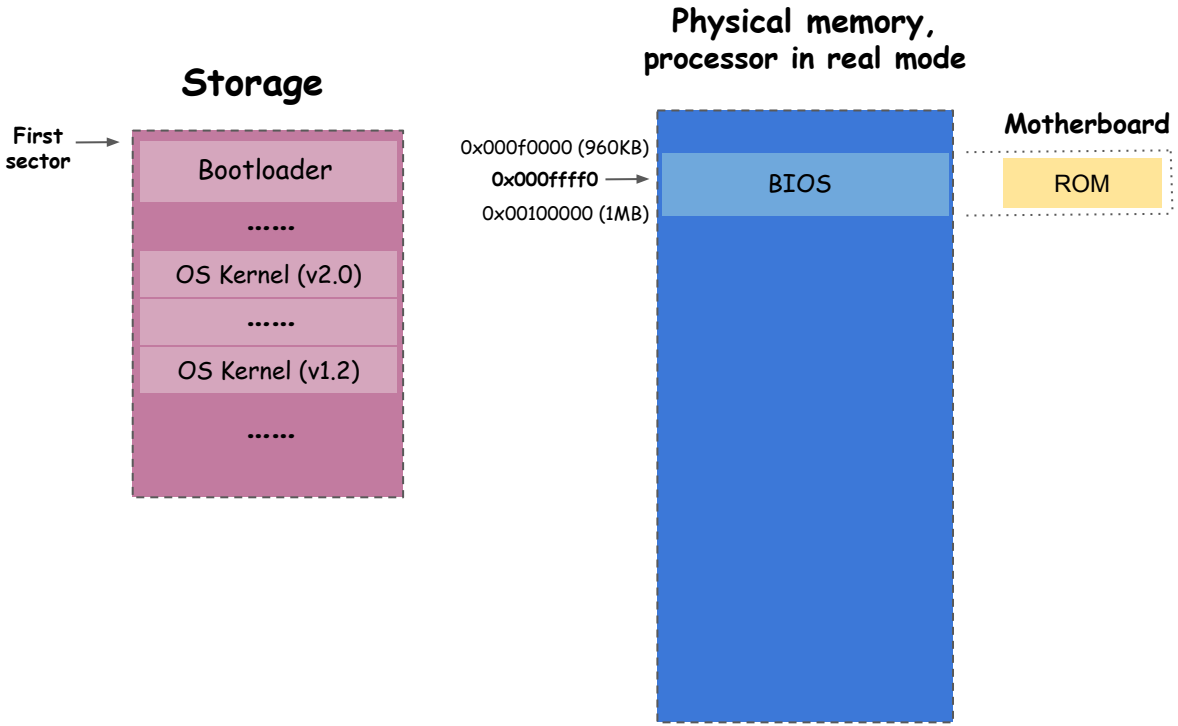
When may the processor access memory?



