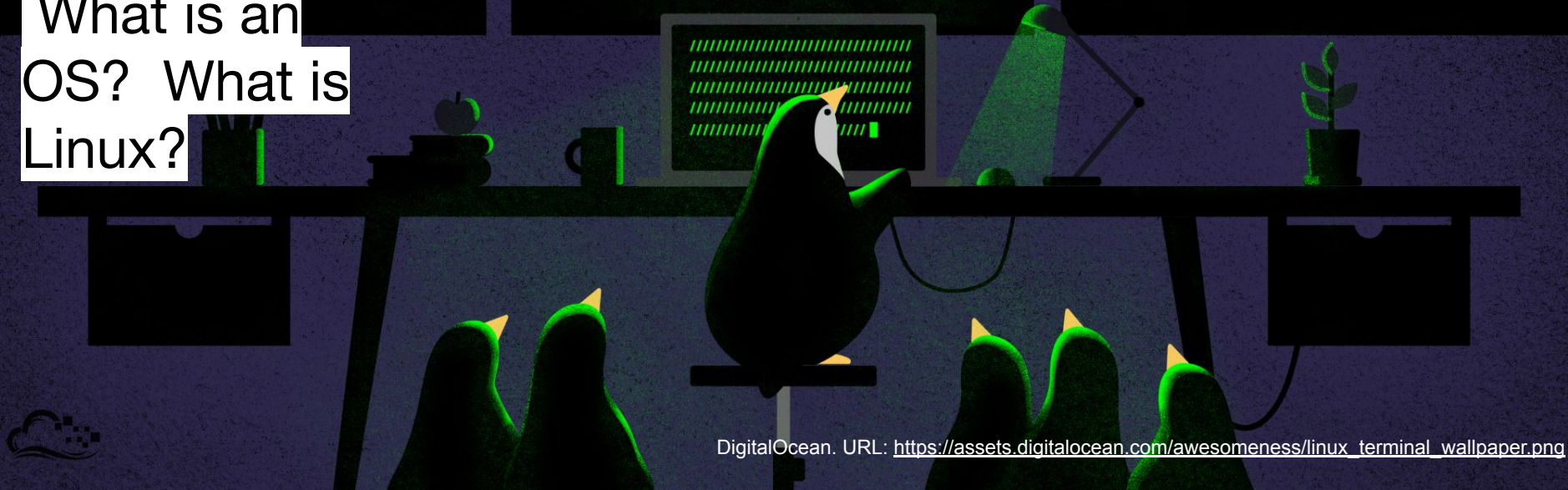


ELEC 424/553

Mobile & Embedded Systems

Lecture 3

What is an
OS? What is
Linux?



Discussion:

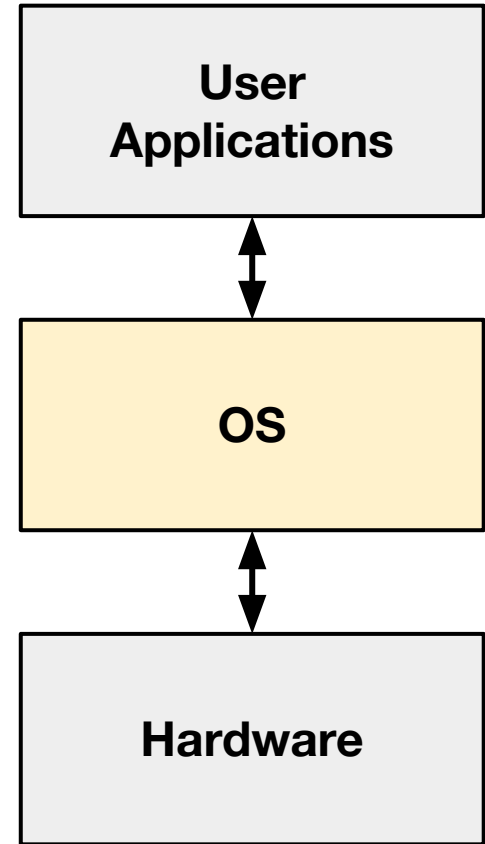
**How would you define an
operating system?**

Discussion:

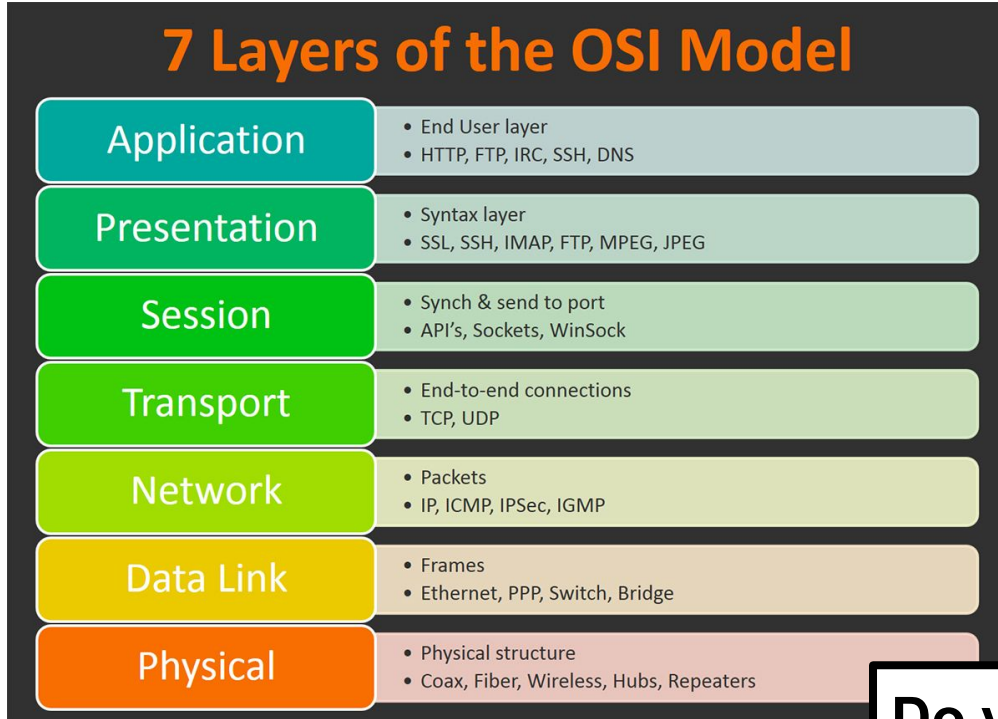
**Which operating system do you
prefer & why?**

What Is An OS (Operating System)?

- **Software interface between user applications and hardware**
- “The software that runs your software”
- Simplifies application development
 - You program your app to work with the operating system
 - You don't have to make different versions for every computer configuration
- Broadens application market



The Importance of Layers of Abstraction

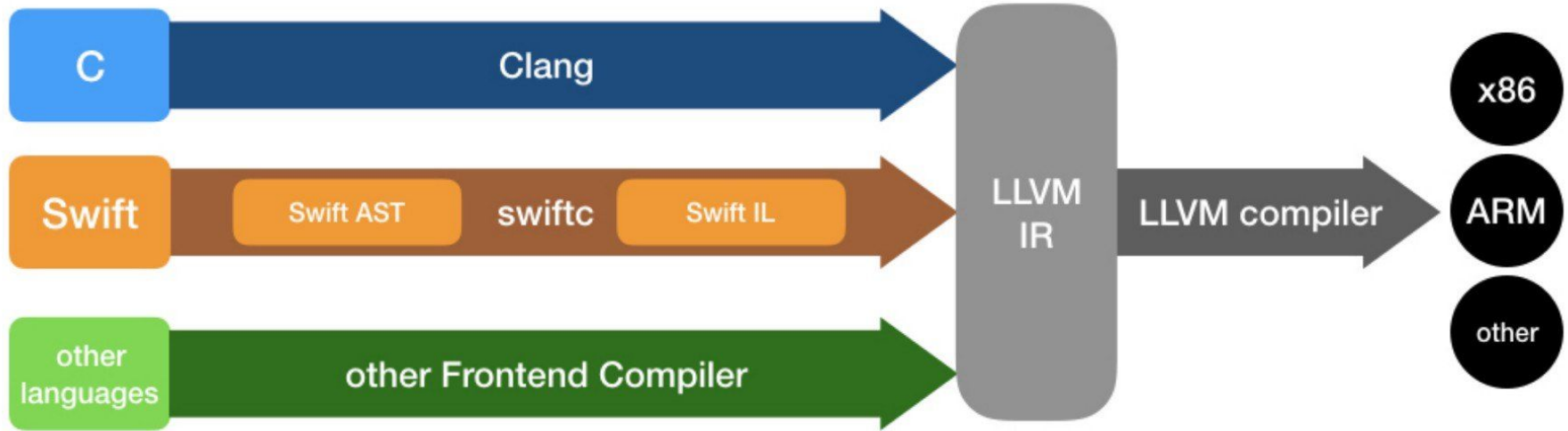


<https://www.bmc.com/blogs/osi-model-7-layers/>

- Critical to get layers right
- Reduces breakage
- Enables layers to be optimized independently
- Enables layer developments to propagate across diverse systems

Do you recognize this?

