CS 310 Operating Systems

Lecture 1: Operating System Introduction

Ravi Mittal
IIT Goa

Acknowledgements !

- Contents of this class presentation have been taken from various sources. Thanks are due to the original content creators:
  - Class presentation: University of California, Berkeley: David Culler, Anthony D. Joseph, John Kubiatowicz, AJ Shankar, George Necula, Alex Aiken, Eric Brewer, Ras Bodik, Ion Stoica, Doug Tygar, and David Wagner
  - Book: Operating Systems: Principles and Practice (2nd Edition) Anderson and Dahlin, Volume 1

Read the following:

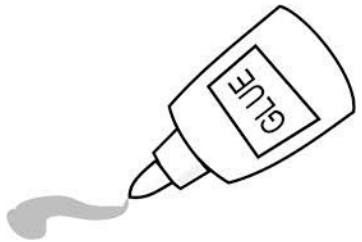
- Operating Systems: Principles and Practice (2nd Edition) Anderson and Dahlin
  - Volume 1, Kernel and Processes
    - Section: 1.1, 1.2, and 1.3

We will study..

- What is an Operating System?

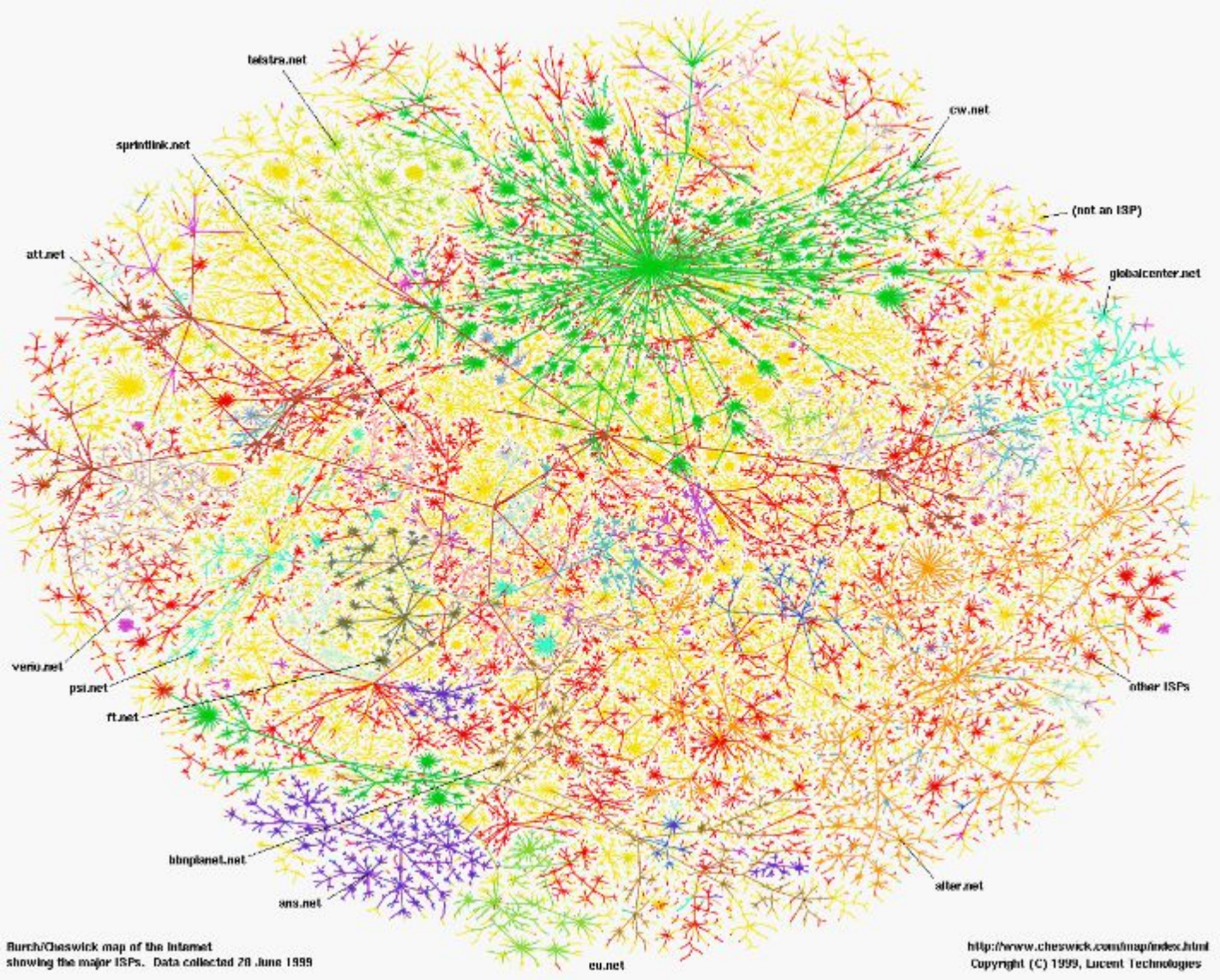
- OS Role: An Illusionist
- OS Role: A Referee

- OS Role: A Glue



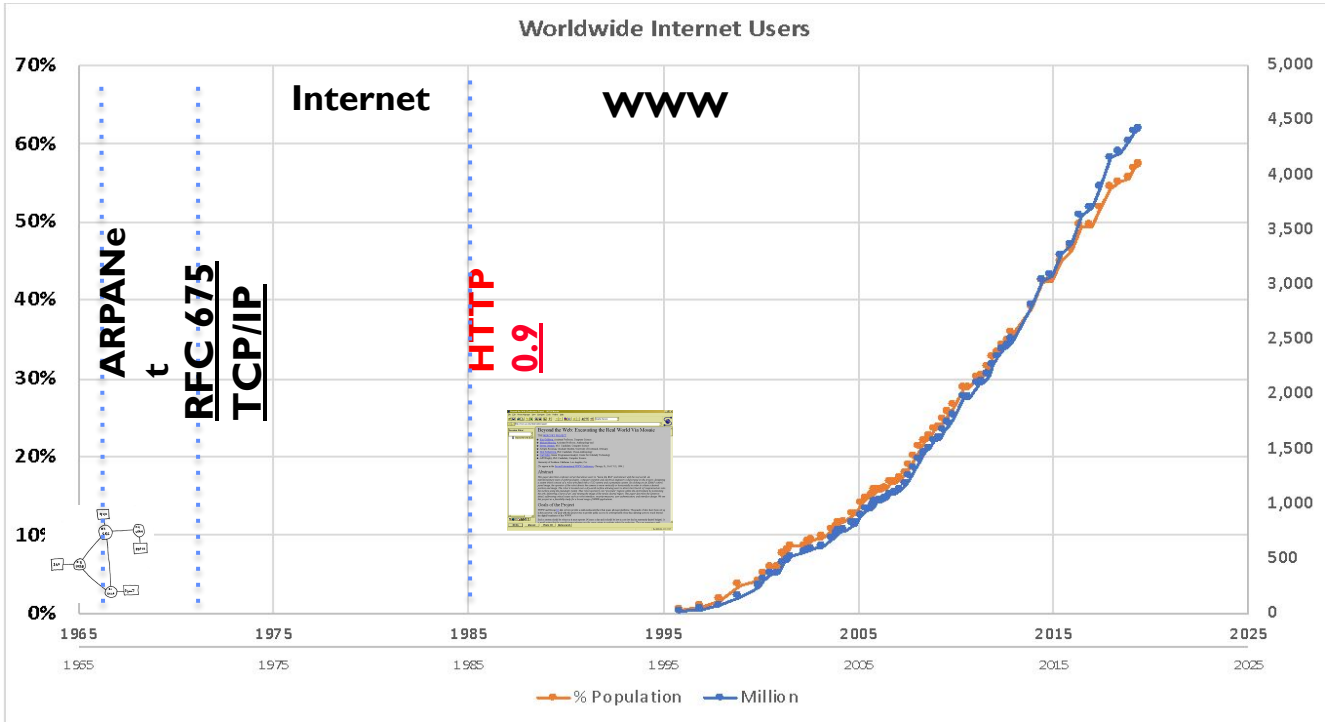
What is an Operating System?

Greatest Artifact of Human Civilization…



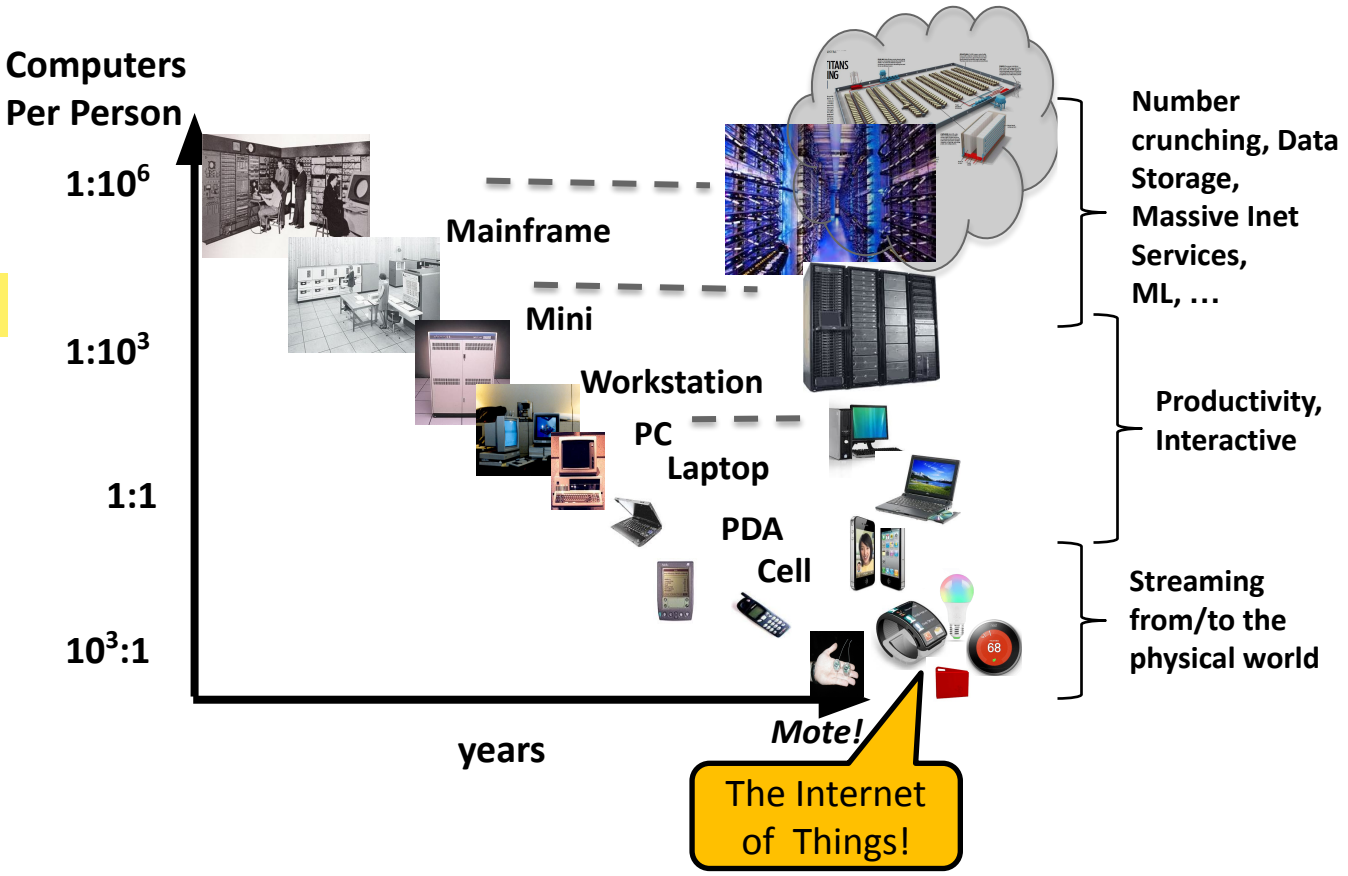
Greatest Artifact of Human Civilization…