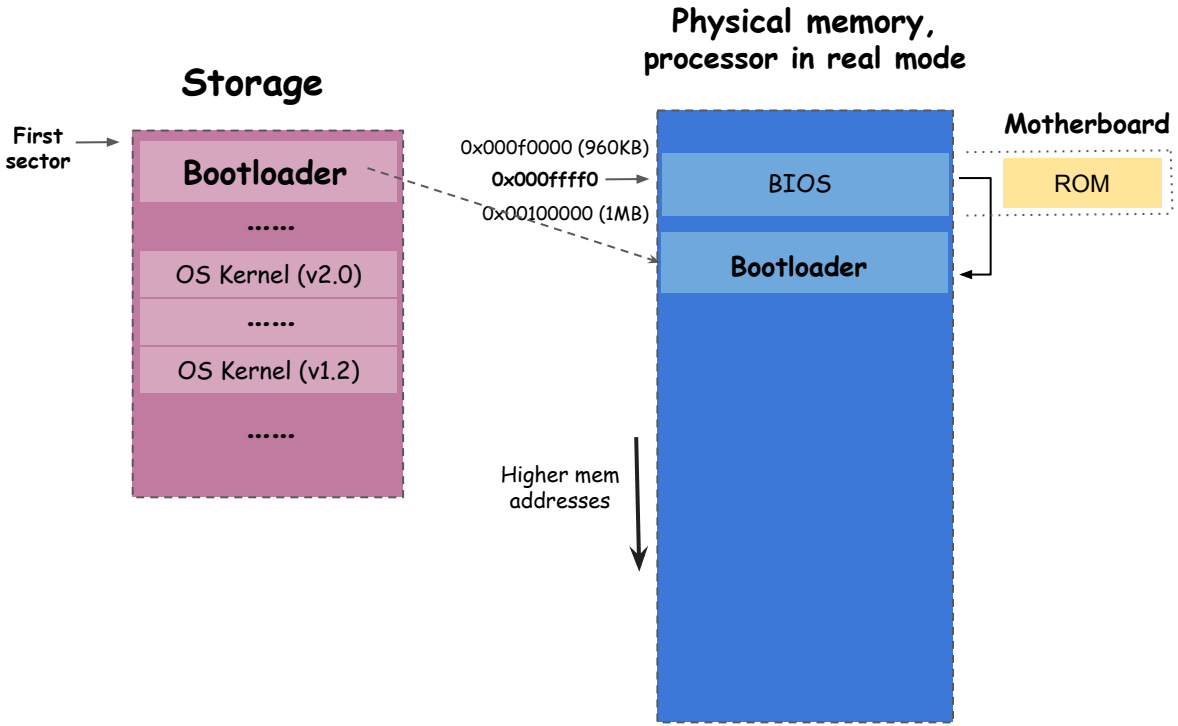
How does the system boot?



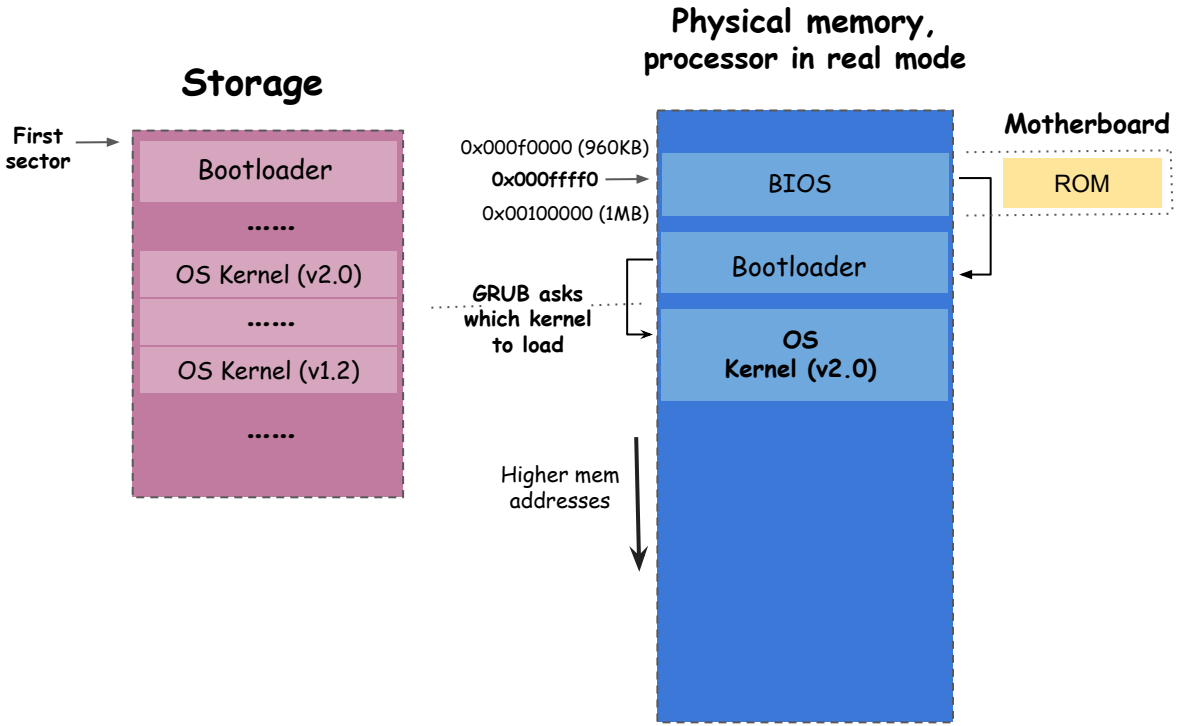
How does the system boot?