The Importance of Layers of Abstraction!



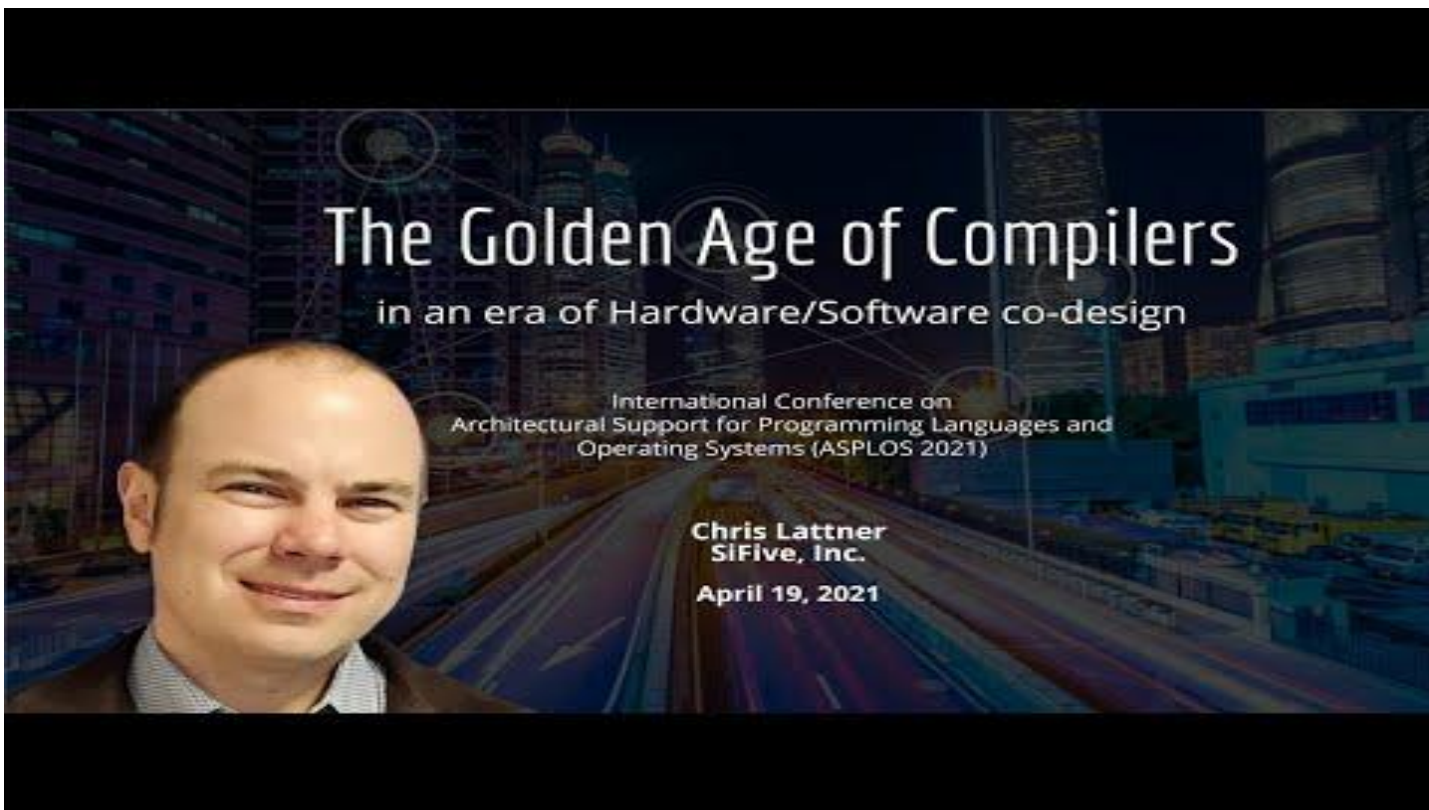
<https://www.omnisci.com/technical-glossary/llvm>

