Operating Systems Overview

What is an Operating System?

Examples

What is an operating system?

?

What is an Operating System

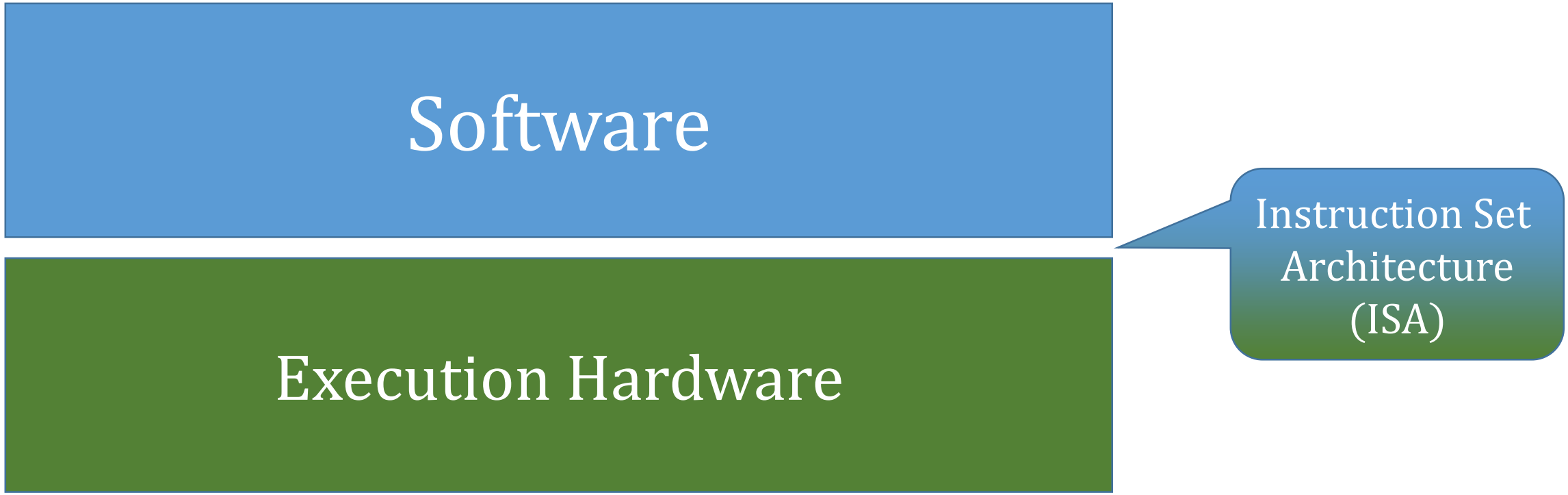
- Software – just like an application program, but…

- OS has special *privileges*
  - Read and write from disk drives
  - Read and write from the network
  - Manage memory
  - Manage CPU Resources

