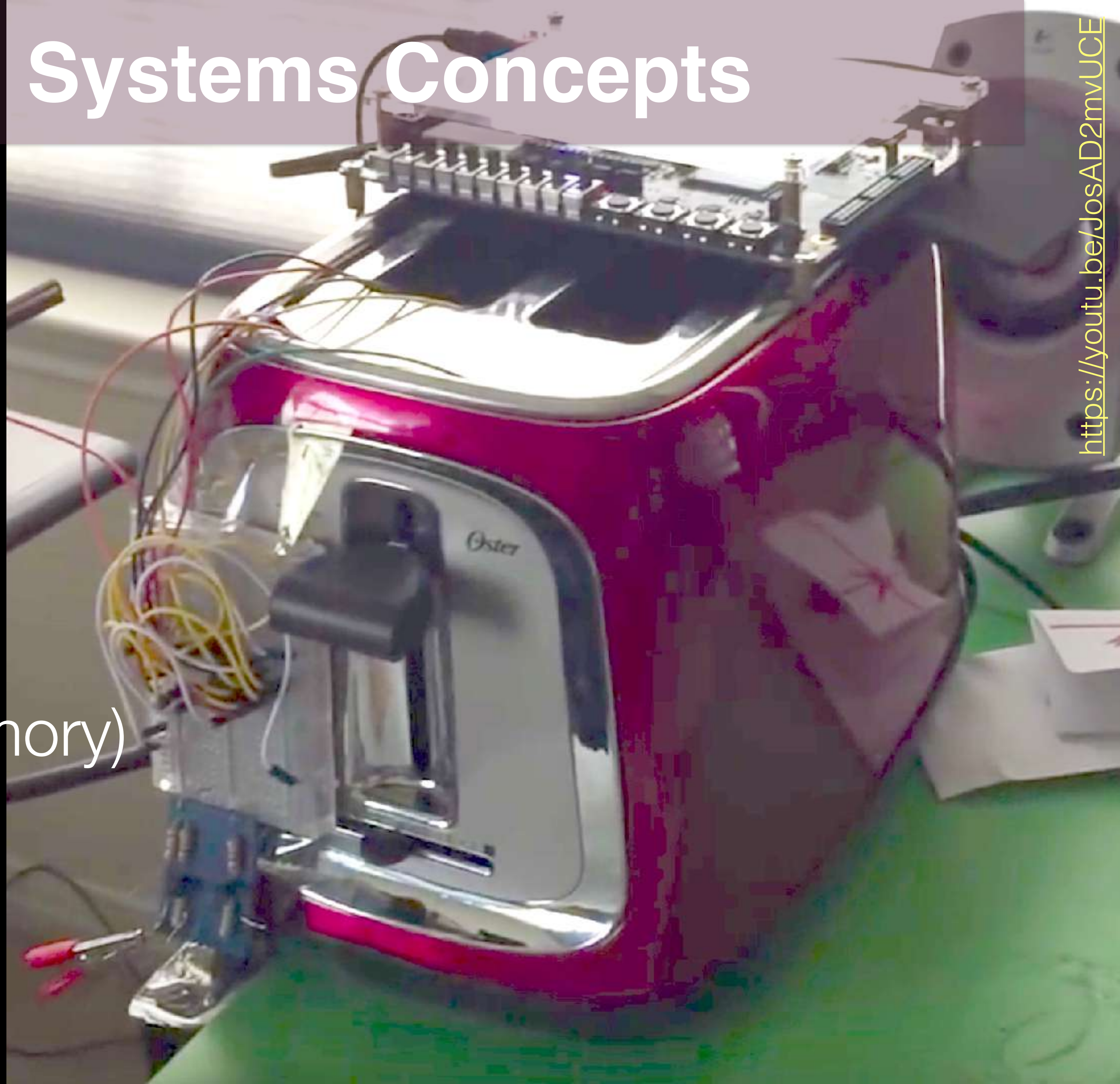
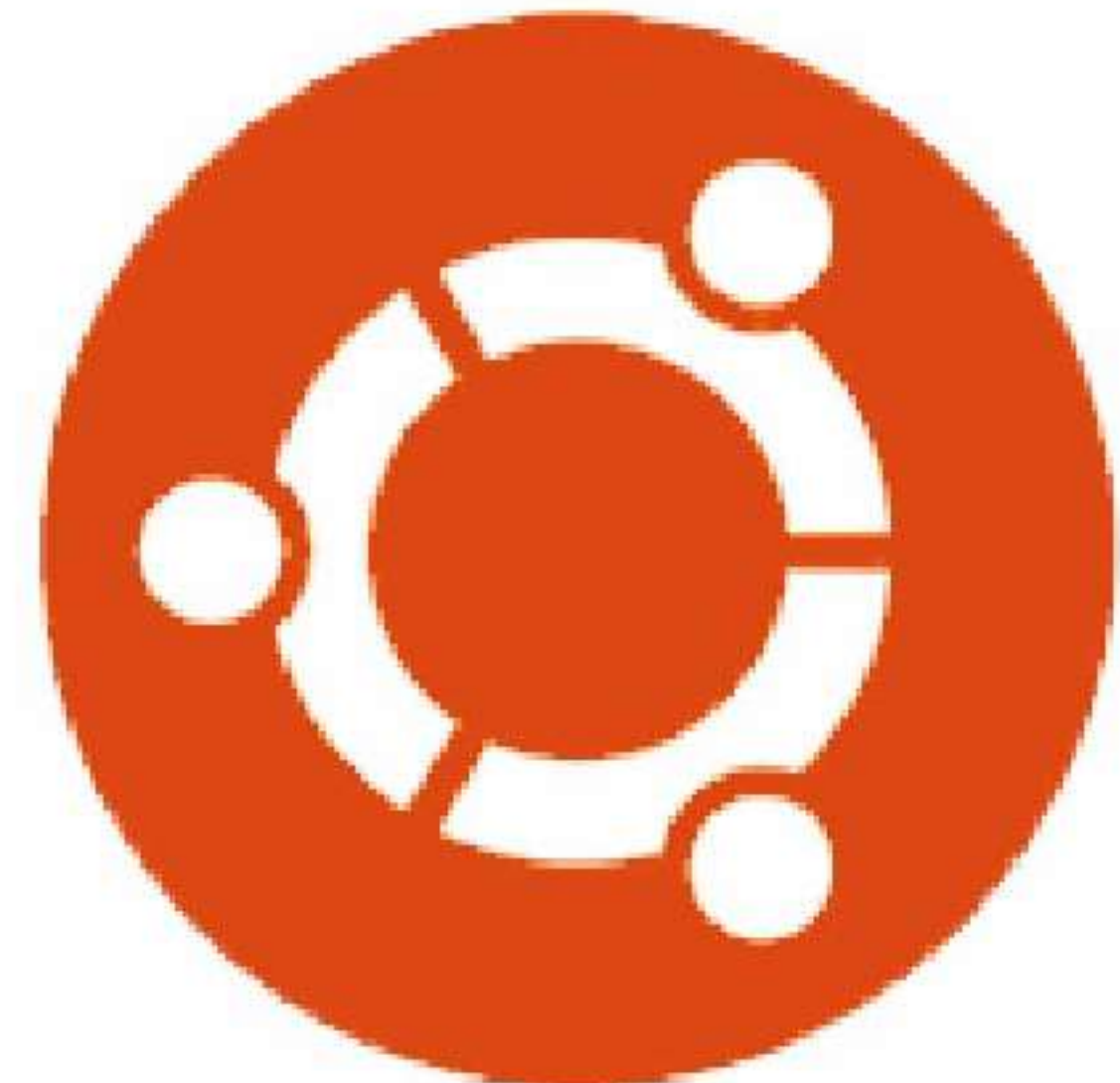