Chris Lattner
Main author of LLVM



<https://insights.dice.com/2017/01/11/chris-lattner-leaves-apple-tesla/>

Chris Lattner Is a Fan



Operating System (OS) Examples



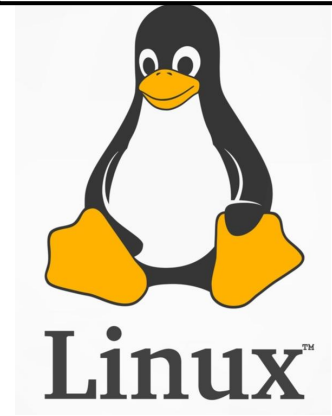
<https://www.crn.com/news/applications-os/microsoft-windows-10-update-to-windows-11-not-until-next-year>

r



Image: Apple; URL:
<https://www.theverge.com/2020/6/22/21295489/apple-macos-big-sur-update-redesign-apps-features-catalyst-wwdc-2020>

Sort Of



<https://wallpaperaccess.com/linux-penguin>

What Do We Mean When We Say Linux?

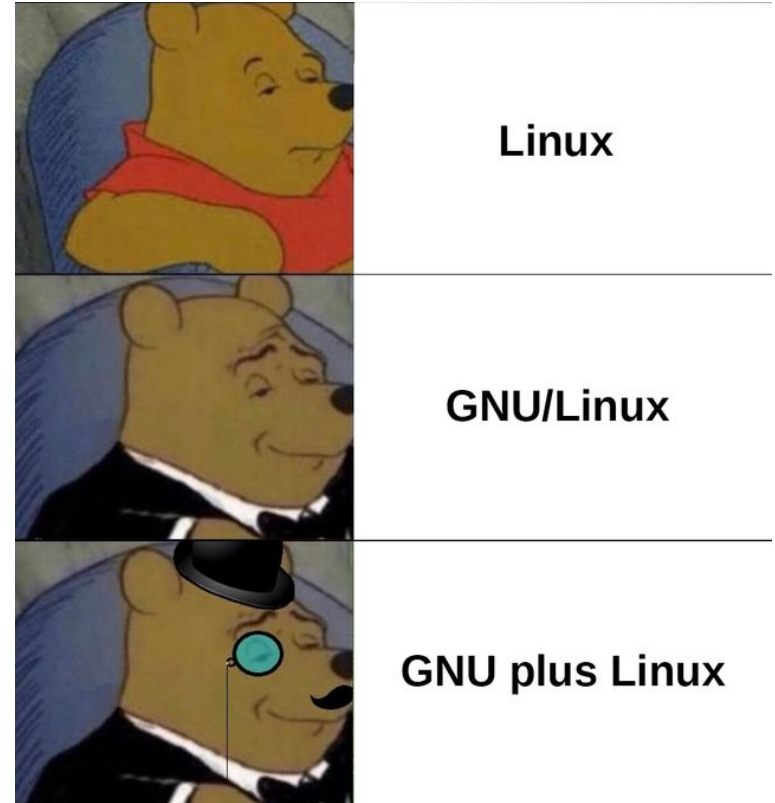
GNU Core Utilities (Coreutils) give us a lot of the features we love - e.g., UNIX commands ([cat](#), ls, and so on)

<https://www.gnu.org/software/coreutils/>
<https://github.com/coreutils/coreutils>

Bash

GNU C Library (glibc)

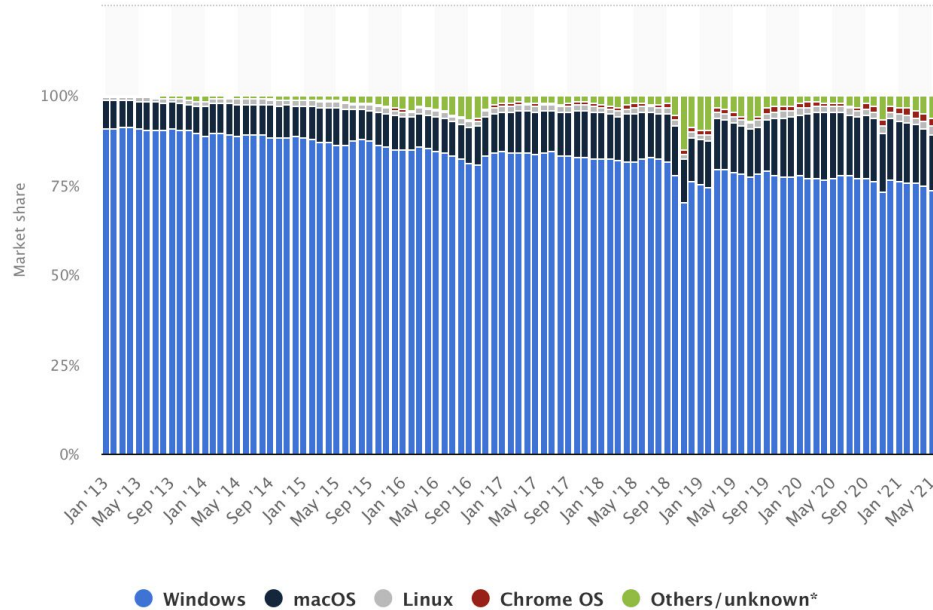
Linux Kernel - Linux technically is just the kernel



