Operating Systems Overview
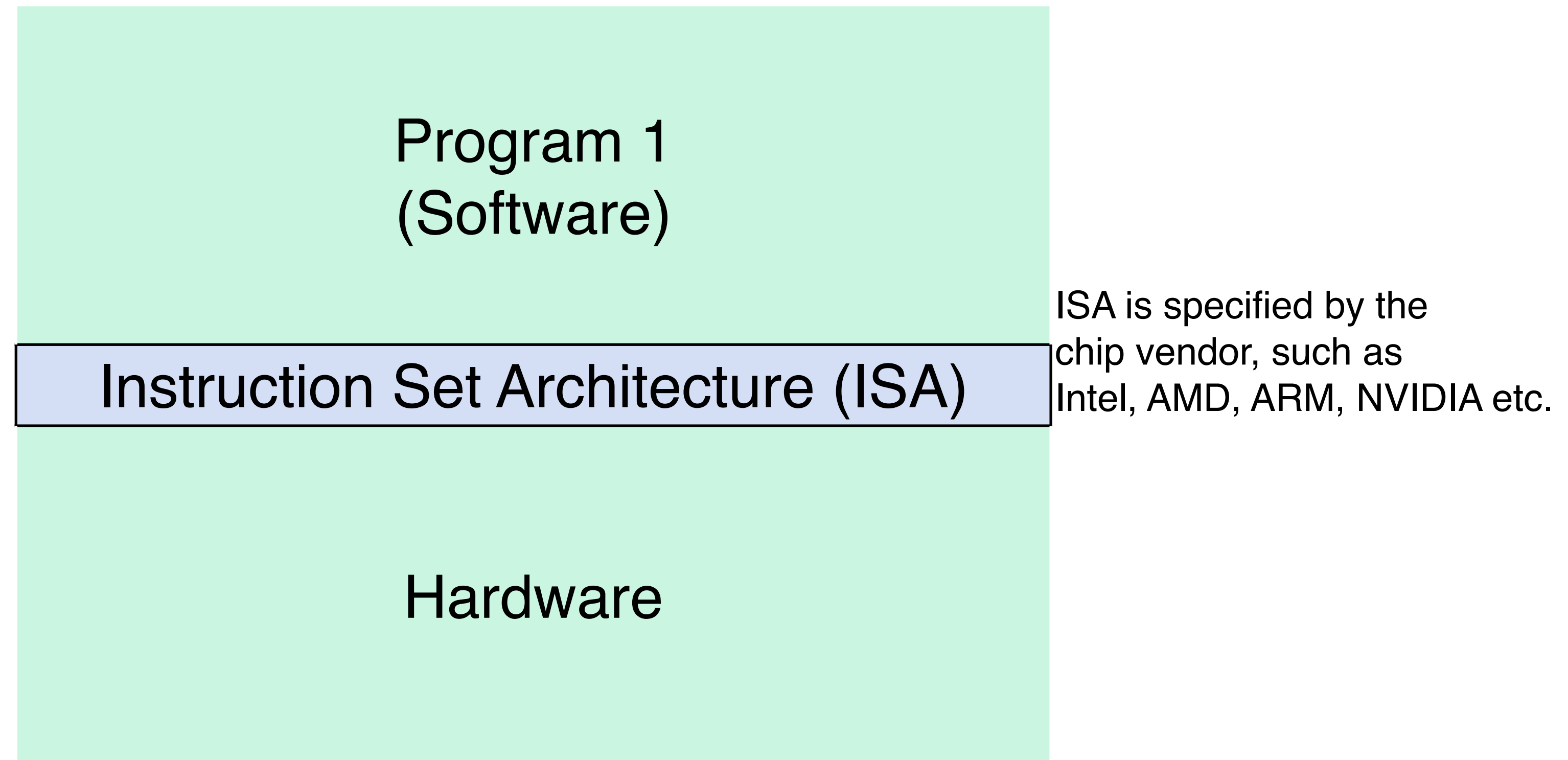
Prof. Kartik Gopalan

What is an Operating System?