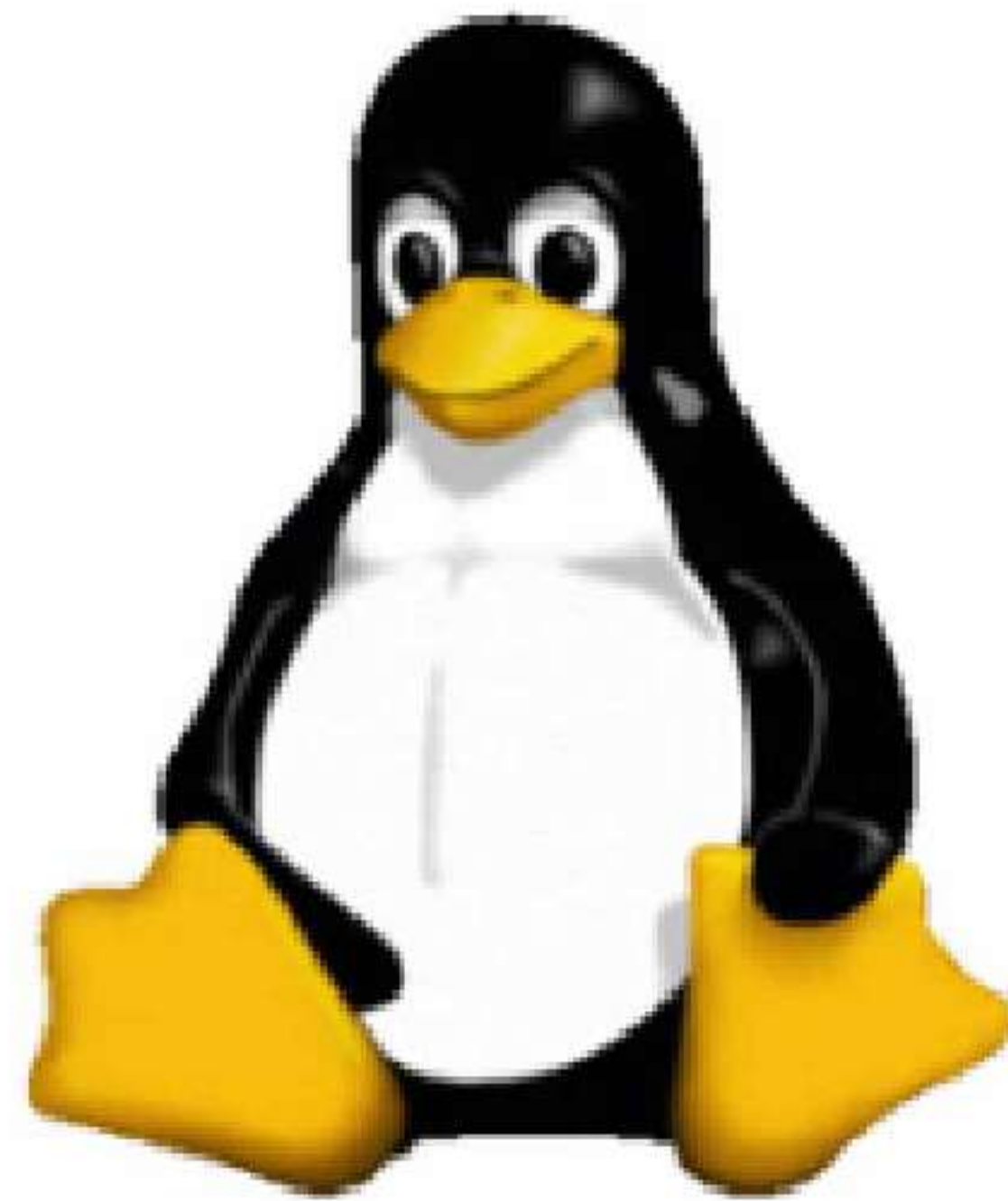