<https://nixcraft.tumblr.com/post/183844164917/rms-approve-d-linux-meme-as-i-have-recently-taken>

Desktop OS Market Share

Desktop PC operating system market share worldwide, from January 2013 to June 2021



“Year of
the Linux
desktop”...

More OS Examples



Android

<https://source.android.com/setup/start/brands>

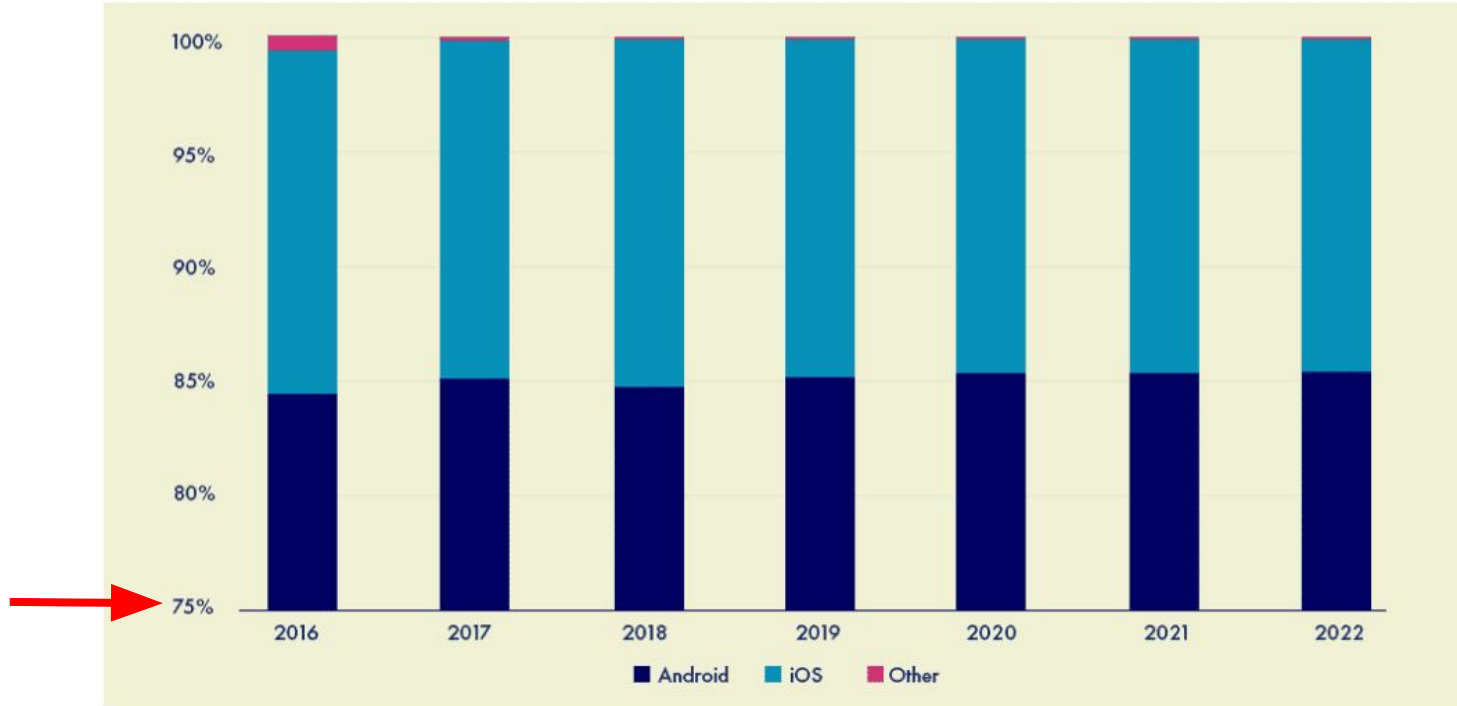


iOS

<https://www.gizchina.com/2021/03/09/ios-15-everything-you-need-to-know-about-apples-next-iphone-os/>