Running Systems at Internet Scale



Across Incredible Diversity

Bell's Law: New computer class every 10 years



And Range of Timescales

"Numbers that Everyone Should Know"

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	25 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	3,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	250,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from disk	20,000,000 ns
Send packet CA->Netherlands->CA	150,000,000 ns

One source of complexity in Computers: Dealing with components that are widely diverse in functionality and timing

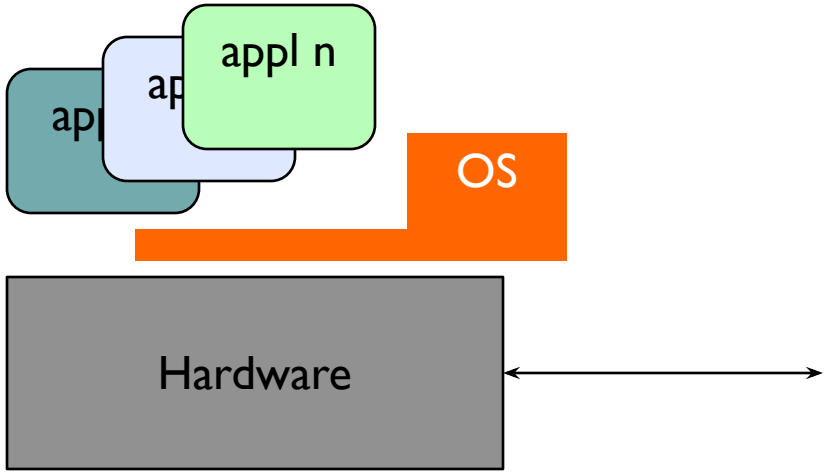
Operating Systems are at the Heart of it All!

- Make the incredible advance in the underlying technology available to a rapidly evolving body of applications
  - Provide consistent abstractions to applications, even on different hardware
  - Manage sharing of resources among multiple applications
- The key building blocks:
  - Processes
  - Threads, Concurrency, Scheduling, Coordination
  - Address Spaces
  - Protection, Isolation, Sharing, Security
  - Communication, Protocols
  - Persistent storage, transactions, consistency, resilience
  - Interfaces to all devices

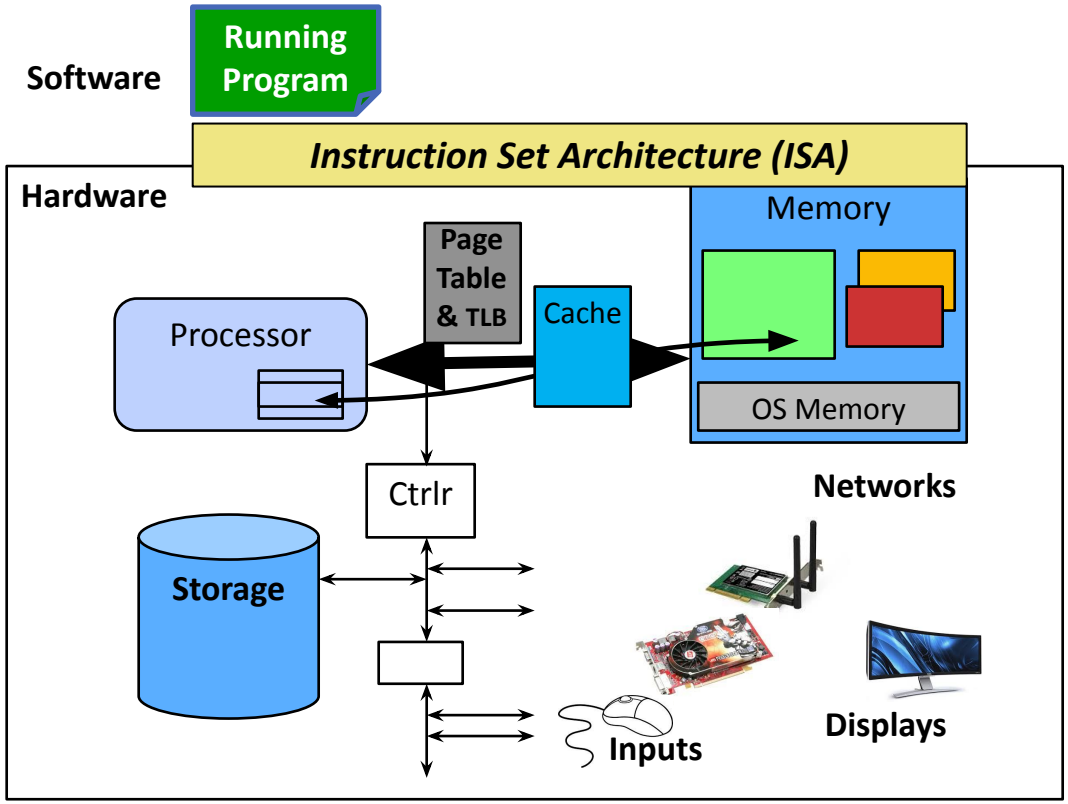
But: What is an operating system?

What is an operating system?

- Special layer of software that provides application software access to hardware resources
  - Convenient abstraction of complex hardware devices
  - Protected access to shared resources
  - Security and authentication
  - Communication amongst logical entities



Hardware/Software Interface



What you learned in CS 211 – Computer Architecture

The OS *abstracts* these hardware details from the application

What does an Operating System do?

- We know:
  - Memory Management
  - I/O Management
  - CPU Scheduling
  - Communication support?
  - Multitasking/multiprogramming
- What about?
  - File System?
  - Multimedia Support?
  - User Interface?

Definition of an Operating System

- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is good approximation
  - But varies wildly

- The one program running at all times on the computer is the kernel
  - Everything else is either a system program (ships with the operating system) or an application program

What is an Operating System?

Role 1: Illusionist

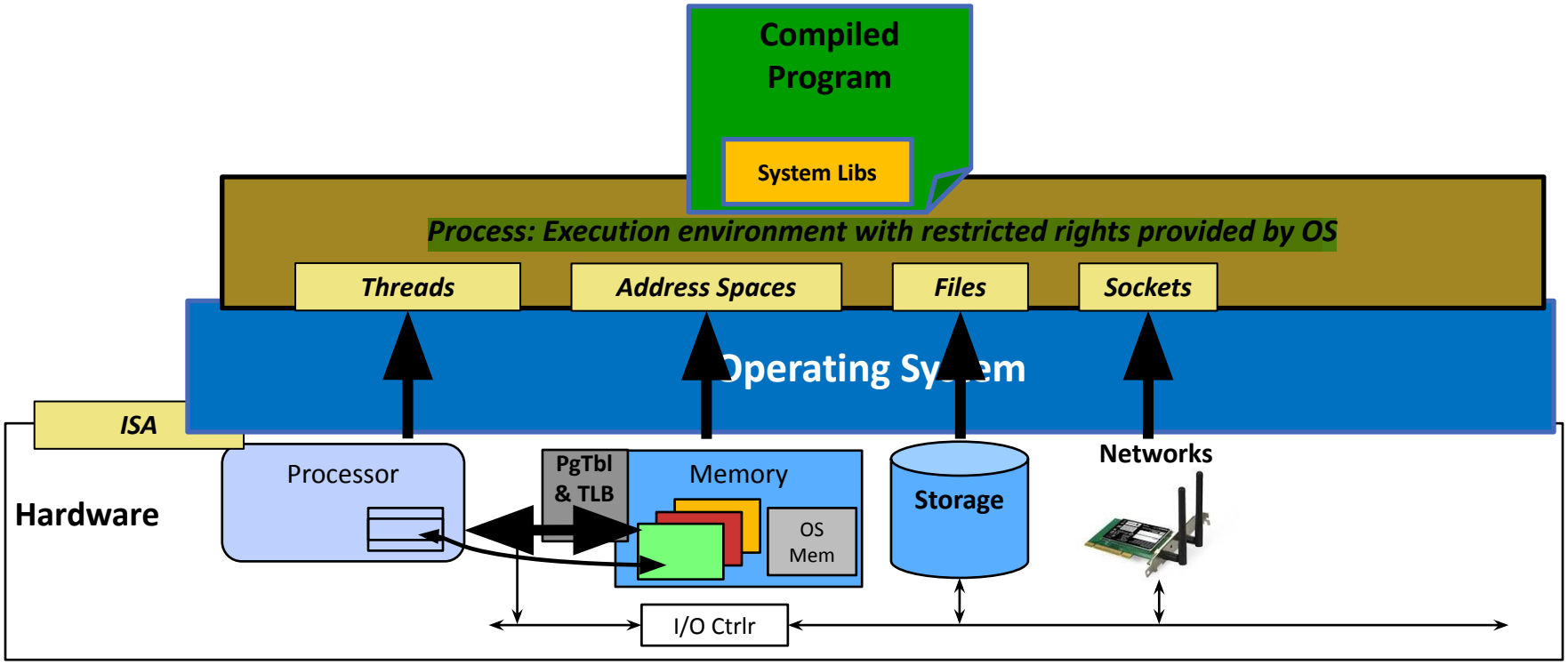
Illusionist

- **It must provide illusion of**
  - clean, easy-to-use abstractions of physical resources
    - Infinite memory, dedicated machine
    - Higher level objects: files, users, messages
    - Masking limitations, virtualization