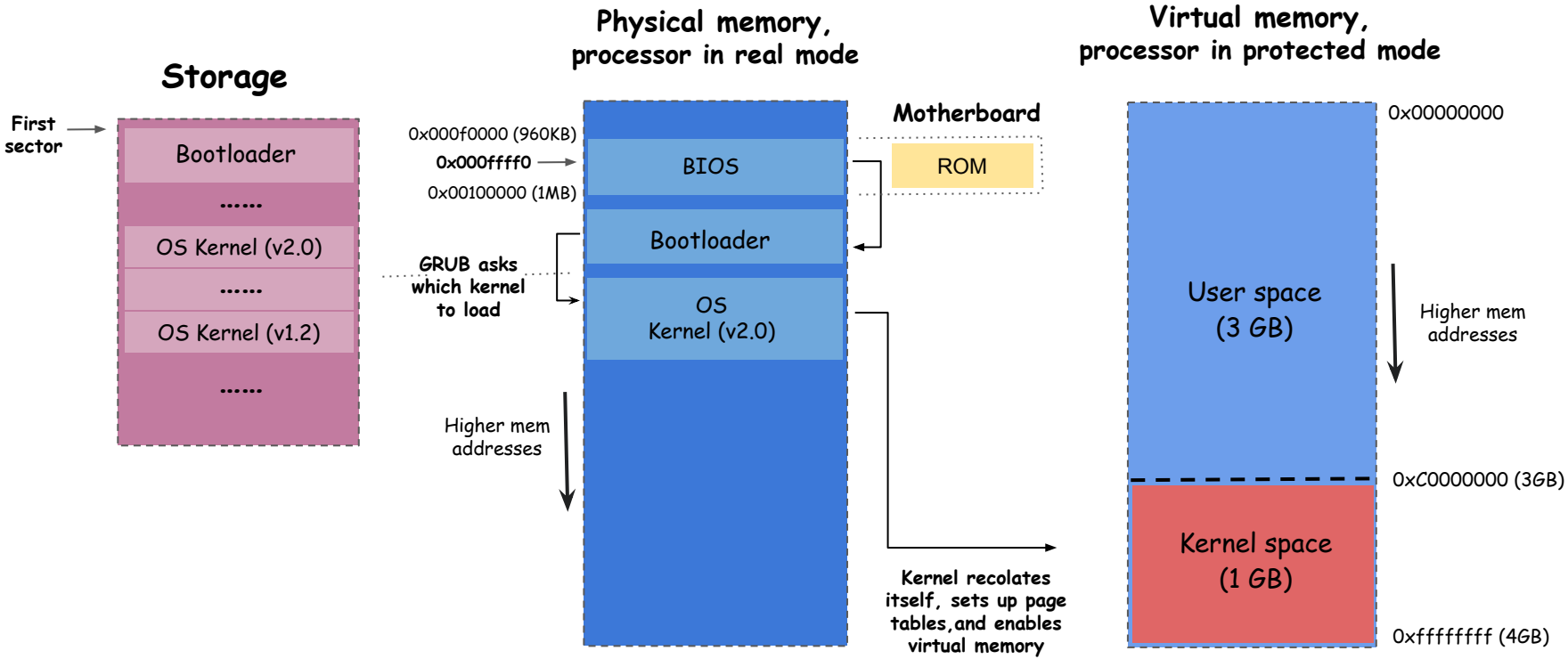
How does the system boot?



How does the system boot?



How does the system boot?



Summary so far

- Introduction (10/10)
  - Q1: What is a OS?
  - Q2: Why need an OS?
  - Q3: Desirable properties for an OS?
  - Q4: Abstract hardware model?
  - Q5: Design principles?
  - Q6: Hardware support to enforce design principles?
  - Q7: Basic hardware components so far?
  - Q8: Reality check on hardware's scale and access times?
  - Q9: How does the processor execute programs?
  - Q10: How does the system boot?