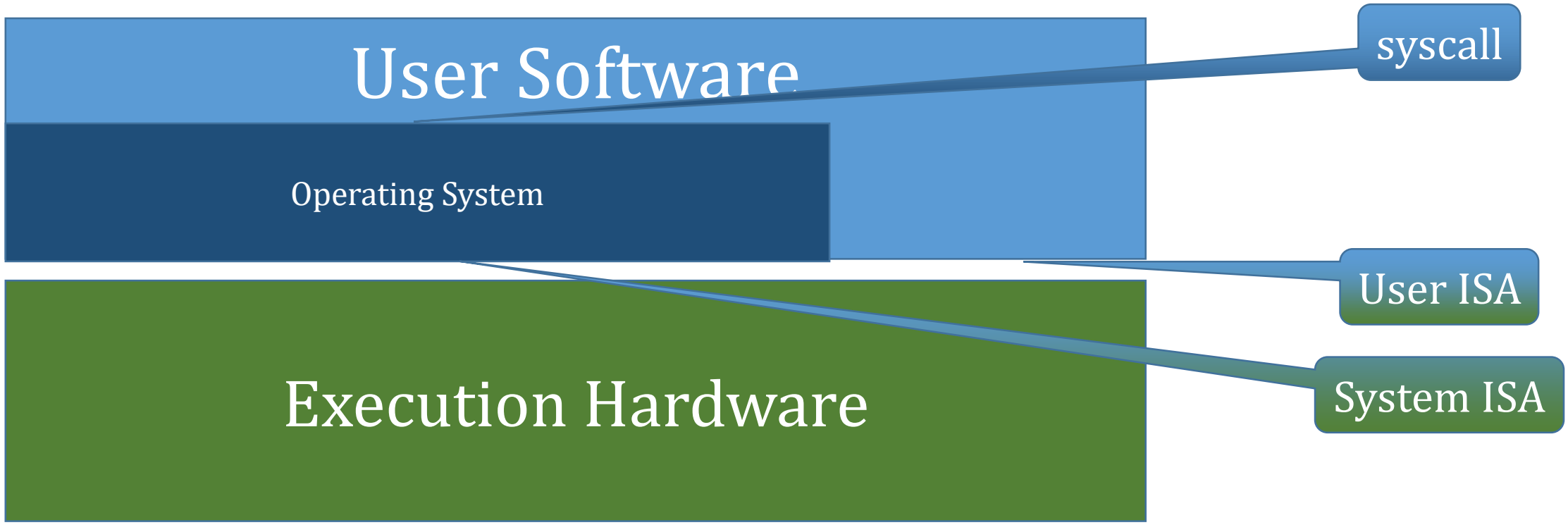
OS Manages Resources

- CPU
- Memory
- Network
- Devices
  - Disk Drives
  - Terminal
  - Keyboard
  - Mouse (touch pad, touch screen)
- Users

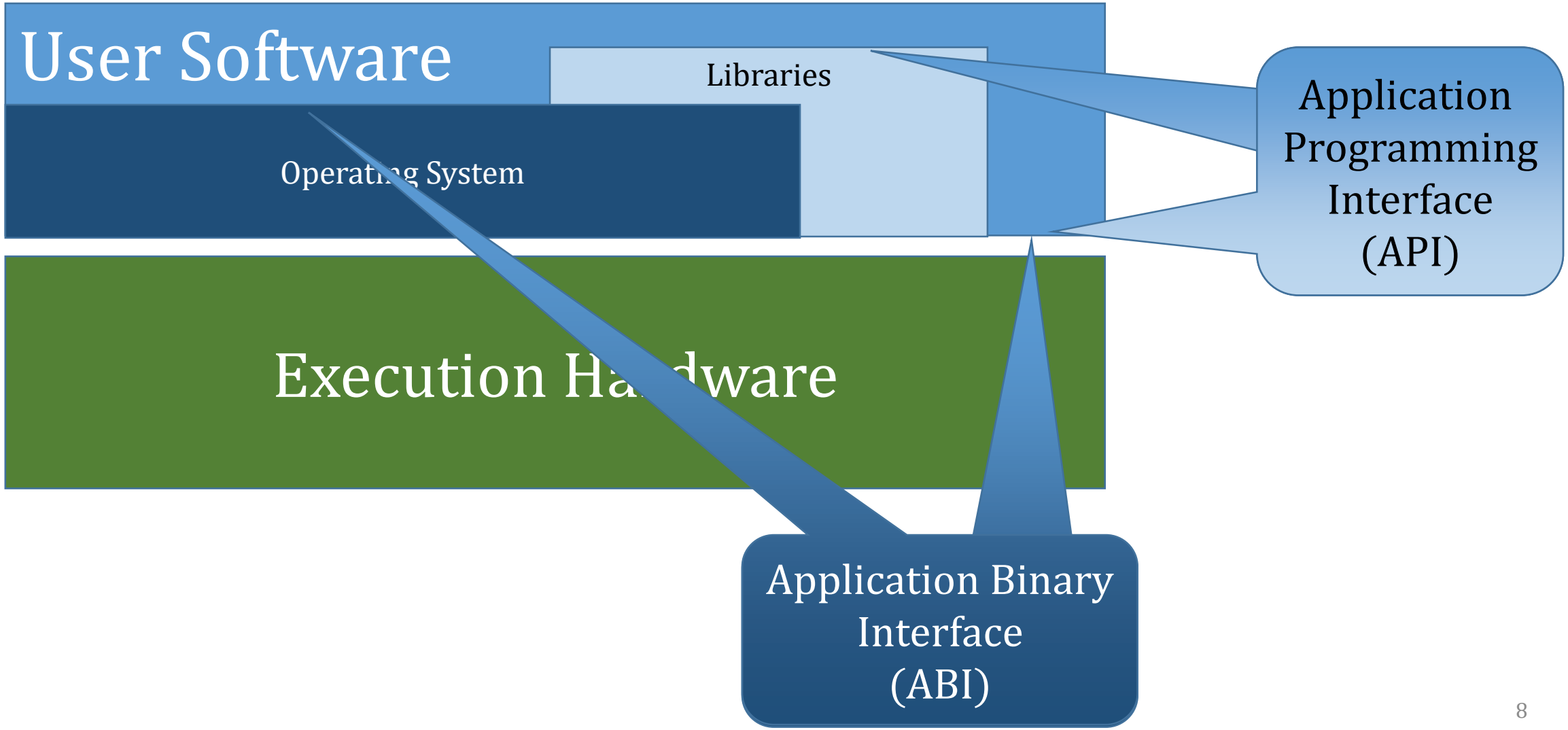
Interfaces: Instruction Set Architecture