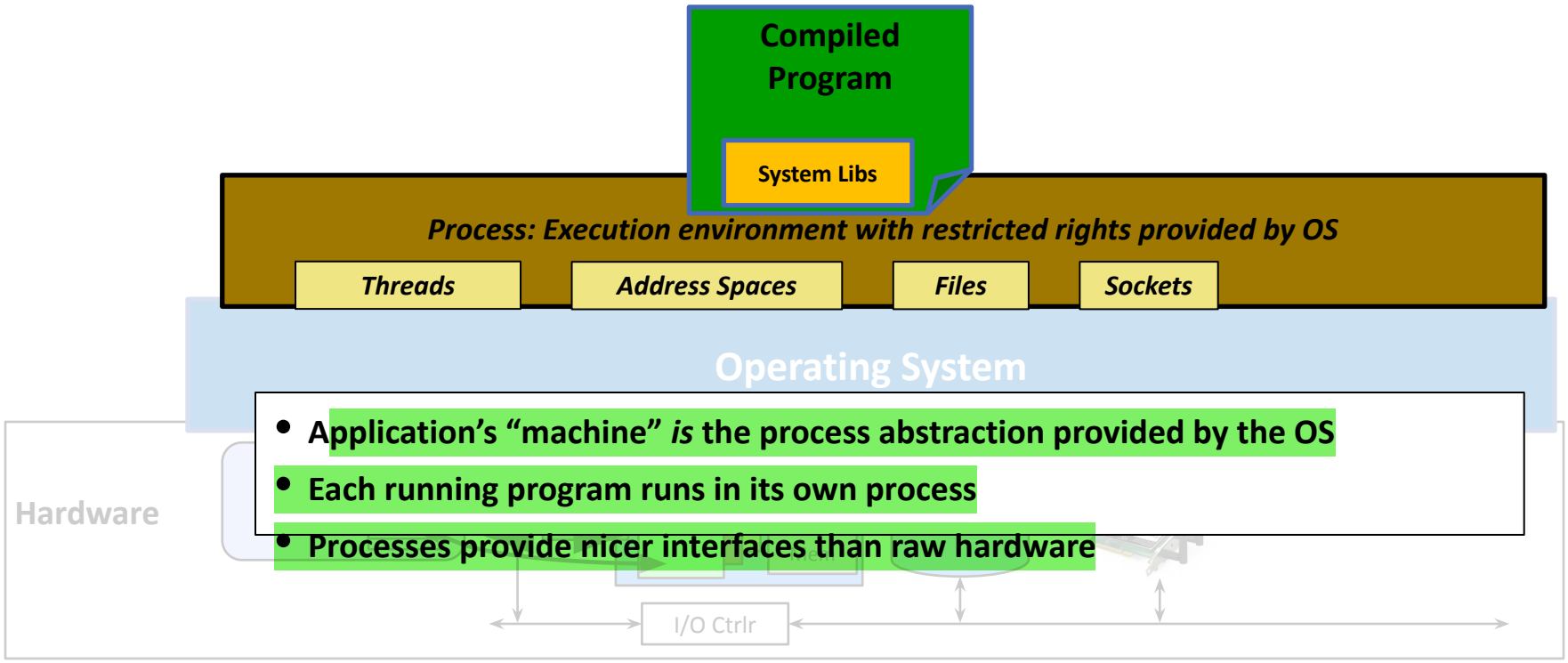
What is an Operating System?

An Illusionist !

OS Basics: Virtualizing the Machine

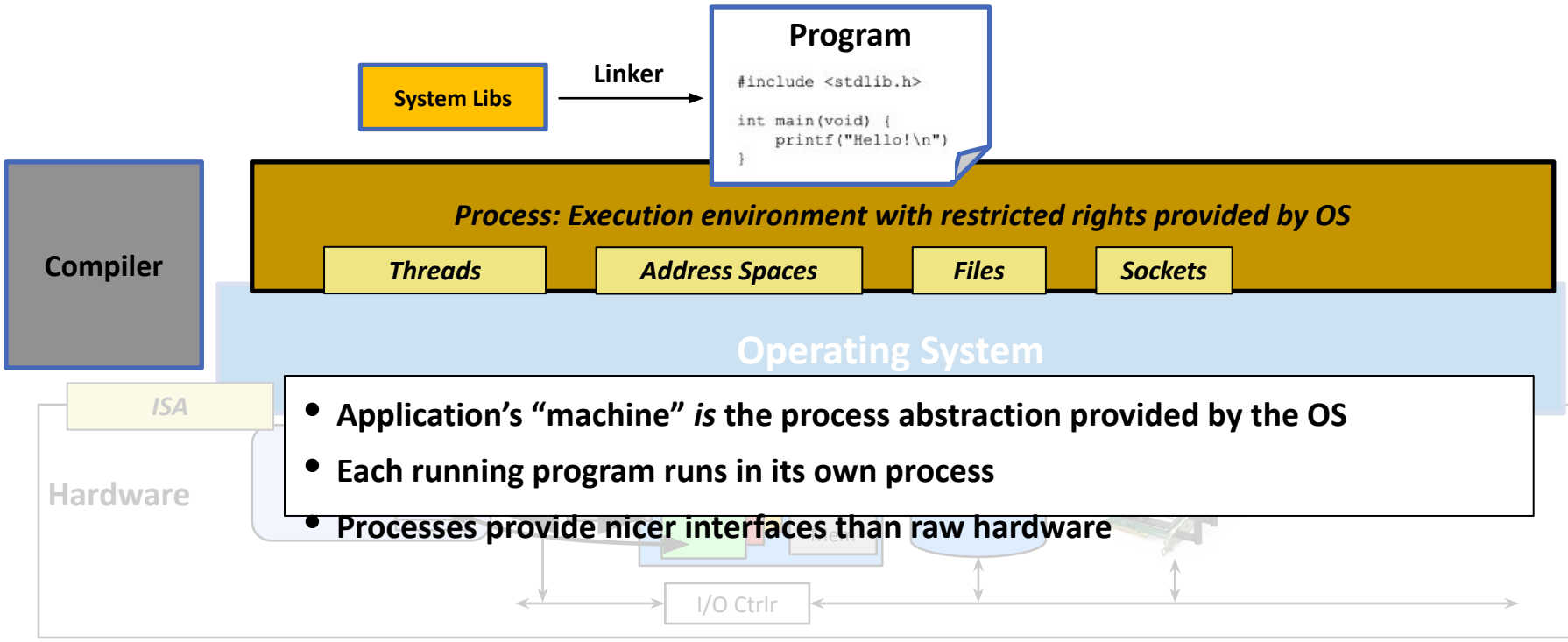


Compiled Program's View of the World



- Application's "machine" *is* the process abstraction provided by the OS
- Each running program runs in its own process
- Processes provide nicer interfaces than raw hardware

System Programmer's View of the World



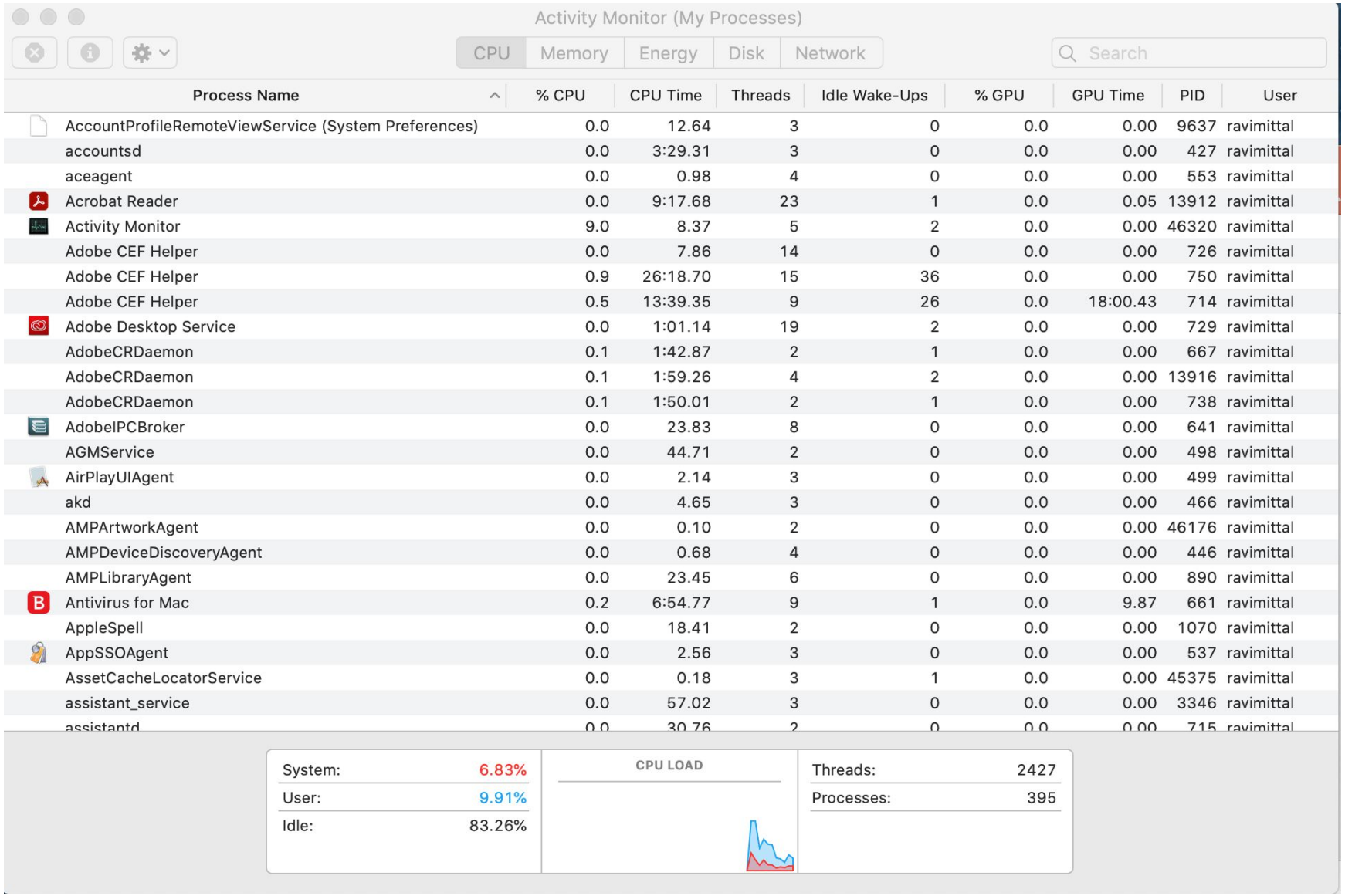
- Application's "machine" *is* the process abstraction provided by the OS
- Each running program runs in its own process
- Processes provide nicer interfaces than raw hardware

What's in a Process?