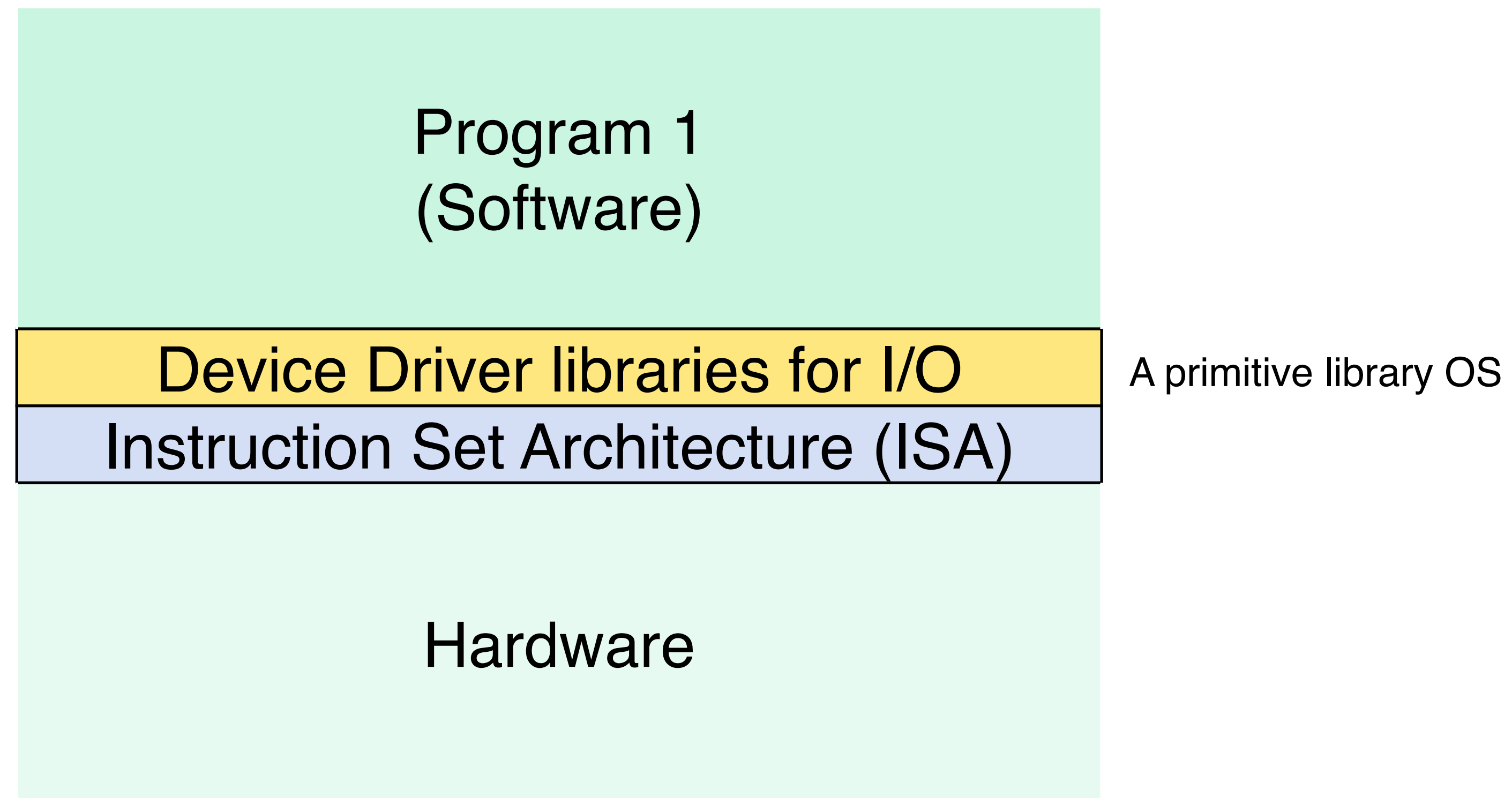# What is an Operating System?

- A bunch of software and data residing somewhere in memory.

  - But its not just *any* software.

- OS is the <u>*most privileged*</u> software in a computer.

  - *Privileged* means that OS can do special things, like write to disk, talk over the network, control memory and CPU usage, etc.