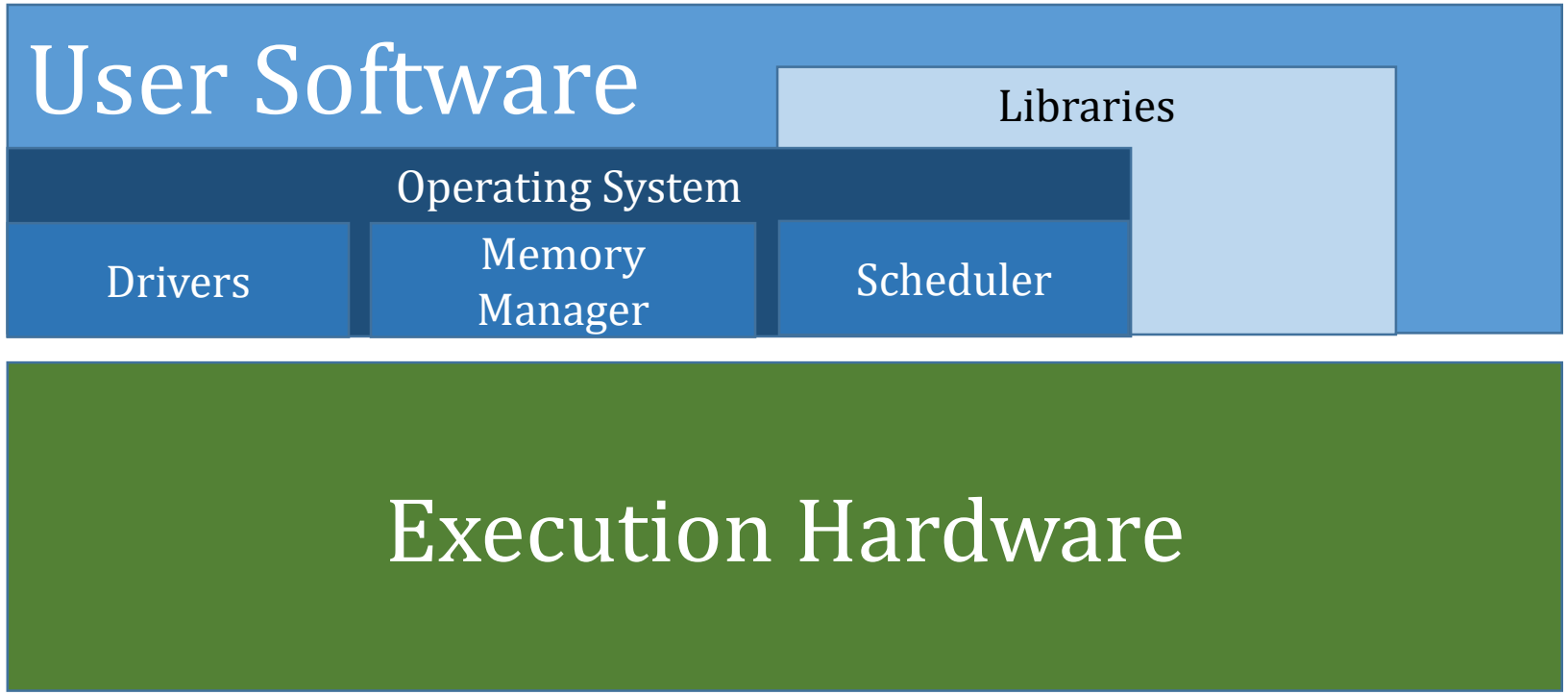
Interfaces: Operating System



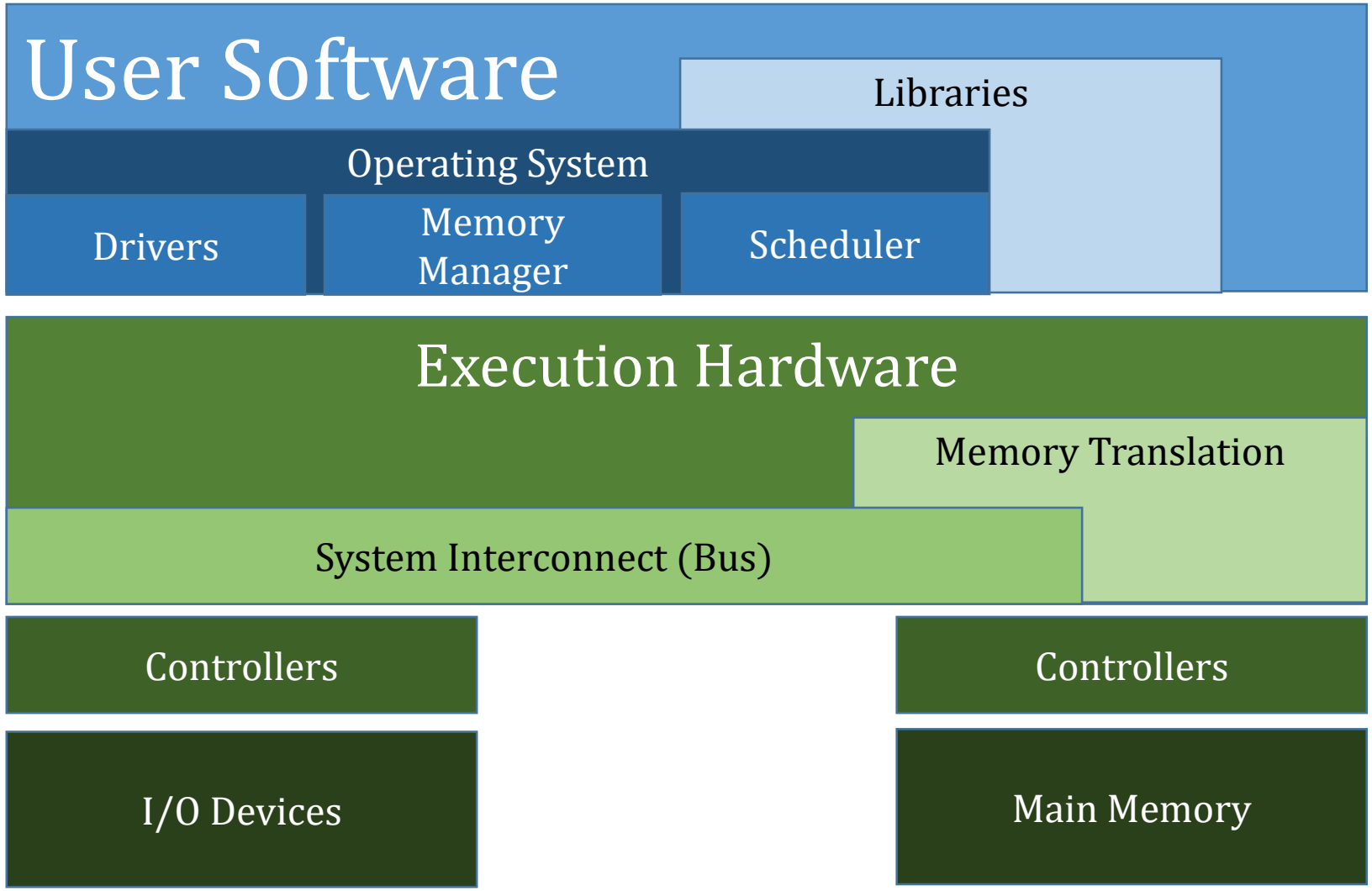
Interfaces: Libraries



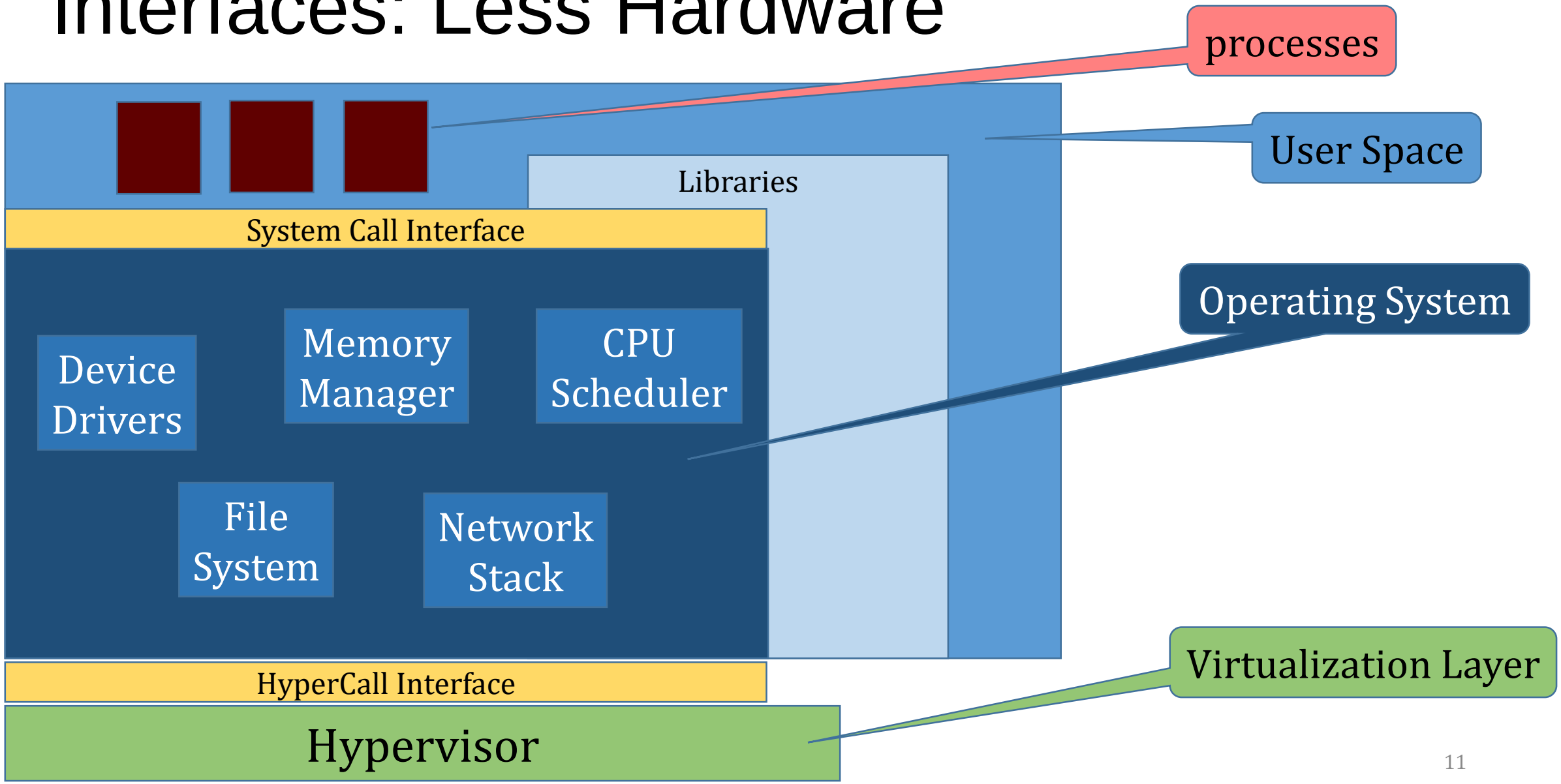
Interfaces: Operating System Detail