A process consists of:

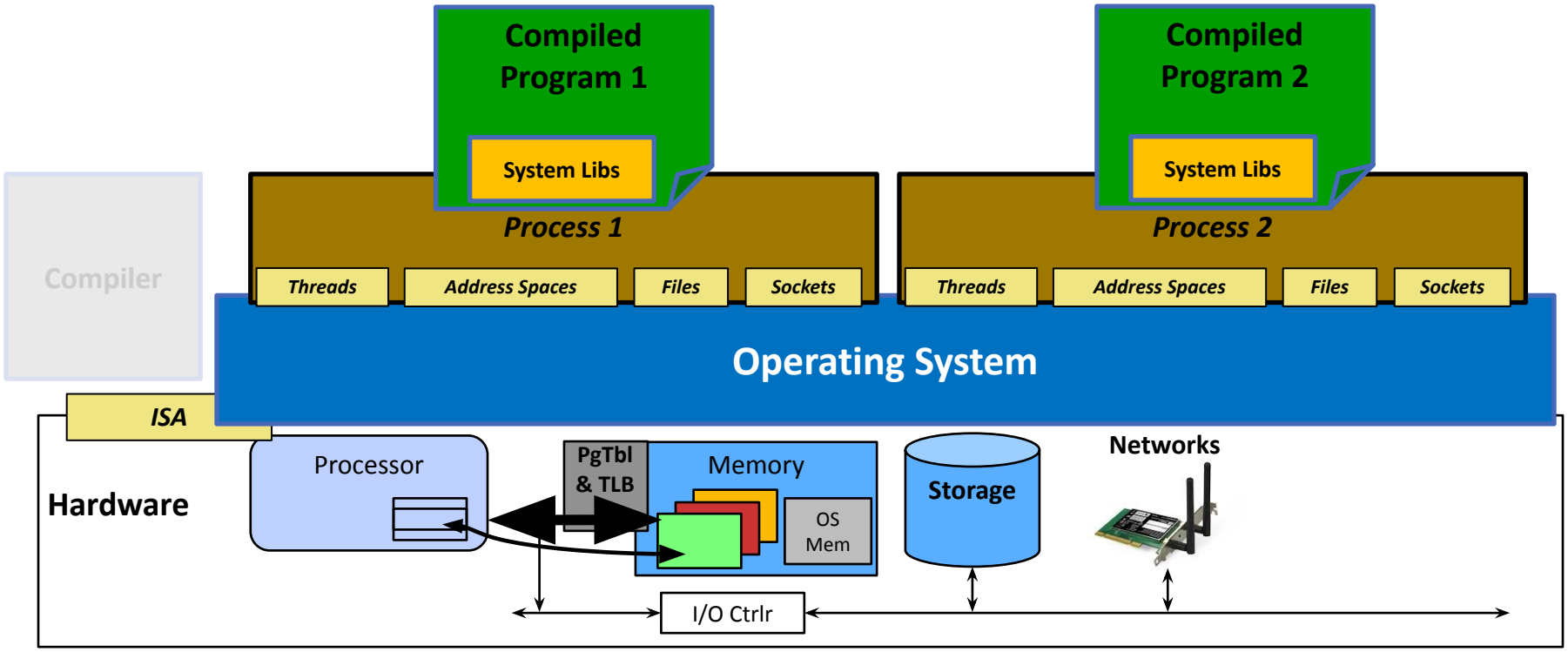
- Address Space
- One or more threads of control executing in that address space
- Additional system state associated with it
  - Open files
  - Open sockets (network connections)
  - …
- OS creates and runs process

For Example…Processes running on my laptop

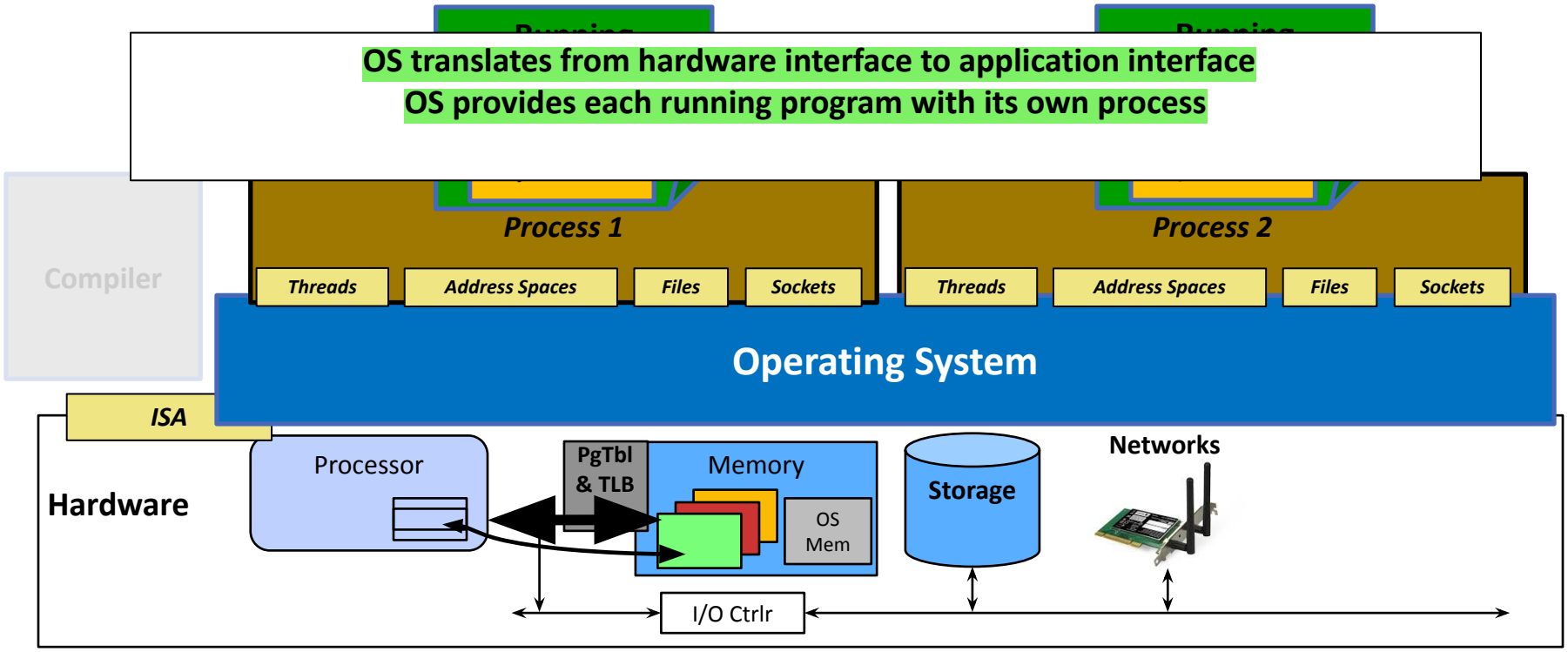


Process Name	% CPU	CPU Time	Threads	Idle Wake-Ups	% GPU	GPU Time	PID	User
AccountProfileRemoteViewService (System Preferences)	0.0	12.64	3	0	0.0	0.00	9637	ravimittal
accountsd	0.0	3:29.31	3	0	0.0	0.00	427	ravimittal
aceagent	0.0	0.98	4	0	0.0	0.00	553	ravimittal
Acrobat Reader	0.0	9:17.68	23	1	0.0	0.05	13912	ravimittal
Activity Monitor	9.0	8.37	5	2	0.0	0.00	46320	ravimittal
Adobe CEF Helper	0.0	7.86	14	0	0.0	0.00	726	ravimittal
Adobe CEF Helper	0.9	26:18.70	15	36	0.0	0.00	750	ravimittal
Adobe CEF Helper	0.5	13:39.35	9	26	0.0	18:00.43	714	ravimittal
Adobe Desktop Service	0.0	1:01.14	19	2	0.0	0.00	729	ravimittal
AdobeCRDaemon	0.1	1:42.87	2	1	0.0	0.00	667	ravimittal
AdobeCRDaemon	0.1	1:59.26	4	2	0.0	0.00	13916	ravimittal
AdobeCRDaemon	0.1	1:50.01	2	1	0.0	0.00	738	ravimittal
AdobeIPCBroker	0.0	23.83	8	0	0.0	0.00	641	ravimittal
AGMService	0.0	44.71	2	0	0.0	0.00	498	ravimittal
AirPlayUIAgent	0.0	2.14	3	0	0.0	0.00	499	ravimittal
akd	0.0	4.65	3	0	0.0	0.00	466	ravimittal
AMPArtworkAgent	0.0	0.10	2	0	0.0	0.00	46176	ravimittal
AMPDeviceDiscoveryAgent	0.0	0.68	4	0	0.0	0.00	446	ravimittal
AMPLibraryAgent	0.0	23.45	6	0	0.0	0.00	890	ravimittal
Antivirus for Mac	0.2	6:54.77	9	1	0.0	9.87	661	ravimittal
AppleSpell	0.0	18.41	2	0	0.0	0.00	1070	ravimittal
AppSSOAgent	0.0	2.56	3	0	0.0	0.00	537	ravimittal
AssetCacheLocatorService	0.0	0.18	3	1	0.0	0.00	45375	ravimittal
assistant_service	0.0	57.02	3	0	0.0	0.00	3346	ravimittal
assistantd	0.0	30.76	2	0	0.0	0.00	715	ravimittal

Operating System's View of the World



Operating System's View of the World



What is an Operating System?