- OS manages all system resources

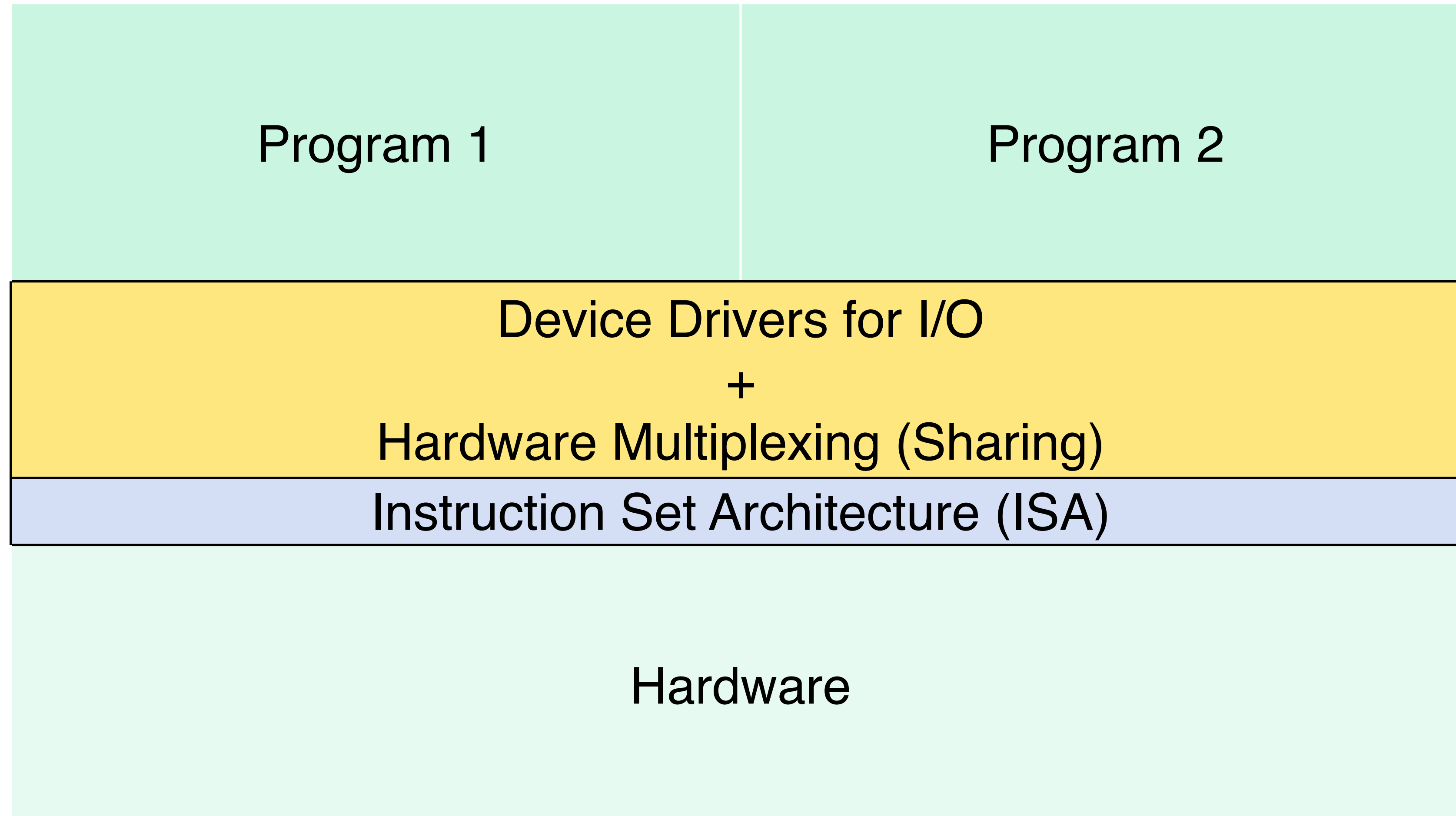 - CPU, Memory, and I/O devices

Why do we need an OS?

Program 1
(Software)

Instruction Set Architecture (ISA)

ISA is specified by the
chip vendor, such as
Intel, AMD, ARM, NVIDIA etc.

Hardware

Why do we need an OS?

But the program doesn't know how to access hardware
devices for input/output (I/O)

Program 1
(Software)

Device Driver libraries for I/O — A primitive library OS

Instruction Set Architecture (ISA)

Hardware

Why do we need an OS?

But what if two programs need to share the hardware?

Program 1	Program 2

**Device Drivers for I/O
+
Hardware Multiplexing (Sharing)**

Instruction Set Architecture (ISA)

Hardware

Why do we need an OS?