Interfaces: More Hardware



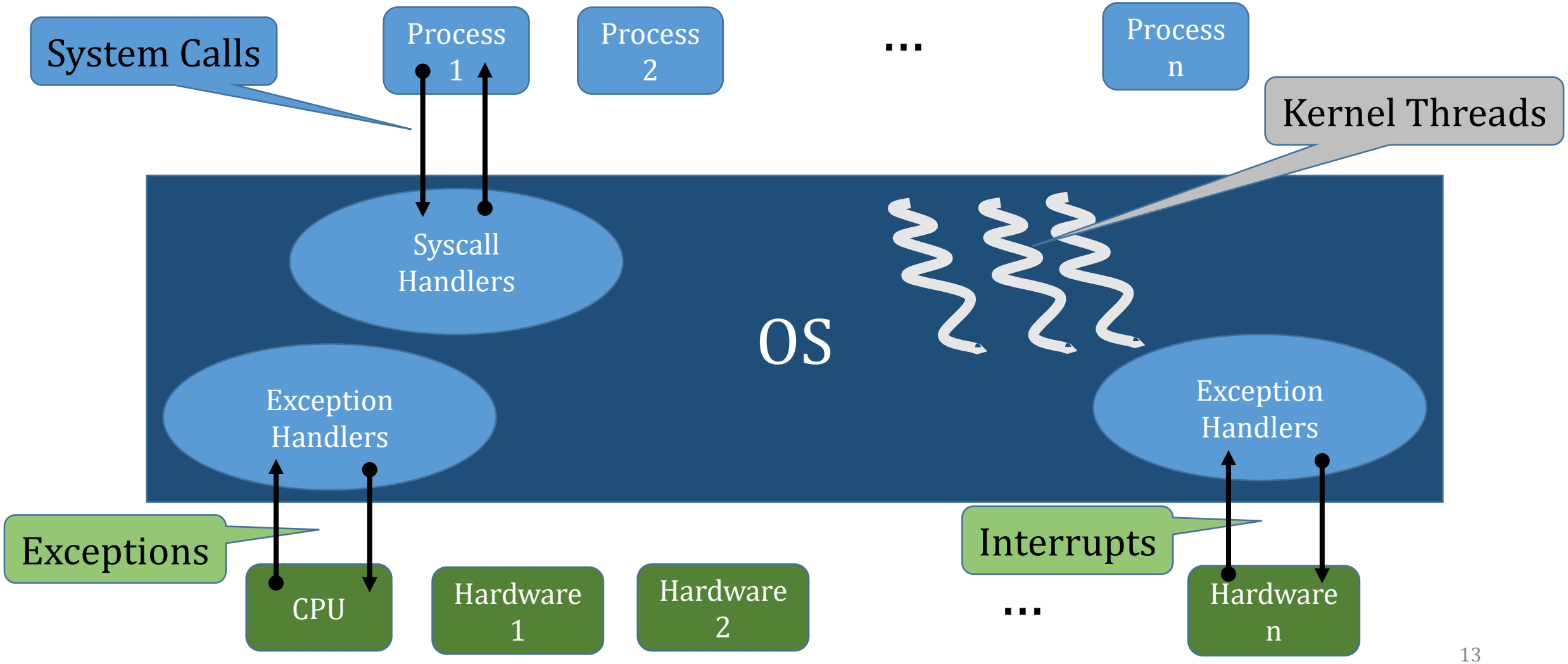
Interfaces: Less Hardware



When does the OS Run?

?

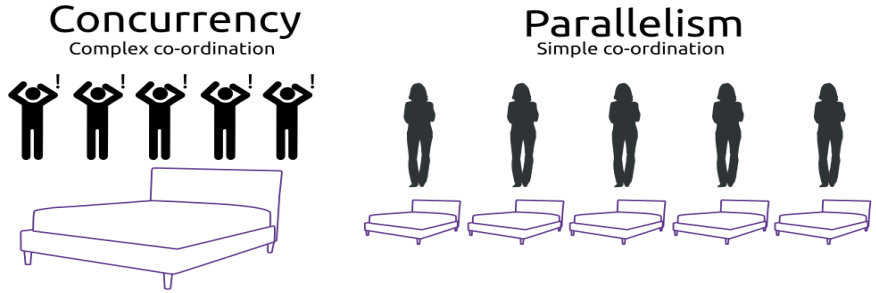
Invoking OS Code



Operating System Tasks

- Virtualization

- Concurrency



- Persistence

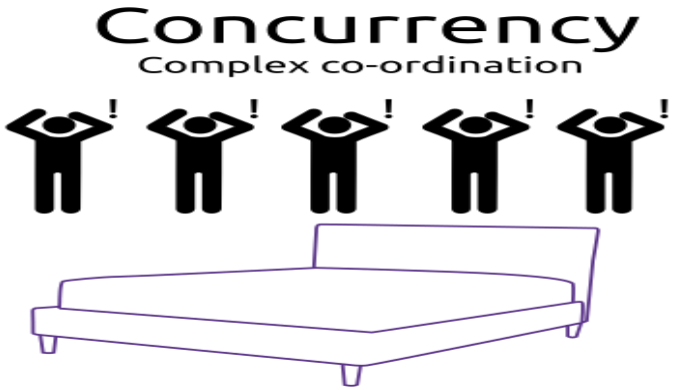
Virtualization

- Make a physical resource look like something else (virtual)
- Why virtualize?
  - To make the computer easier to understand, use, and program
- Examples
  - Process owns a virtual keyboard, monitor, and mouse
  - Process can multi-thread, using multiple virtual CPUs
  - Process can access a single, very large virtual address space
  - Process can access an entire virtual file system