A Referee!

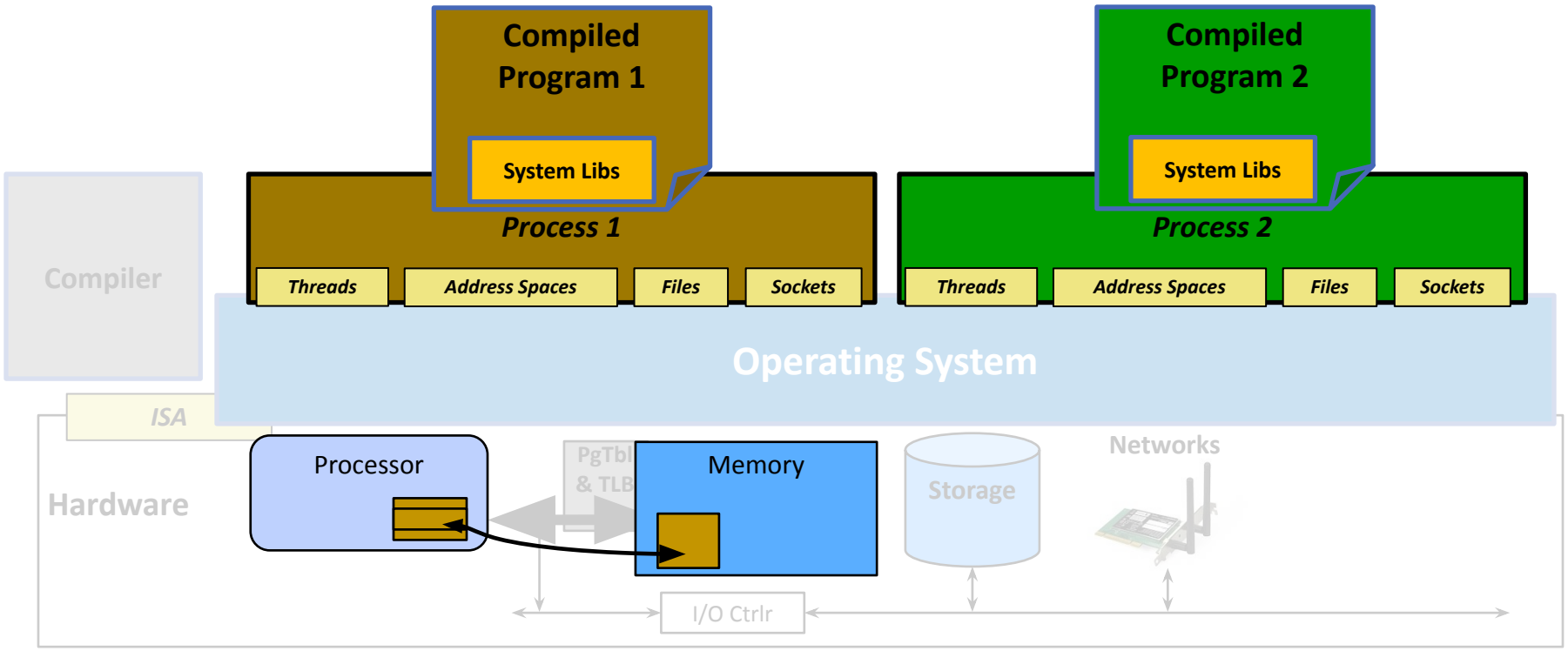
What is an Operating System?

Role 2: Referee

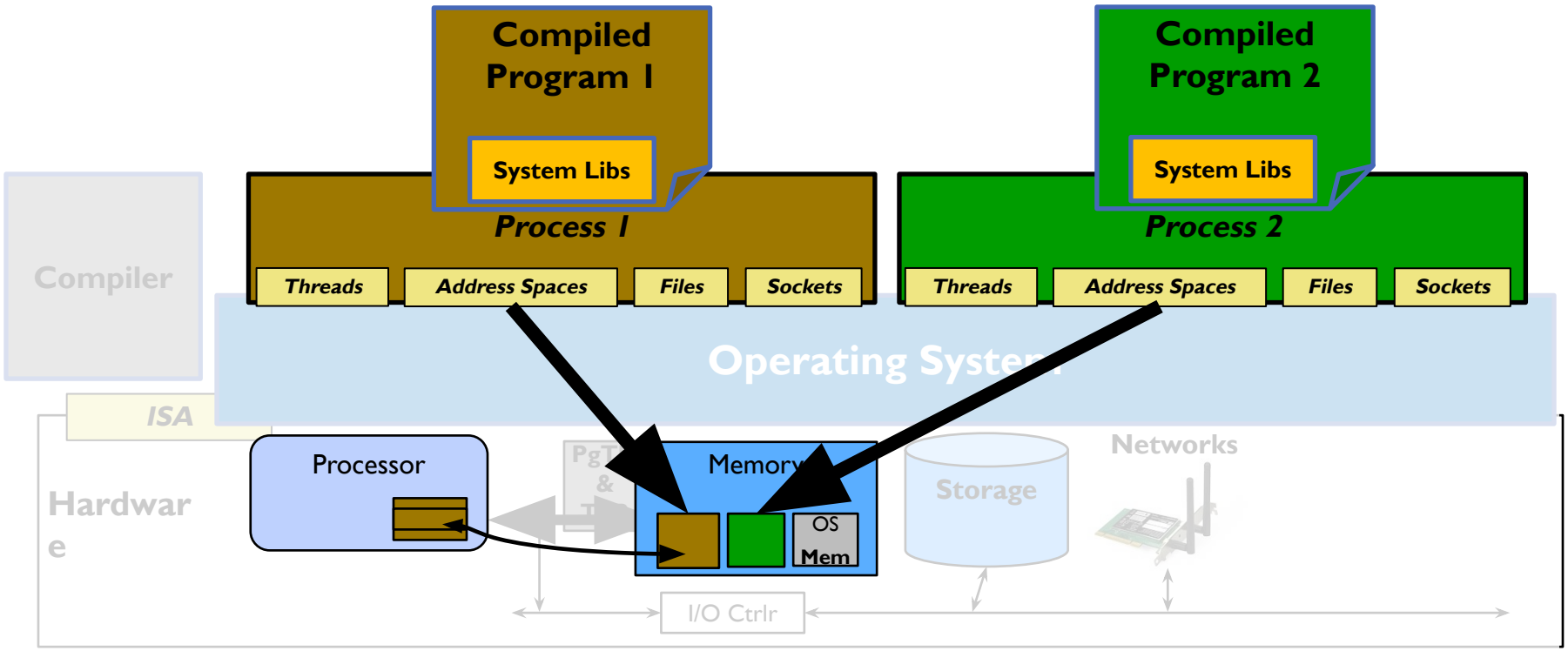
Referee

- **It must**
  - Manage protection, isolation, and sharing of resources
    - Resource allocation and communication
- It creates and runs processes