# CSE 4001 Operating Systems Concepts

## Topics

- OS Structures
- Processes and Threads
- CPU Scheduling
- Process Synchronization
- Memory (and Virtual Memory)
- File-Systems
- I/O Systems





# OneCore to rule them all: How Windows Everywhere finally happened

Microsoft promised developers that Windows would run anywhere. This summer, it finally will.

PETER BRIGHT - 5/20/2016, 7:00 AM



Microsoft

Everywhere Windows 10 can be. And on the server, too, though there it gets a different branding.

*One OS to rule them all, One OS to find them,  
One OS to bring them all and in the darkness bind them*

restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select safe Mode.

Technical information:

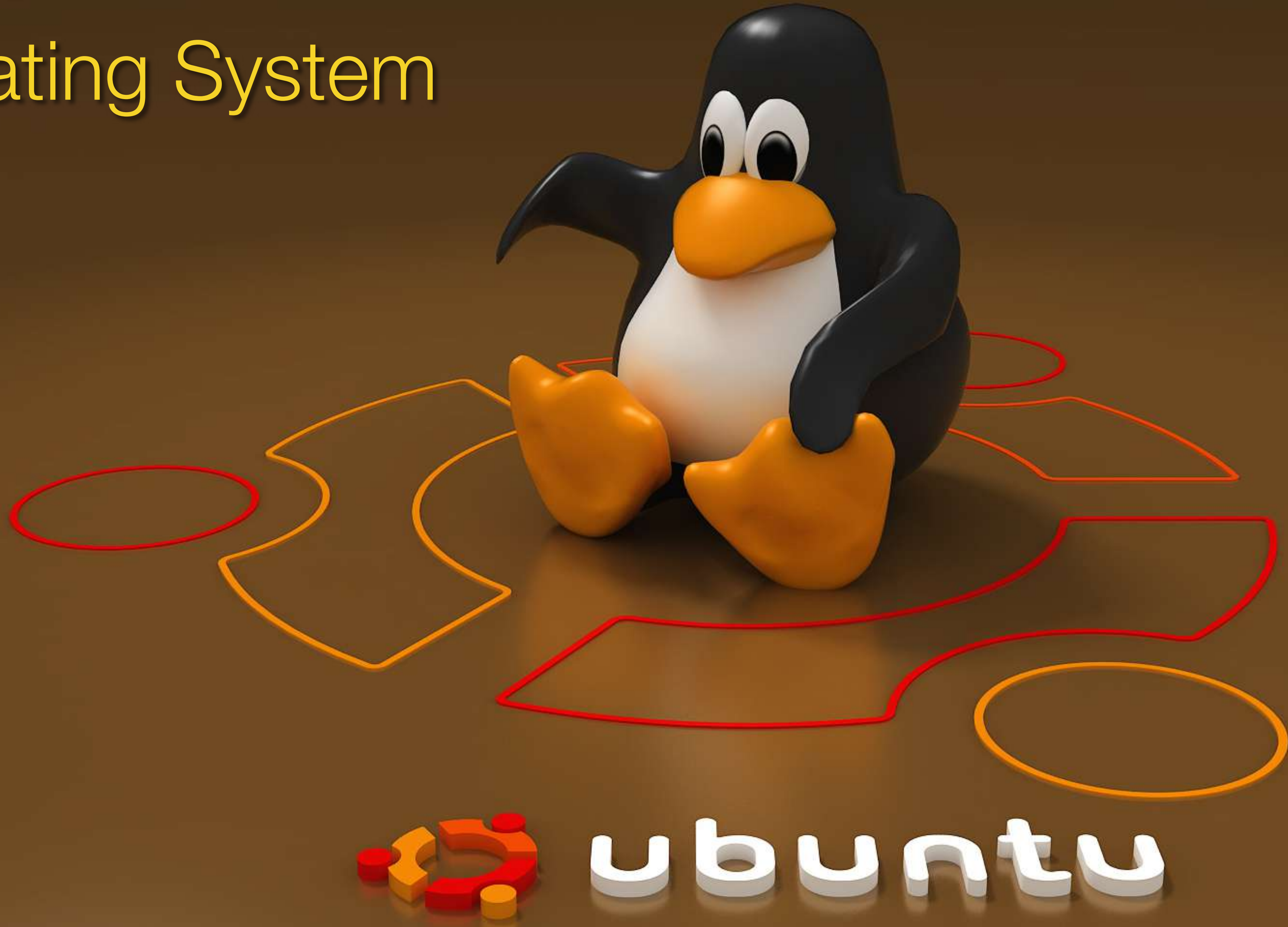
\*\*\* STOP: 0x000000E2 (0x00000000,0x00000000,0x00000000,0x00000000)