Concurrency

Juggling many tasks together with limited resources

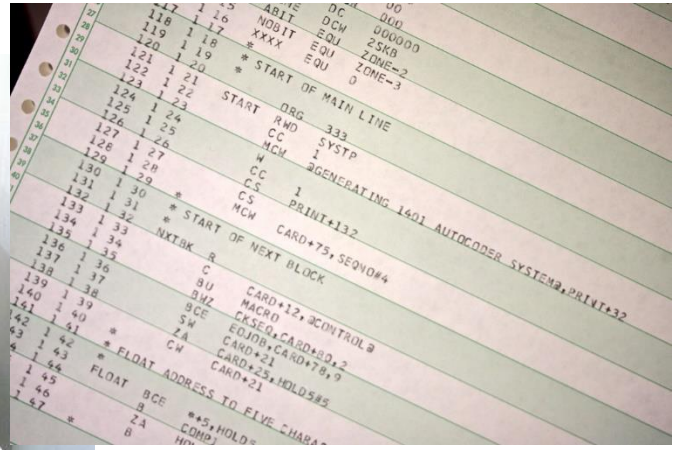
- Examples
  - Many processes share one (or two) CPU's
  - One process may run many threads
  - One OS juggles process execution, system calls, interrupts, exceptions, CPU scheduling, memory management, etc.
- Lot's of concurrency leads to system complexity



Persistency

- Storing Data Forever
- On hard disks, SSDs, CDs, floppy disks, tapes, paper, etc.
- Not just storing raw bytes – Users' want to:
  - Organize data (file systems)
  - Share data (view network or cloud)
  - Access data easily
  - Back-up and recover lost data
  - Protect data from being stolen

Operating System History: Batch



OS History: IBM MVS vs. VM/CMS

MVS

- Large development team
- Top down design
- Behind Schedule
- Very complicated

VM/CMS

- Small, smart developers
- Bottom up design
- Highly virtualized – simple
- Customer participation
- No schedule
- Took half the IBM OS market!