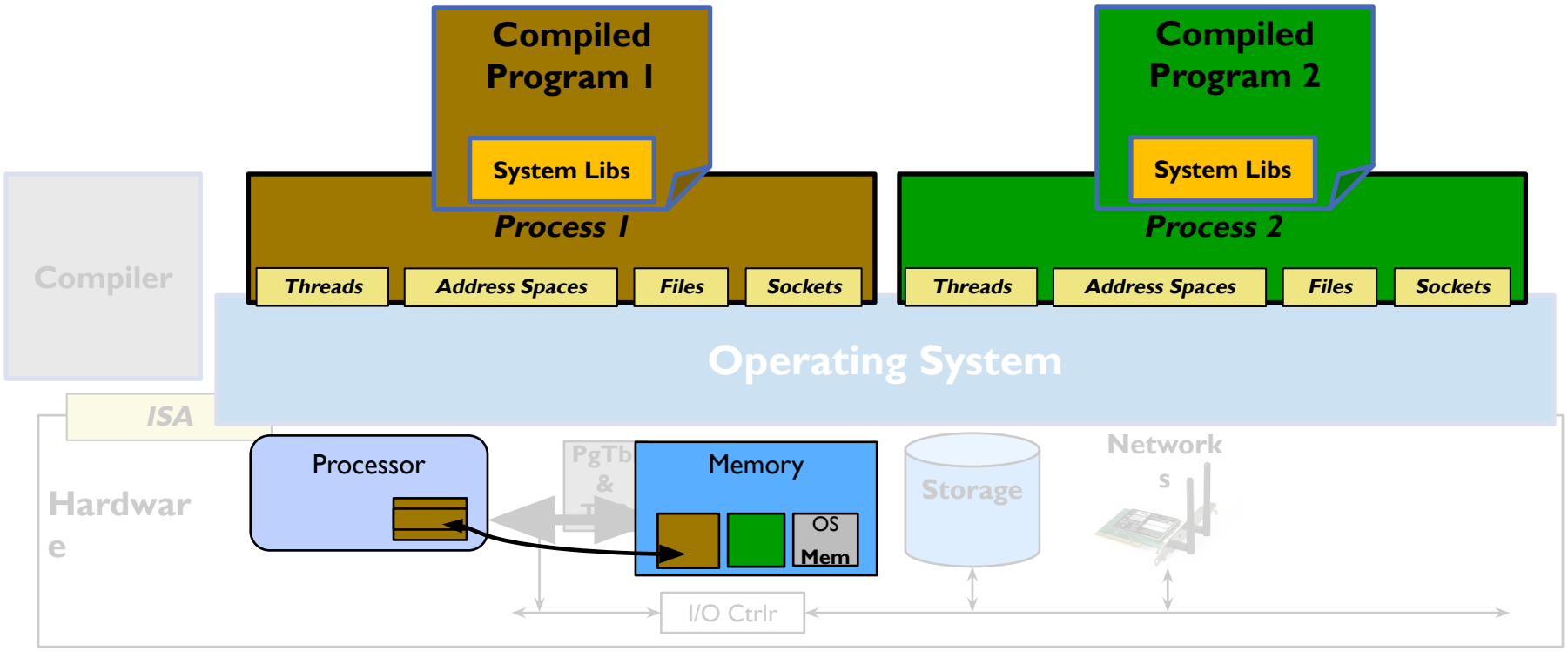
OS Basics: Running a Process



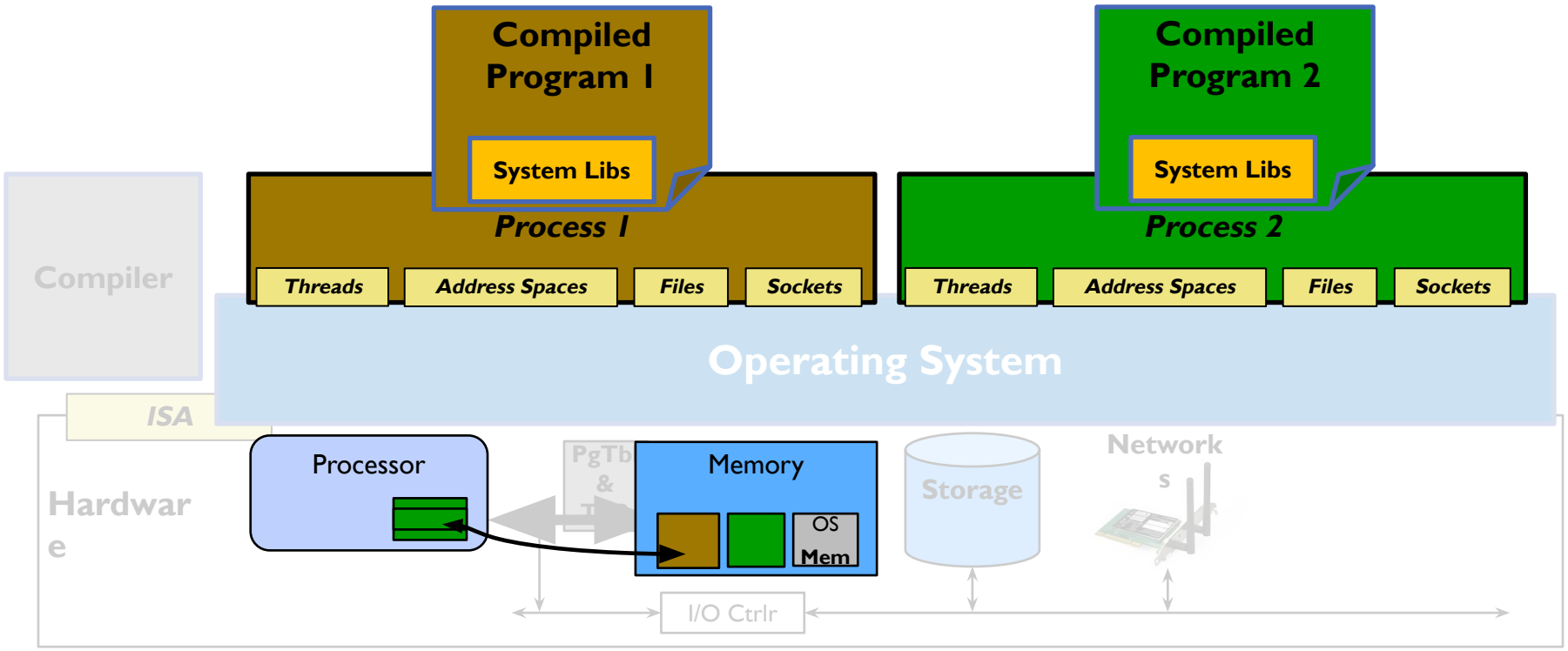
OS Basics: Switching Processes



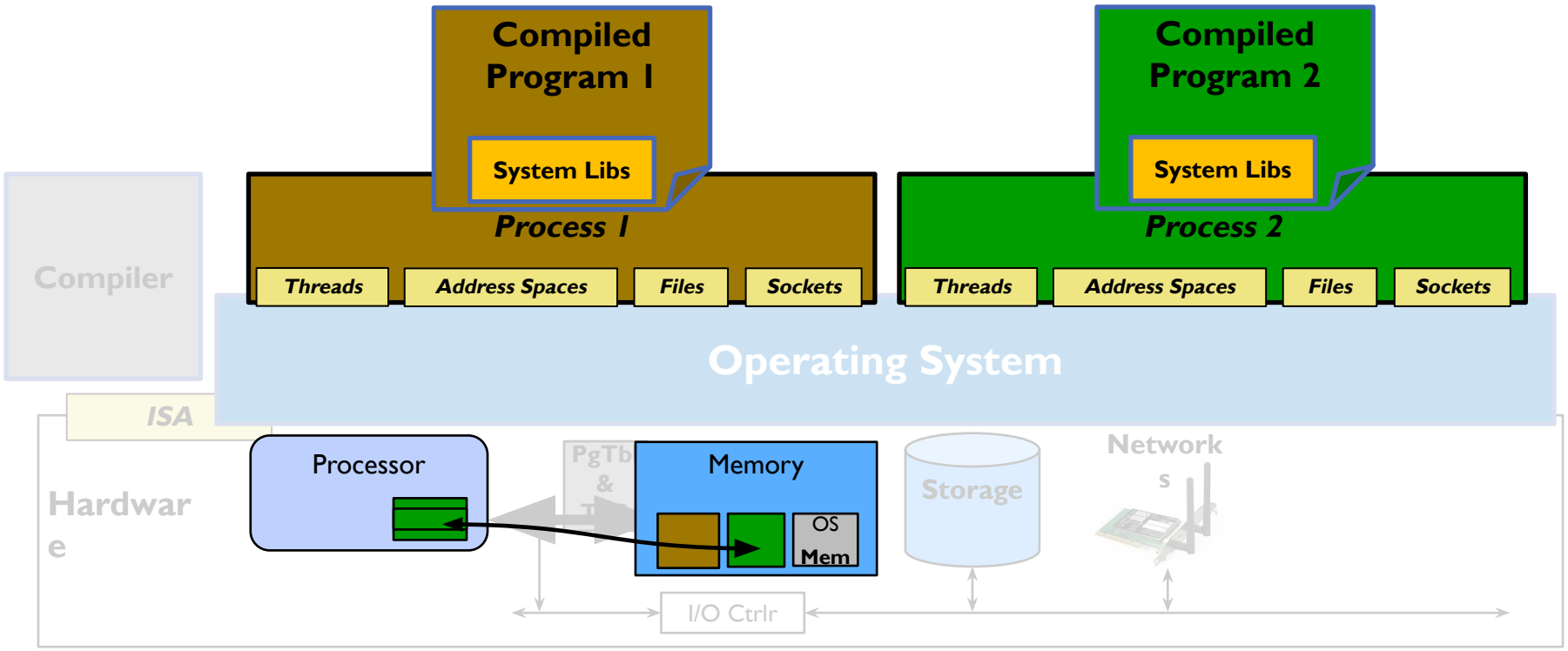
OS Basics: Switching Processes



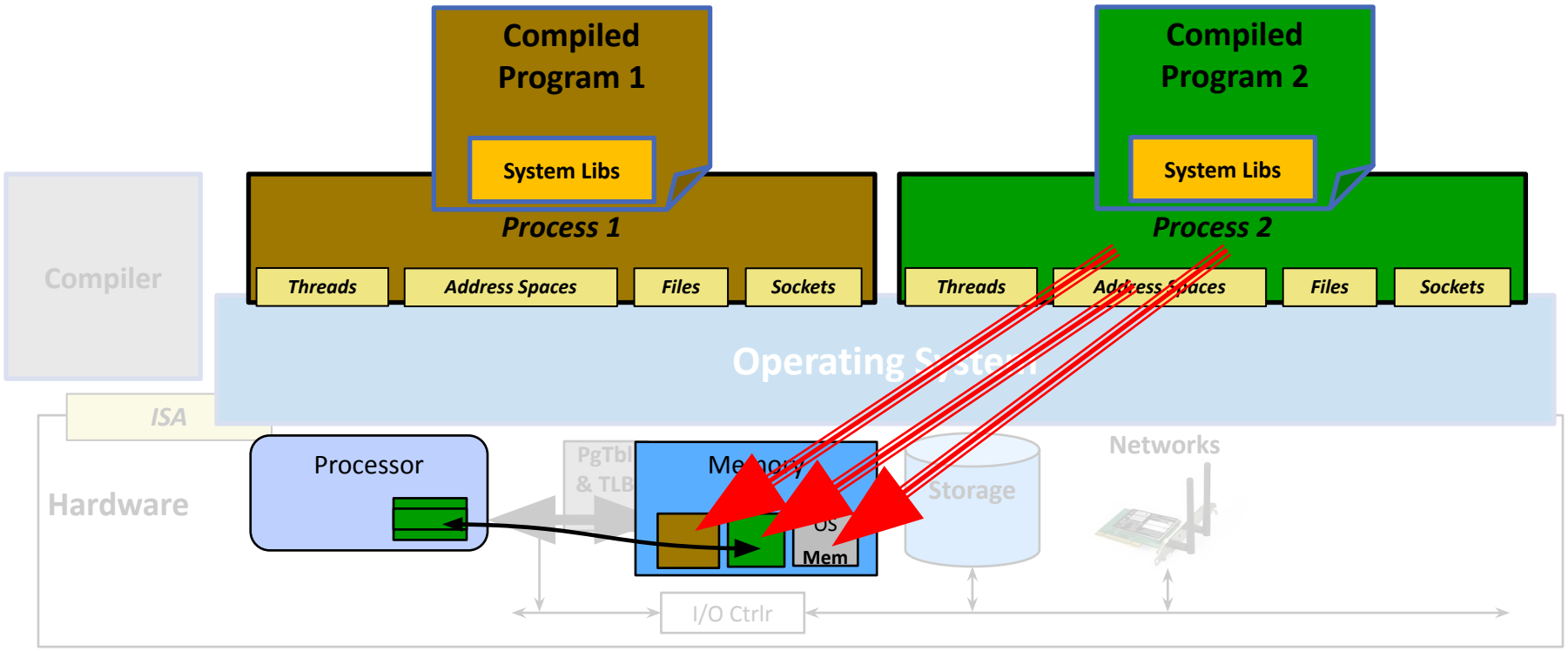
OS Basics: Switching Processes



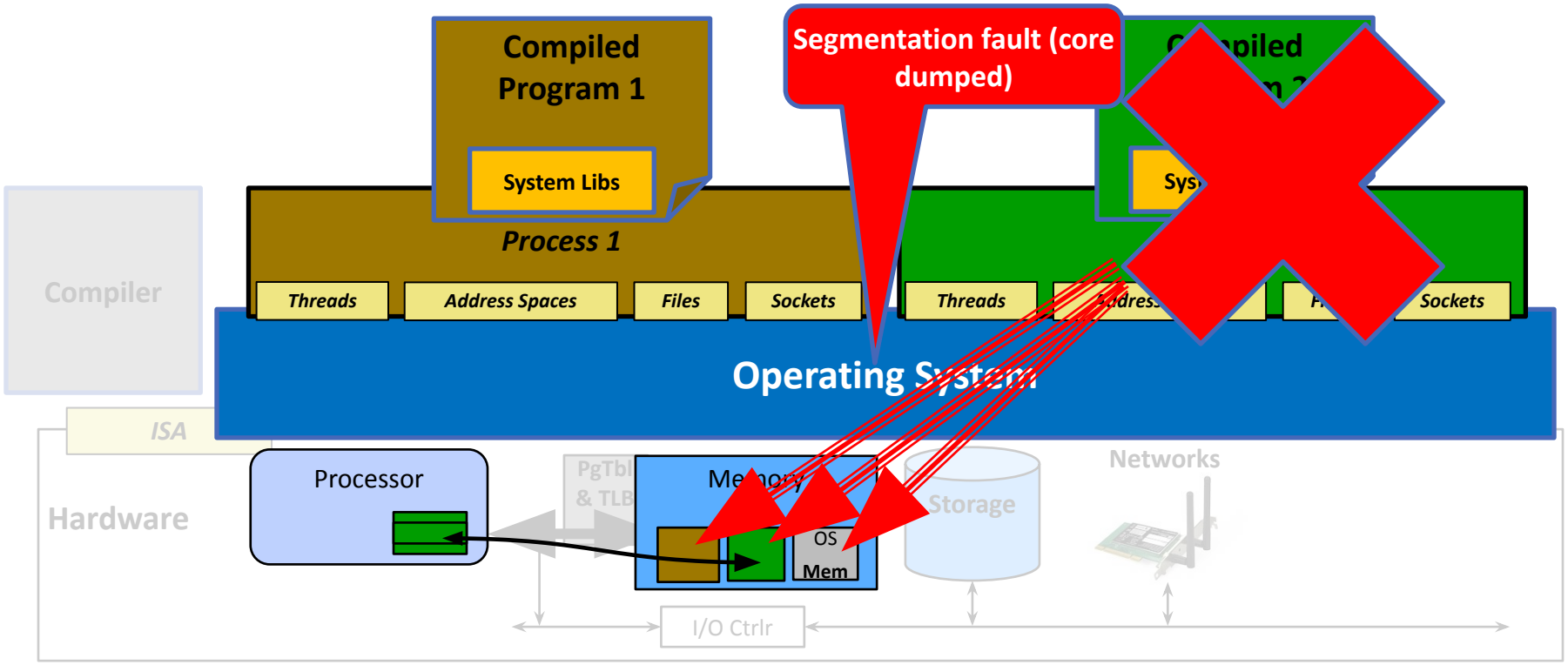
OS Basics: Switching Processes



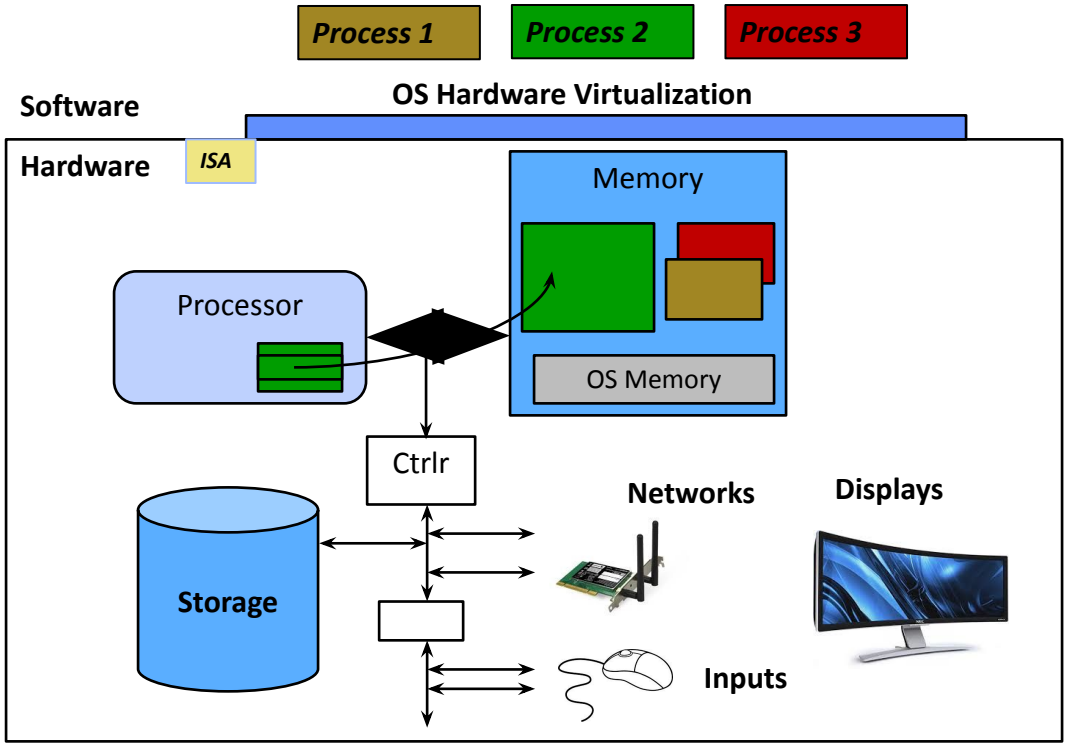
OS Basics: Protection



OS Basics: Protection



OS Basics: Protection

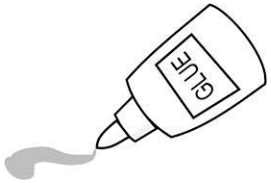


- OS isolates processes from each other
- OS isolates itself from other processes