But what if
- two programs don't trust each other?
- OS doesn't trust programs?
- hardware doesn't trust programs?
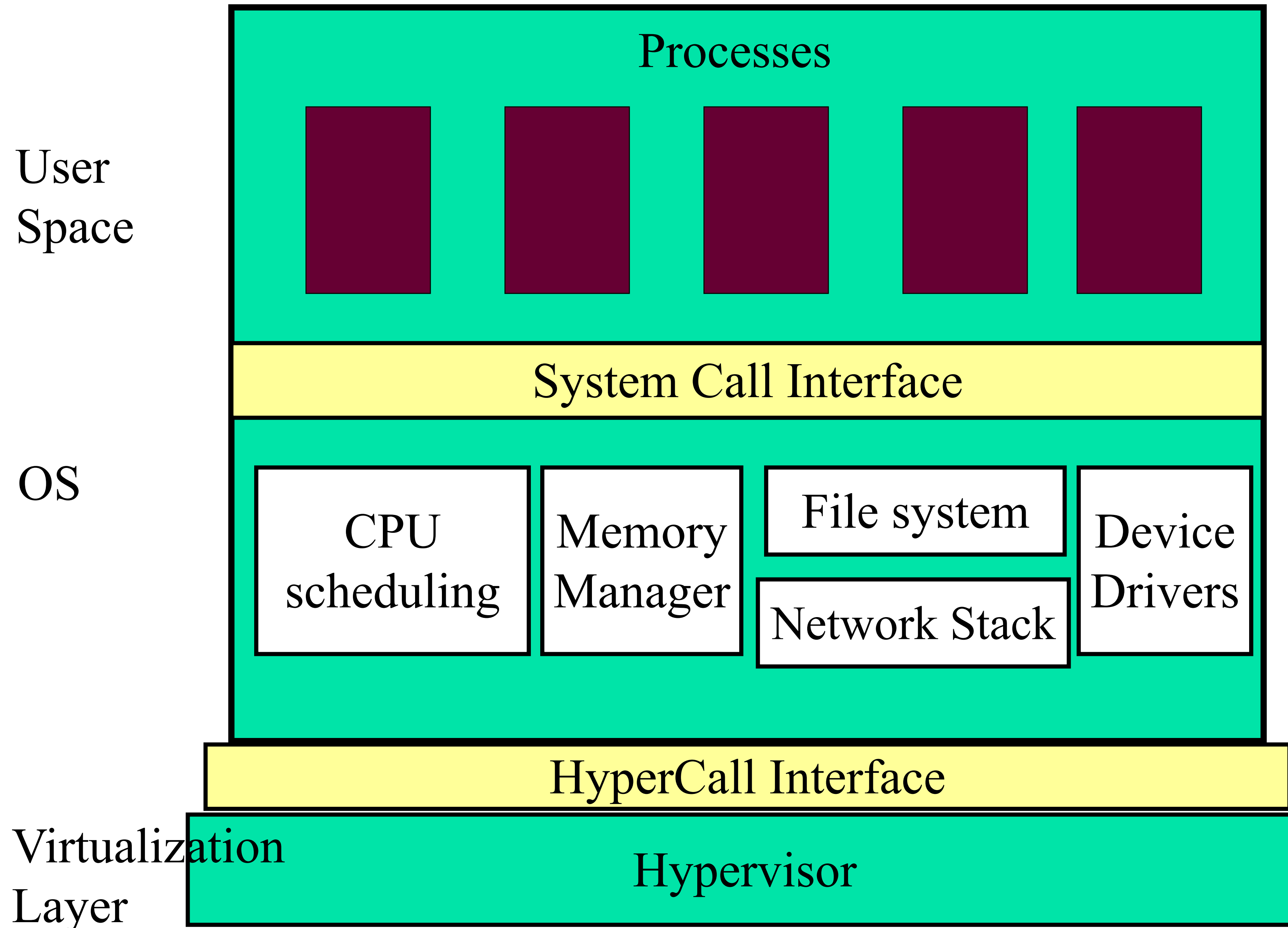
Program 1	Program 2

I/O
+
Sharing } = A modern OS
+
Protection

Instruction Set Architecture (ISA)