Beginning dump of physical memory  
Physical memory dump complete.

Contact your system administrator or technical support group for further assistance.



# Operating System



# Programming language



C



C++



Java/C#

# Assignments and Exams

## Type

## % of Total Grade

Midterm exams

30%

Final exam

40%

Assignments

25%

Quizzes

5%

# Office Hours

Tuesdays and Thursdays:  
from 2.00pm to 3.00pm  
Harris Center for Science  
and Engineering

208



**DR. E. RIBEIRO**

**COMPUTER  
SCIENCES**





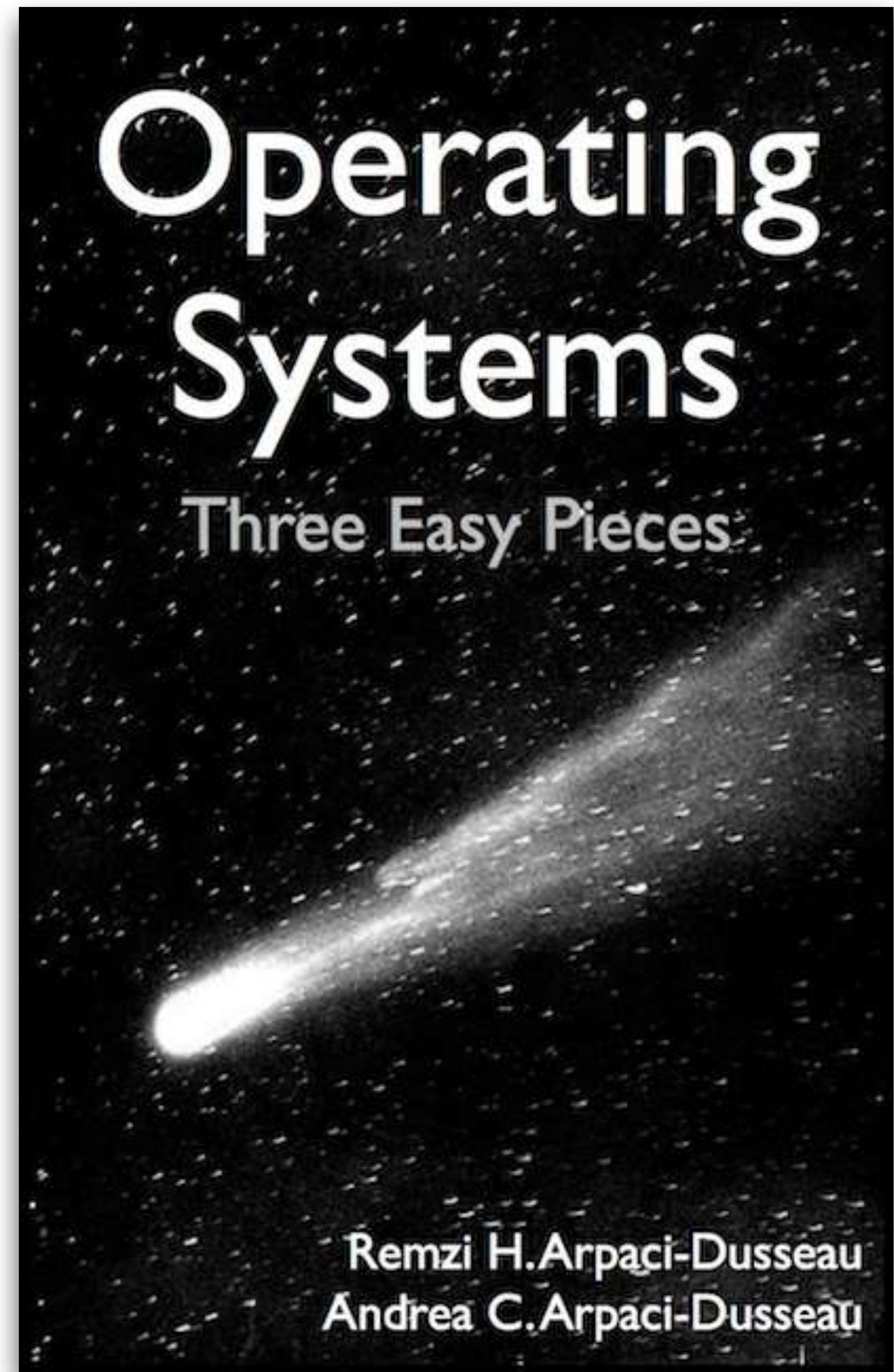
Book (required)