- … even though they are actually running on the same hardware!

What is an Operating System?

A Glue !

What is an Operating System?
Role 3: Glue

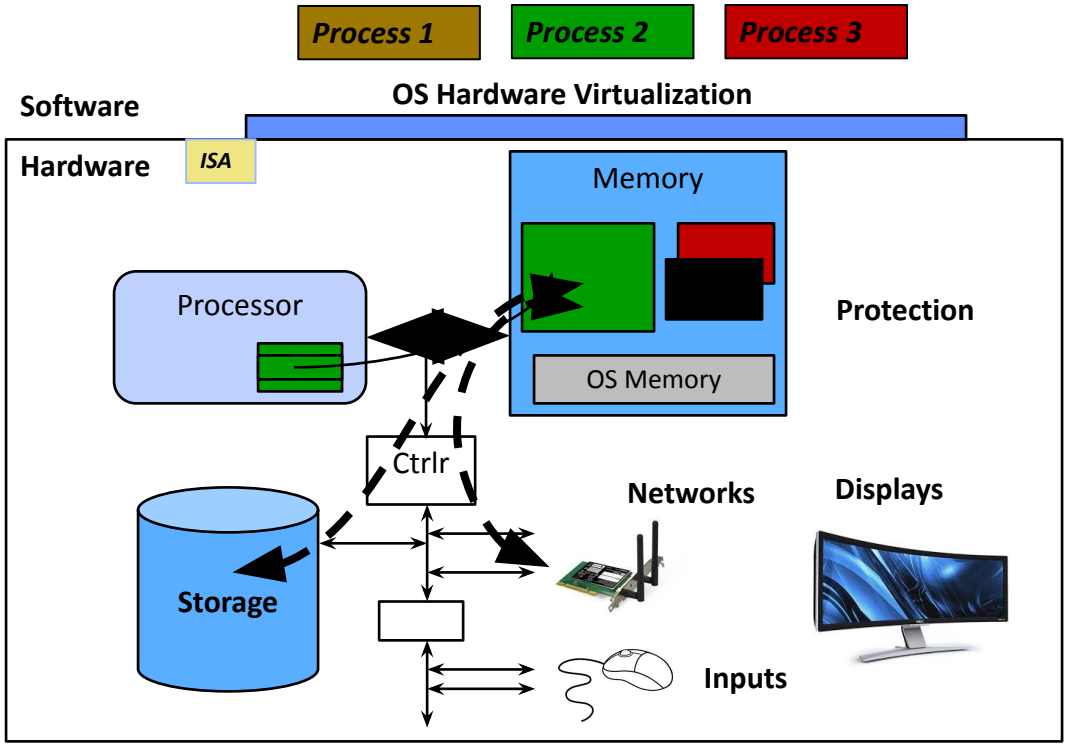


- OS plays a key role in providing a set of common and standard services to applications
- It provides
  - The same look Look and feel
  - Common features like CTRL-C, CTRL-V
- Example: A webserver must be able to read file that a text editor wrote
  - Standard file format
  - Standard file and directory system
  - Standard way of applications to pass messages and share memories