Hardware
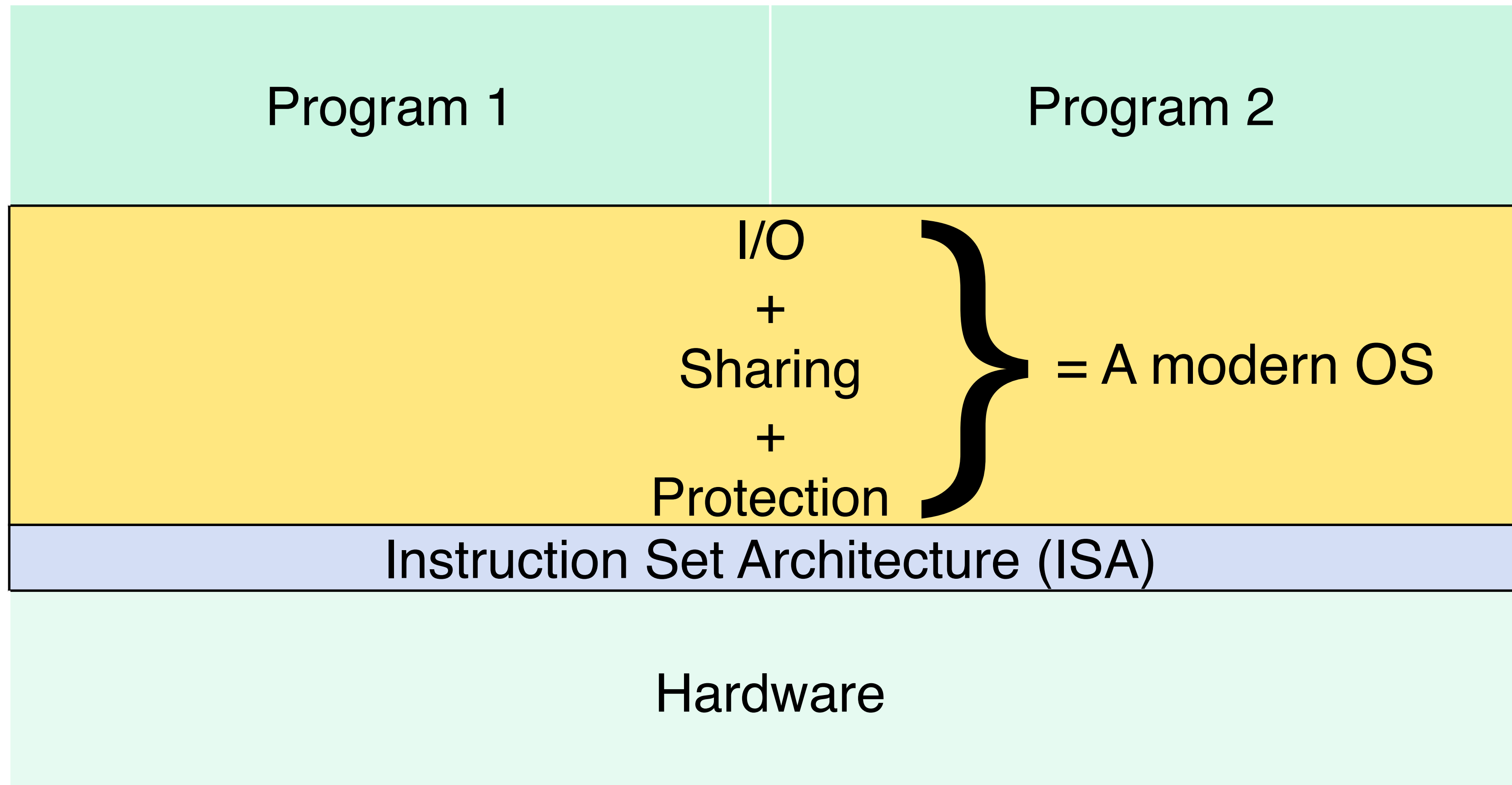
But when does the OS "run"?

But when does the OS "run"?