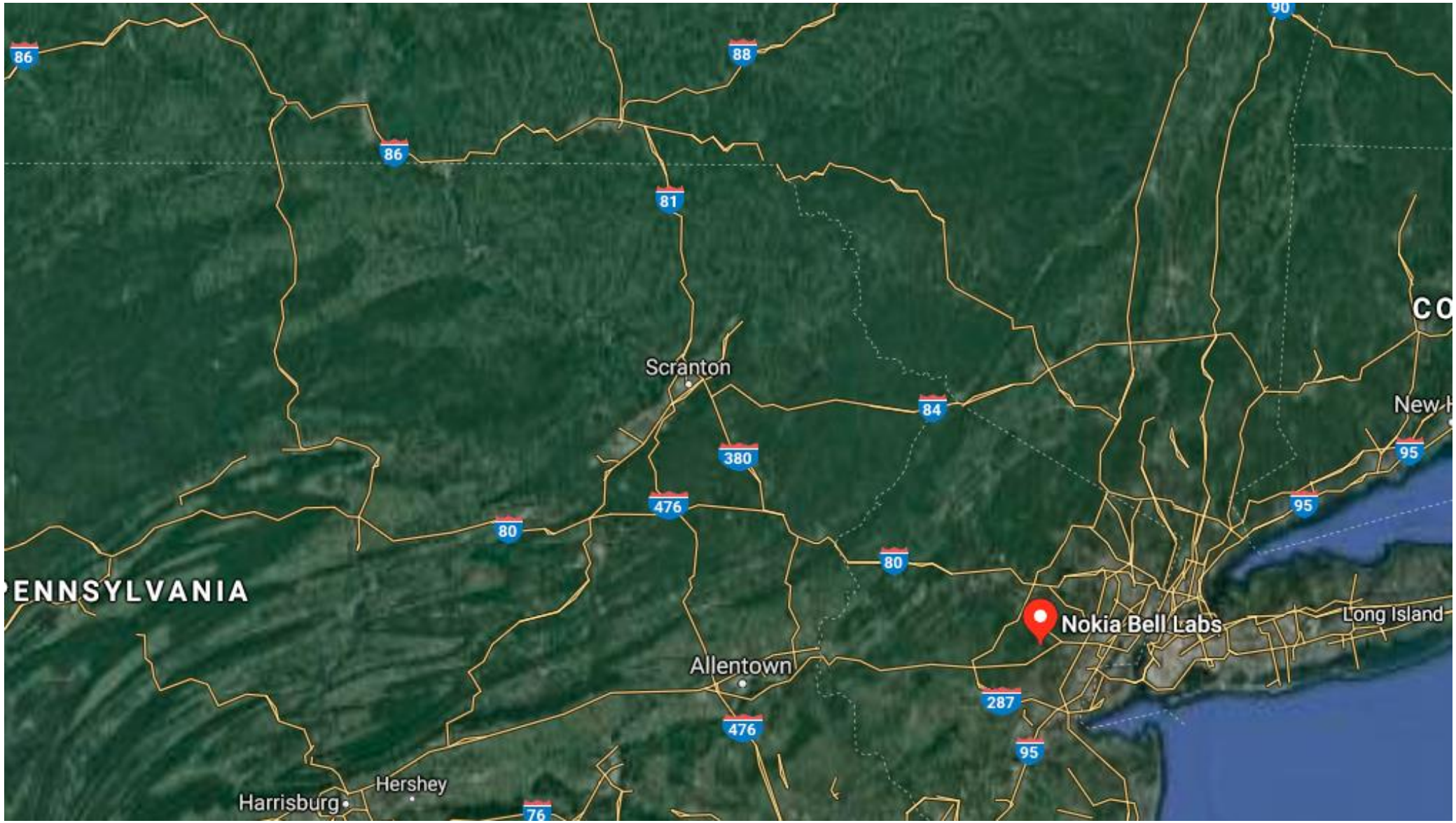
Operating System History: IBM MVT

OS History: DEC PDP 11

- Digital Equipment Corporation
- 1970s – 1990s
- 16 bit address space
- Memory mapped I/O
- Most popular minicomputer
- Compatible with C
- Ran first UNIX in 1970

OS History: Bell Labs, Murray Hill NJ

- Division of AT&T (Ma Bell)
- Moved from NYC in 1967
- Early transistor, laser, photovoltaic cell, radio astronomy
- 9 Nobel prizes
- Claude Shannon: Information Theory
- Invention of C (and C++)
- And UNIX


86
88
86
81
Scranton
84
380
476
80
80
ENNSYLVANIA
Nokia Bell Labs
Long Island
Allentown
287
476
95
Hershey
Harrisburg
76
95
95

OS History : 1960's and 70's

- Concurrency, memory protection, Kernel model, system calls, hardware privilege levels, trap handlings
- Mainframe – Hundreds/Thousands of concurrent users timesharing
- Multics – (1964, Cambridge Mass.) GE/Bell Labs/Honeywell
  - Single level store, Dynamic linking, on-line reconfiguration, multiprocessor, hierarchical file system, symbolic links
- UNIX (1969) – file systems, shells, pipes, C language

OS History: 1980's – the PC era

- Simplified single user operating systems
  - 8 and 16 bit addresses
- DOS first chosen for IBM PC
- Xerox Palo Alto Research Center (PARC) GUI / mouse (1981)
- Windows 1.0 announced in 1983, released 1985
- Mac OS (1984) used PARC technology in Apple Lisa and Macintosh
- IBM OS/2 (1987)
- Many lessons from mainframe concurrency were forgotten and had to be painfully relearned

OS History: 1980's Unix Fragmentation

- Bell labs licensed UNIX to government and academic organizations
  - Gained popularity
- Several start-ups and large companies licensed UNIX and wrote their own "proprietary" versions of UNIX
  - IBM – AIX, Sun – Solaris, Hewlitt-Packard – HP/UX, Apple – macOS
  - Legal issues abound (IP, copyrights)
  - Used for "Workstation" computers – professional "PC"s

OS History: 1990's / 2000's

- UNIX standardization (BSD, POSIX)
- Open Source (GNU, Linux)
  - Led the way to modern OS and cloud platforms
- Wider adoption of threads and parallelism

- Mobile Device OS
  - Android, iOS
- Hypervisors
  - VMWare ESX, Xen, Linux/KVM etc.