Example: File system in an OS

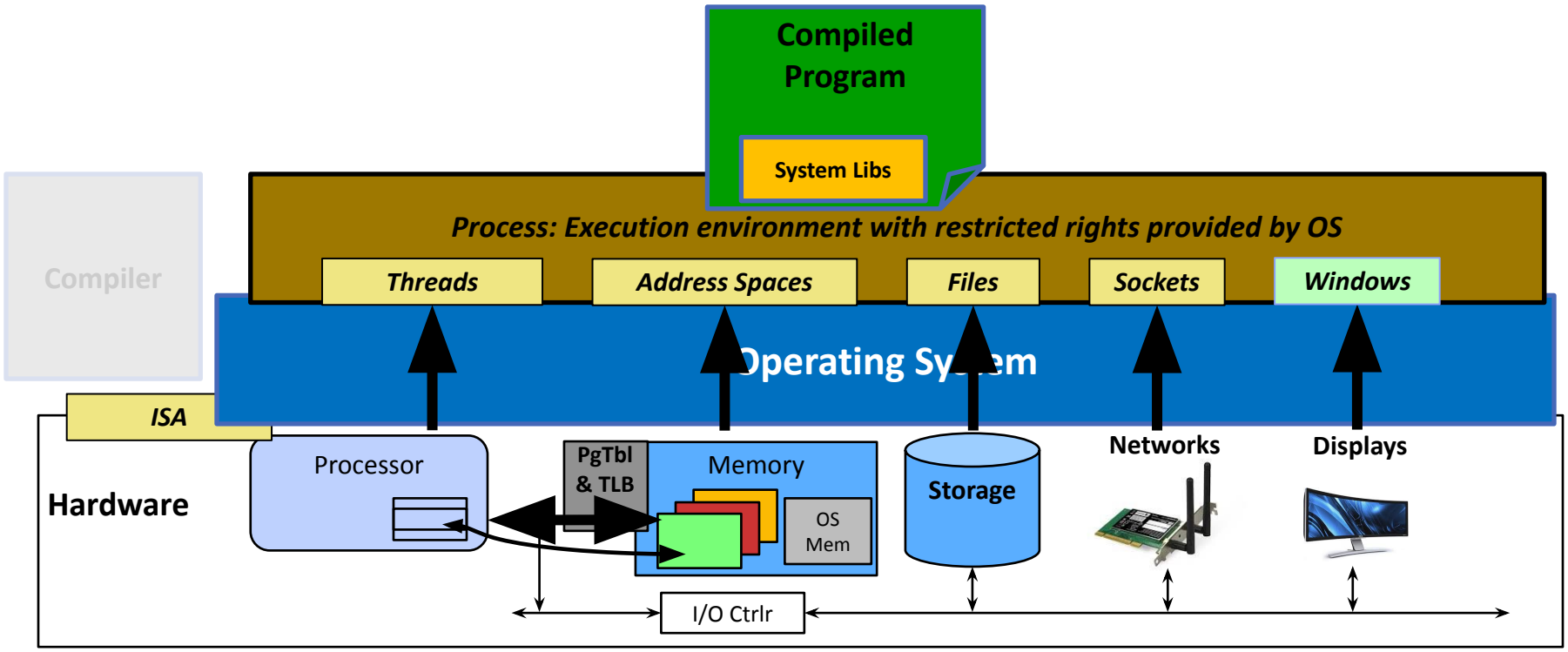
- Referee
  - Prevents others from accessing the file system without permission
  - Reuse storage space after files are deleted
- Illusionist
  - File grows and shrinks without user knowing about it
    - File can exist on RAID or multiple storage devices
      - Invisible to user
    - Files persist even during certain hardware faults
- Glue
  - Directories
  - Standard APIs for file I/O

OS Basics: I/O

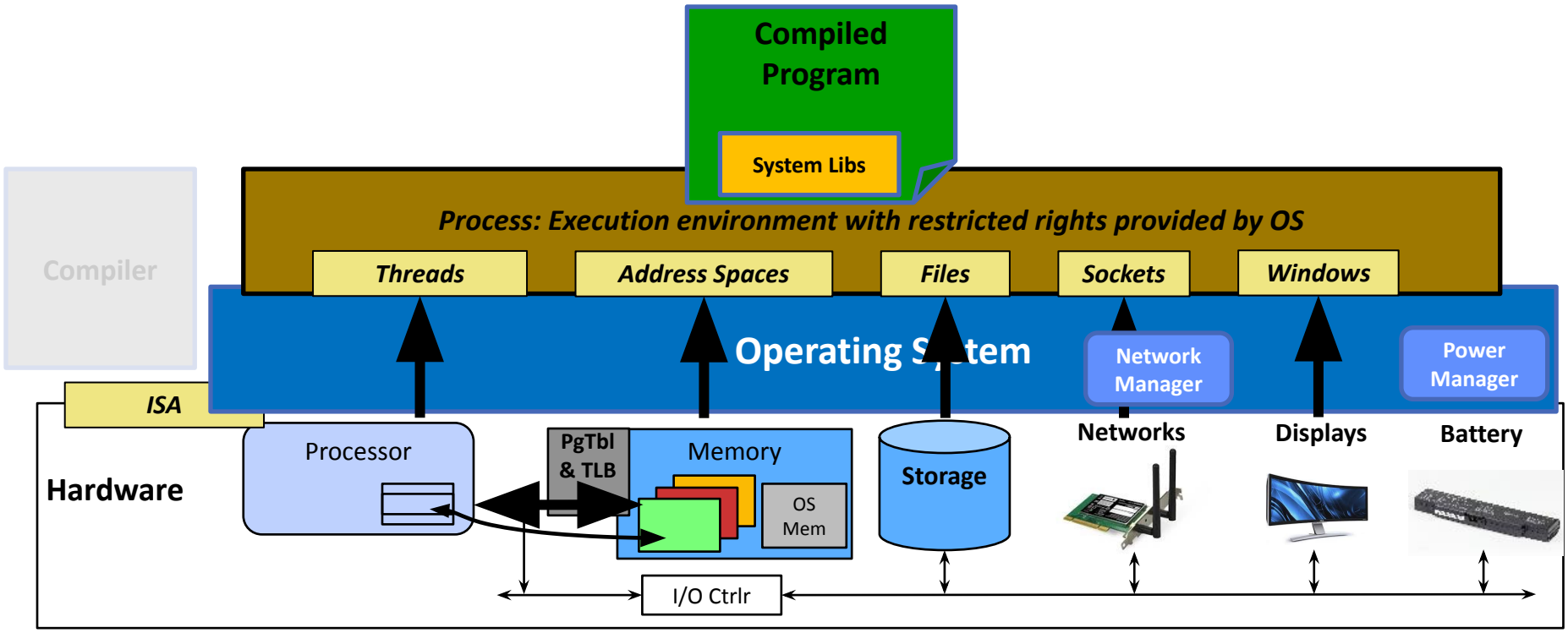


- OS provides common services in the form of I/O

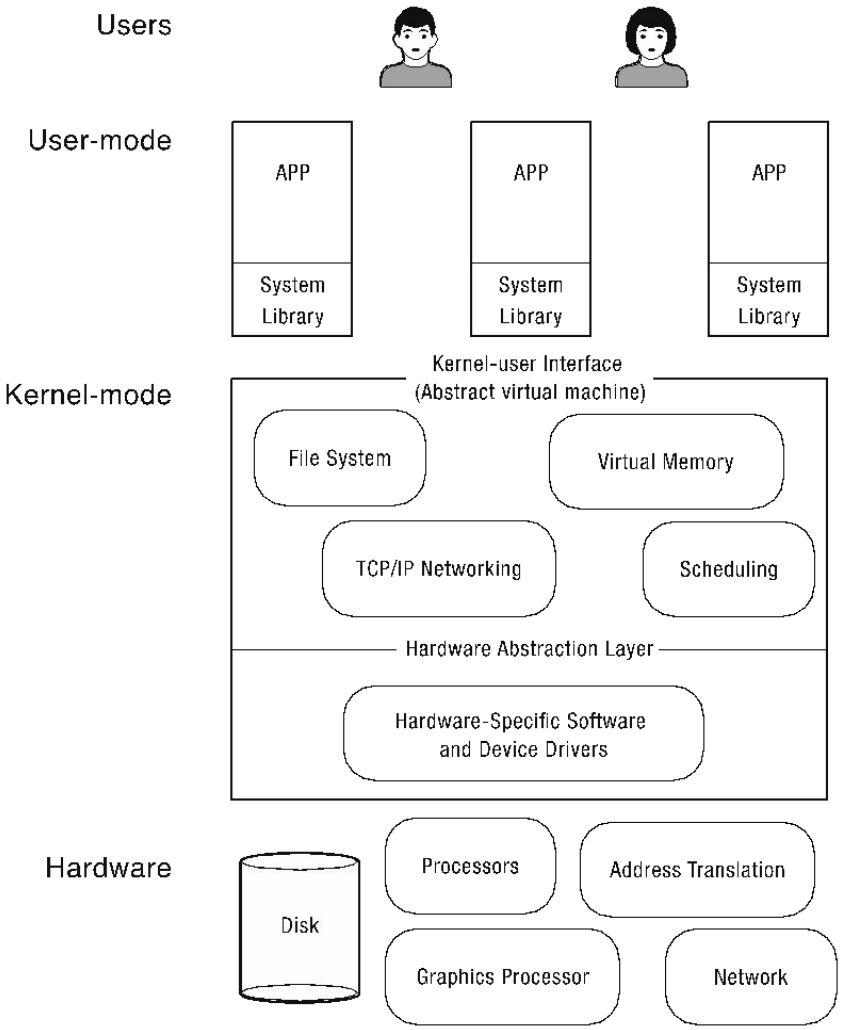
OS Basics: Look and Feel



OS Basics: Background Management



Structure of General Purpose OS

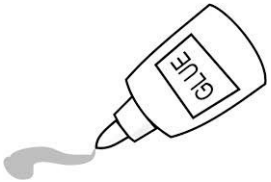


Book: OS Principles and Practices: Anderson and Dahlin

Summary: What is an Operating System?

- Referee
  - Manage protection, isolation, and sharing of resources
    - Resource allocation and communication

- Illusionist
  - Provide clean, easy-to-use abstractions of physical resources
    - Infinite memory, dedicated machine
    - Higher level objects: files, users, messages
    - Masking limitations, virtualization



- Glue
  - Common services
    - Storage, Window system, Networking
    - Sharing, Authorization
    - Look and feel