Four ways to invoke OS code

Layers of Software

Processes

User Space

System Call Interface

OS

CPU scheduling

Memory Manager

File system

Network Stack

Device Drivers

HyperCall Interface

Virtualization Layer

Hypervisor
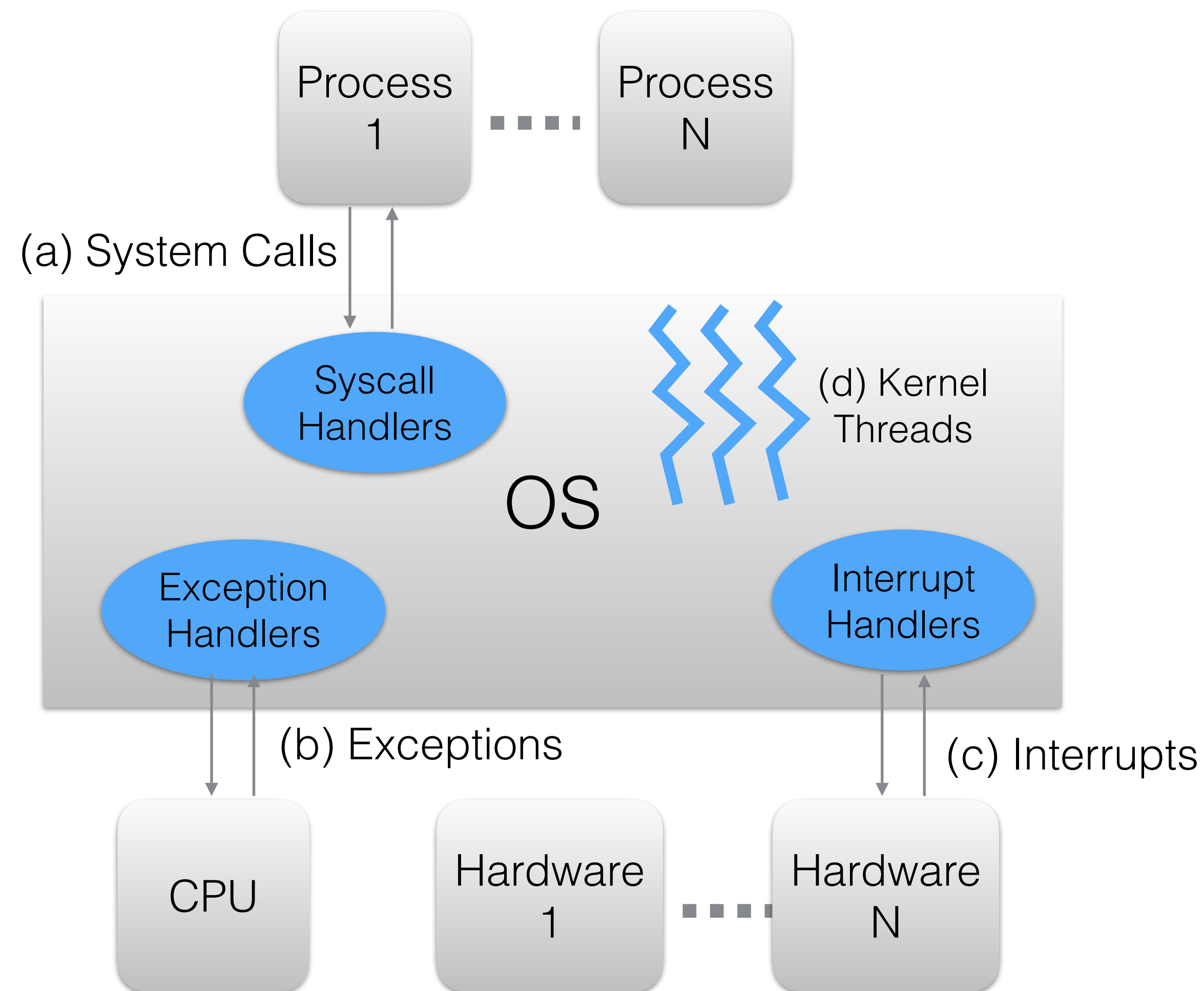
Interfaces in a Computer System

(1)

User-level processes

(2)

(3) Libraries (3)

Operating System

(8) (8) (8) (7) (7)

Execution hardware (9)

(10) (10) Memory Translation

System Interconnect (Bus) (12)

(11)

Controllers (13) Controllers (14)