**Operating Systems: Three Easy Pieces**

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

Book and materials available from:

<http://pages.cs.wisc.edu/~remzi/OSTEP/>



# Operating Systems: Three Easy Pieces

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

Intro	Virtualization		Concurrency	Persistence	Appendices
<a href="#">Preface</a>	<a href="#">3 <i>Dialogue</i></a>	<a href="#">12 <i>Dialogue</i></a>	<a href="#">25 <i>Dialogue</i></a>	<a href="#">35 <i>Dialogue</i></a>	<a href="#">Dialogue</a>
<a href="#">TOC</a>	<a href="#">4 <a href="#">Processes</a></a>	<a href="#">13 <a href="#">Address Spaces</a></a>	<a href="#">26 <a href="#">Concurrency and Threads</a> <sup>code</sup></a>	<a href="#">36 <a href="#">I/O Devices</a></a>	<a href="#">Virtual Machines</a>
<a href="#">1 <i>Dialogue</i></a>	<a href="#">5 <a href="#">Process API</a> <sup>code</sup></a>	<a href="#">14 <a href="#">Memory API</a></a>	<a href="#">27 <a href="#">Thread API</a></a>	<a href="#">37 <a href="#">Hard Disk Drives</a></a>	<a href="#">Dialogue</a>
<a href="#">2 <a href="#">Introduction</a> <sup>code</sup></a>	<a href="#">6 <a href="#">Direct Execution</a></a>	<a href="#">15 <a href="#">Address Translation</a></a>	<a href="#">28 <a href="#">Locks</a></a>	<a href="#">38 <a href="#">Redundant Disk Arrays (RAID)</a></a>	<a href="#">Monitors</a>
	<a href="#">7 <a href="#">CPU Scheduling</a></a>	<a href="#">16 <a href="#">Segmentation</a></a>	<a href="#">29 <a href="#">Locked Data Structures</a></a>	<a href="#">39 <a href="#">Files and Directories</a></a>	<a href="#">Dialogue</a>
	<a href="#">8 <a href="#">Multi-level Feedback</a></a>	<a href="#">17 <a href="#">Free Space Management</a></a>	<a href="#">30 <a href="#">Condition Variables</a></a>	<a href="#">40 <a href="#">File System Implementation</a></a>	<a href="#">Lab Tutorial</a>
	<a href="#">9 <a href="#">Lottery Scheduling</a> <sup>code</sup></a>	<a href="#">18 <a href="#">Introduction to Paging</a></a>	<a href="#">31 <a href="#">Semaphores</a></a>	<a href="#">41 <a href="#">Fast File System (FFS)</a></a>	<a href="#">Systems Labs</a>
	<a href="#">10 <a href="#">Multi-CPU Scheduling</a></a>	<a href="#">19 <a href="#">Translation Lookaside Buffers</a></a>	<a href="#">32 <a href="#">Concurrency Bugs</a></a>	<a href="#">42 <a href="#">FSCK and Journaling</a></a>	<a href="#">xv6 Labs</a>
	<a href="#">11 <a href="#">Summary</a></a>	<a href="#">20 <a href="#">Advanced Page Tables</a></a>	<a href="#">33 <a href="#">Event-based Concurrency</a></a>	<a href="#">43 <a href="#">Log-structured File System (LFS)</a></a>	
		<a href="#">21 <a href="#">Swapping: Mechanisms</a></a>	<a href="#">34 <a href="#">Summary</a></a>	<a href="#">44 <a href="#">Data Integrity and Protection</a></a>	
		<a href="#">22 <a href="#">Swapping: Policies</a></a>		<a href="#">45 <a href="#">Summary</a></a>	
		<a href="#">23 <a href="#">Case Study: VAX/VMS</a></a>		<a href="#">46 <a href="#">Dialogue</a></a>	
		<a href="#">24 <a href="#">Summary</a></a>		<a href="#">47 <a href="#">Distributed Systems</a></a>	
				<a href="#">48 <a href="#">Network File System (NFS)</a></a>	
				<a href="#">49 <a href="#">Andrew File System (AFS)</a></a>	
				<a href="#">50 <a href="#">Summary</a></a>	

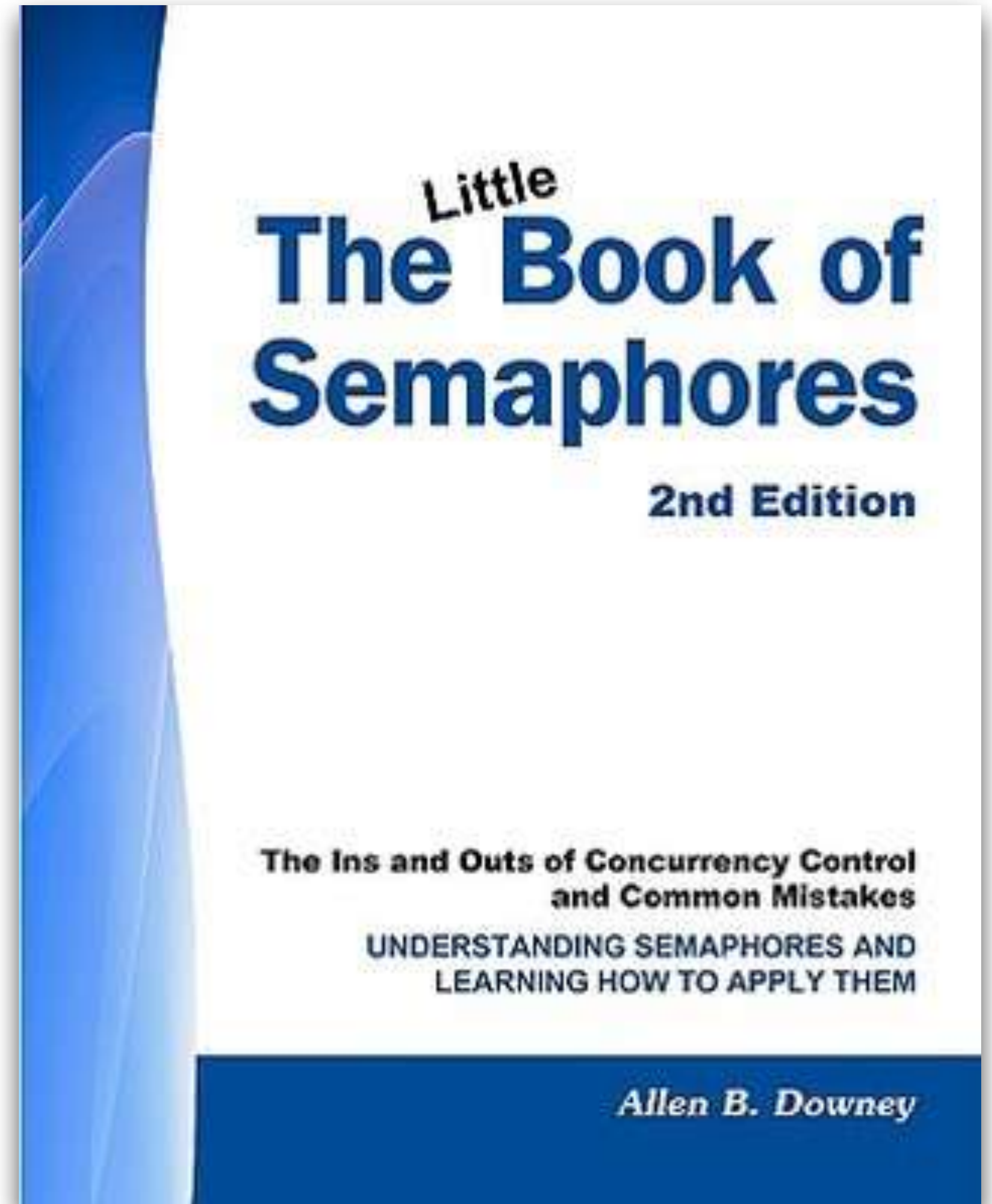
# Another book

## **The little book of semaphores**

Allen Downey

Book and materials available from:

[http://www.greenteapress.com/  
semaphores/](http://www.greenteapress.com/semaphores/)