I/O Devices Main Memory
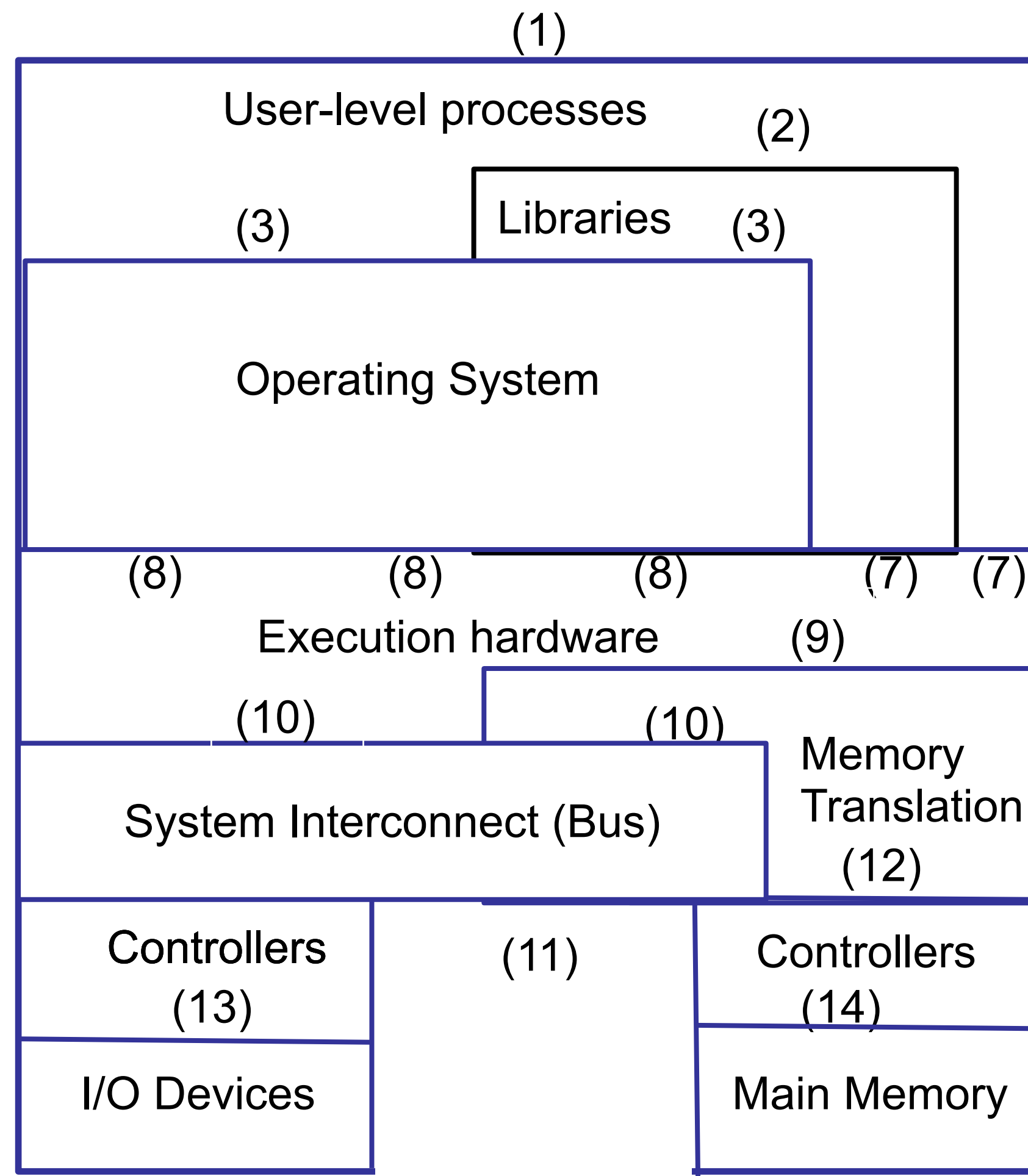
- Instruction Set Architecture (ISA) = 7+8

- User ISA = 7

- System ISA = 8

- System calls = 3

- Application Binary Interface (ABI) = 3 + 7

- Application Programmer's Interface (API) = 2 + 7

History of OS

- 1950s and 1960s: Early operating systems were simple batch processing systems
  - Users provided their own "OS" as libraries.
- 1960s and1970s: Multi-programming on mainframes
  - Concurrency, memory protection, Kernel mode, system calls, hardware privilege levels, trap handling
  - Earliest Multics hardware and OS on IBM mainframes
  - Which led to the first UNIX OS which pioneered file systems, shell, pipes, and the C language.
- 1980s: Personal computing era
  - MacOS, IBM PC and its DOS, Windows, and so forth.

- 1980s also saw the fragmentation of UNIX
  - Each big company had its own version (IBM, Apple, HP, SUN, SGI, NCR, AT&T….)
  - LOT of legal wrangling over IP and copyrights
- 1990s: Then came BSD and Linux
  - Open source.
  - Led the way to modern OSes and cloud platforms
- 1990s also saw wider adoption of threads and parallelism
- 2000 and beyond: Mobile device OS and hypervisors
  - Android, iOS
  - VMWare ESX, Xen, Linux/KVM etc.