# Contents

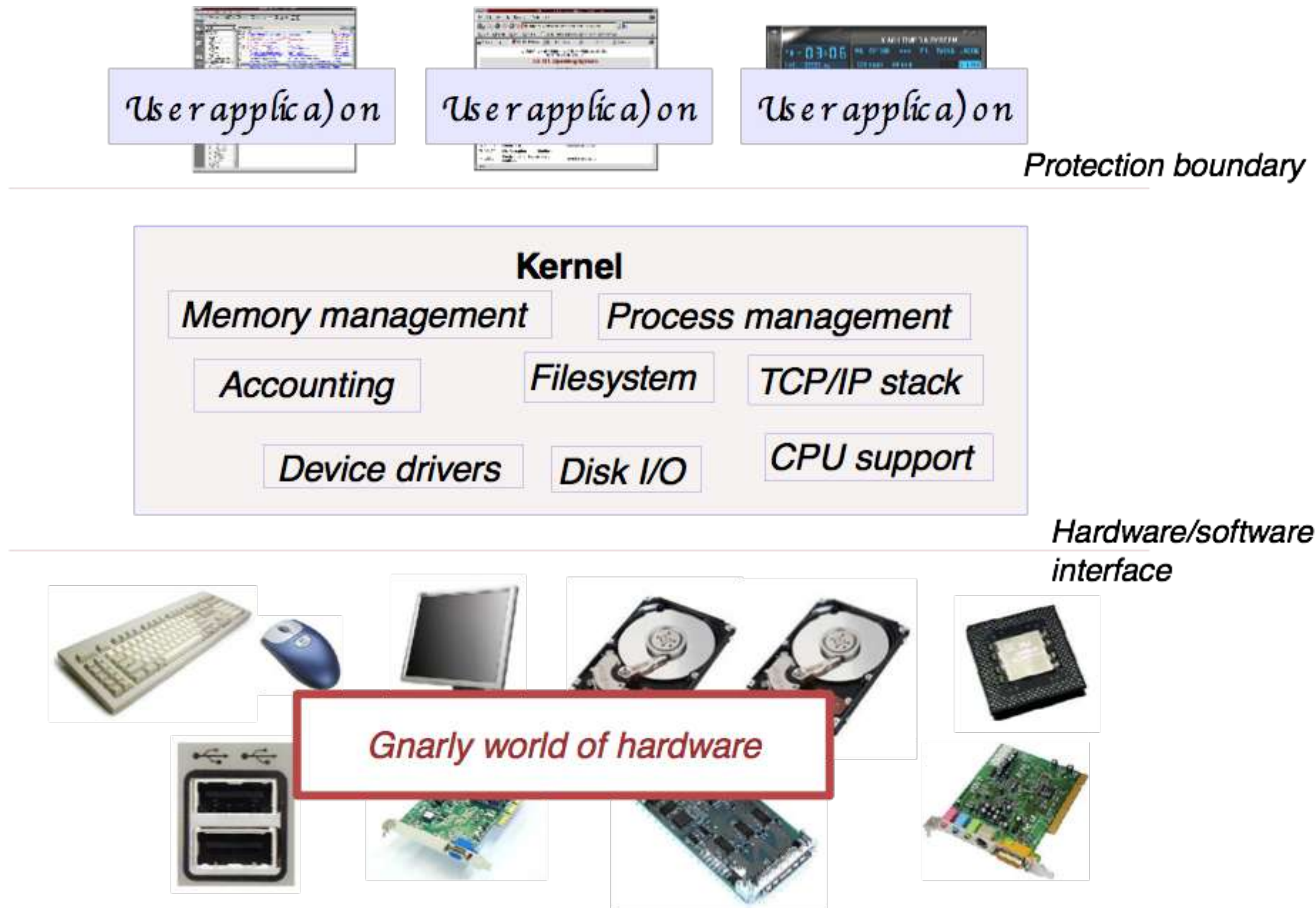
- What is an operating system?
- OS functions

# What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and simplify solving user problems.
  - Make the computer system convenient to use
  - Use the computer hardware efficiently

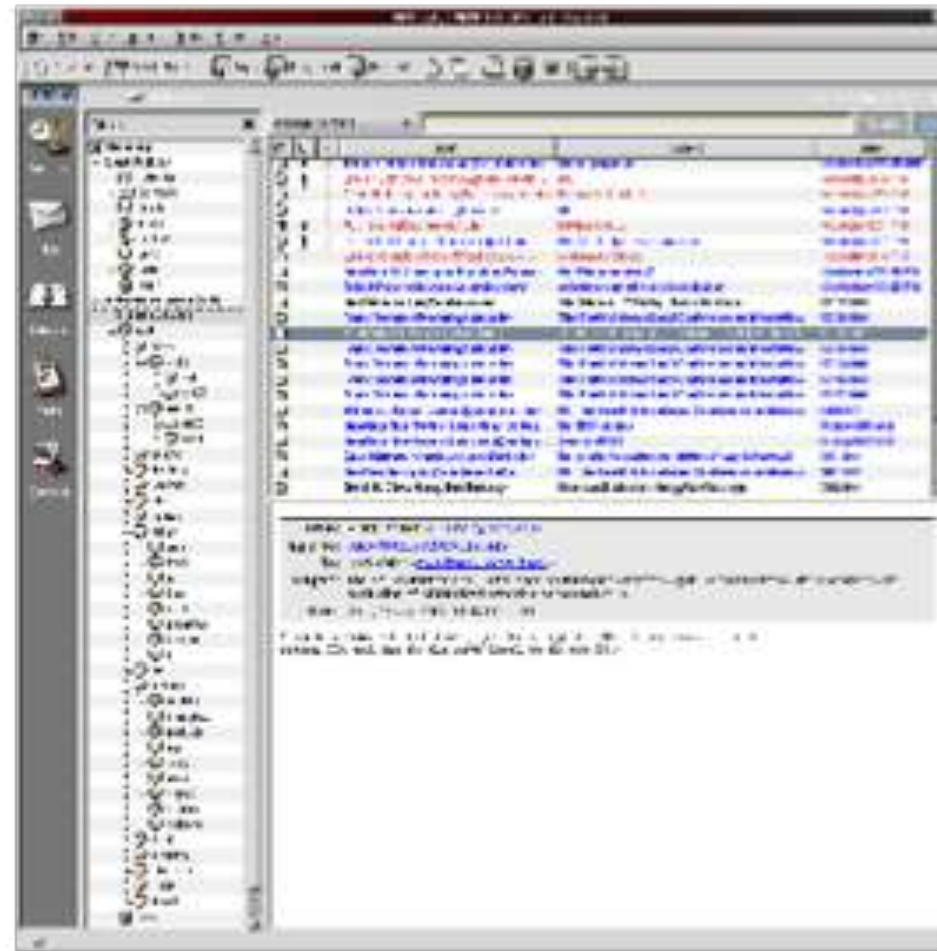
# What is an operating system?

Software that provides an elaborate illusion to applications



# One OS Function: Concurrency

Give every application the illusion of having its own CPU!



*I think I have my own CPU*

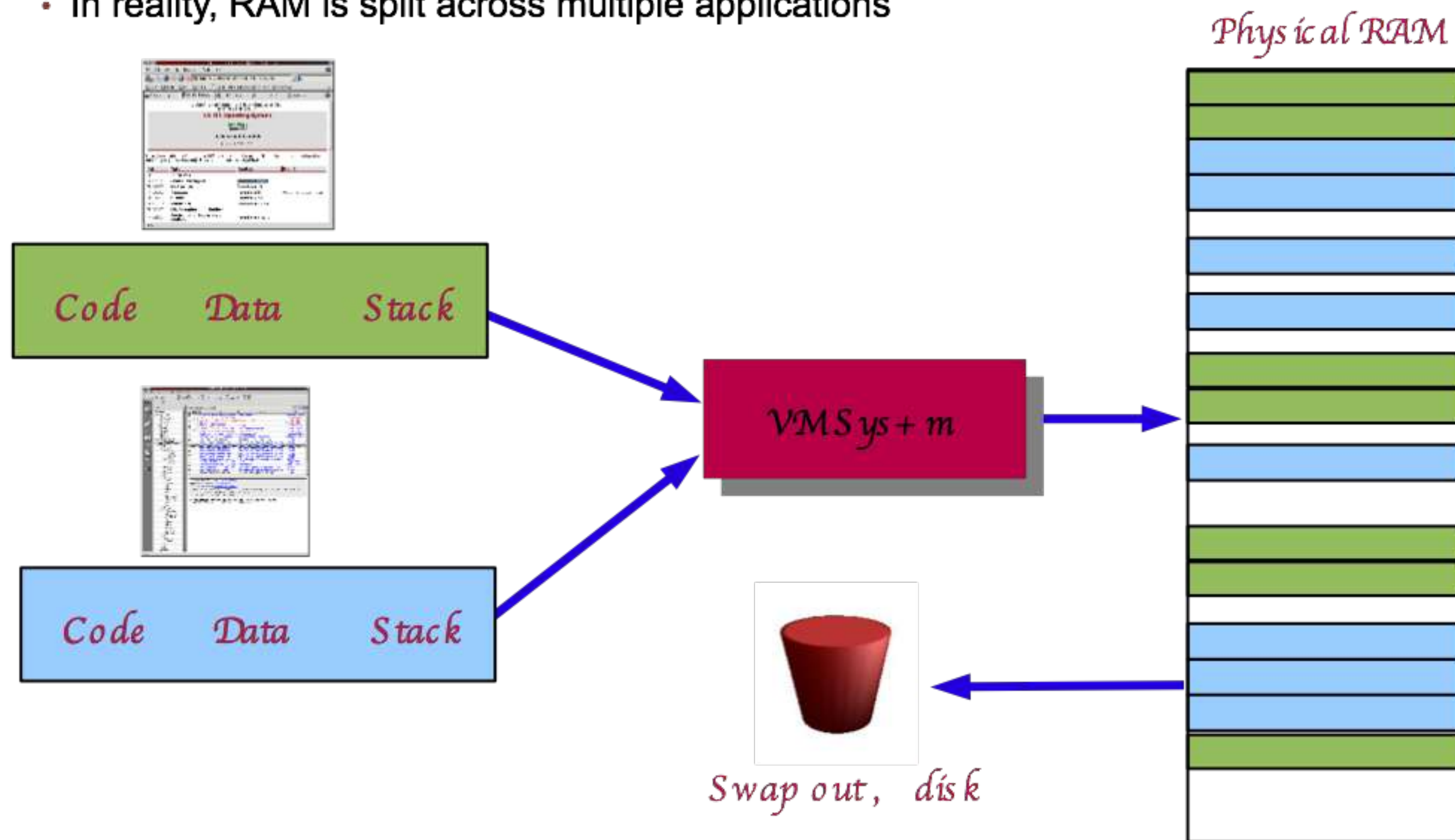
*So do I*



# Another OS Function: Virtual Memory

Give every application the illusion of having infinite memory

- And, that it can access any memory address it likes!
- In reality, RAM is split across multiple applications





# More OS Functions

## Multiprocessor support

- Modern systems have multiple CPUs
- Can run multiple applications (or **threads** within applications) in parallel
- OS must ensure that memory and cache contents are consistent across CPUs

## Filesystems

- Real disks have a hairy, sector-based access model
- User applications see flat files arranged in a hierarchical namespace

## Network protocols

- Network interface hardware operates on the level of unreliable packets
- User apps see a (potentially reliable) byte-stream *socket*

## Security and protection

- Prevent multiple apps from interfering with each other and with normal system operation