Apple & Google Dominate Smartphones

Global Android & iPhone Market Share



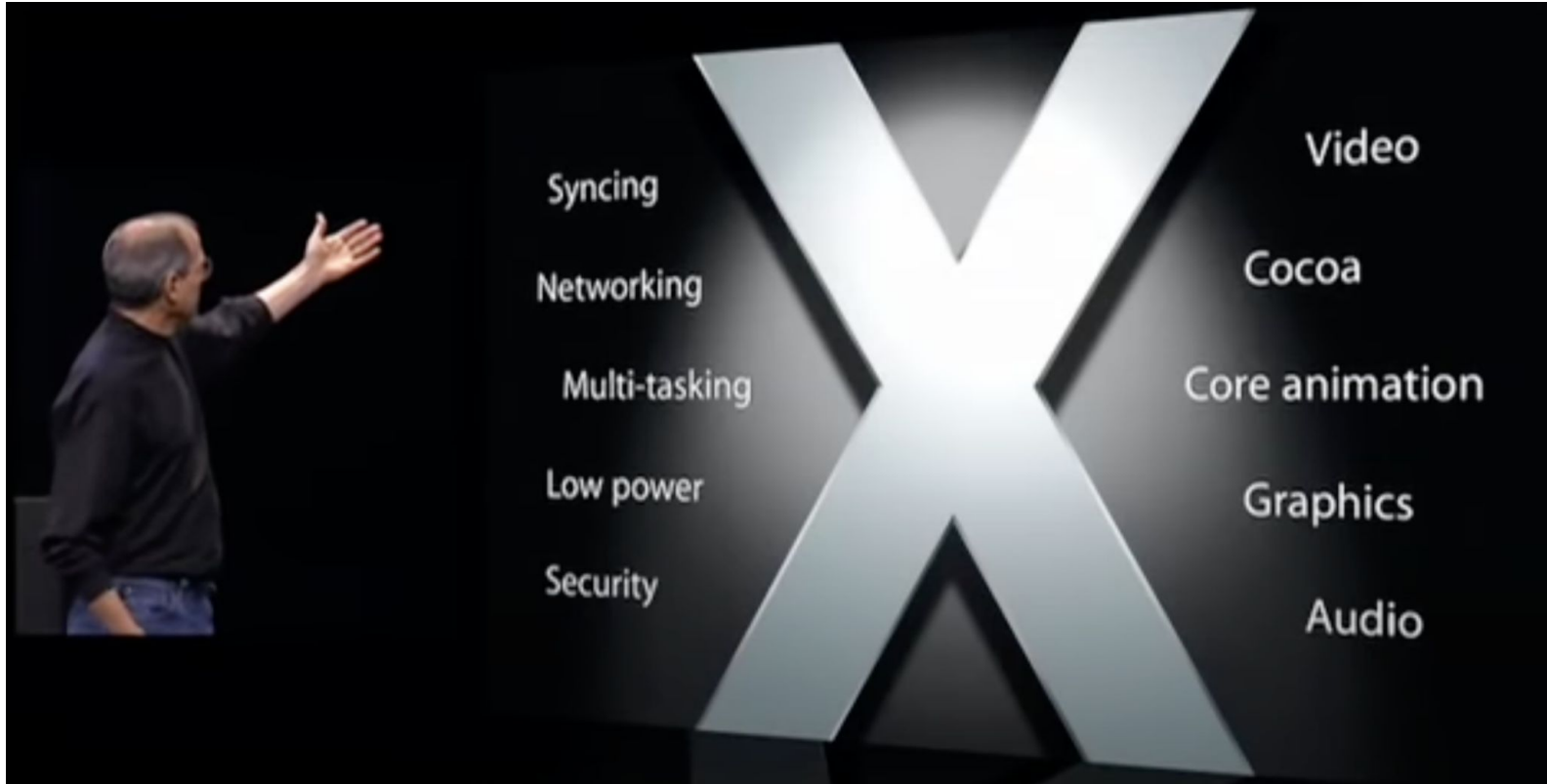
**Android is
built on top
of Linux**

So, Linux
dominates
smartphones

Discussion:

**What are the parts of an
operating system?**

Steve Jobs OS X Overview

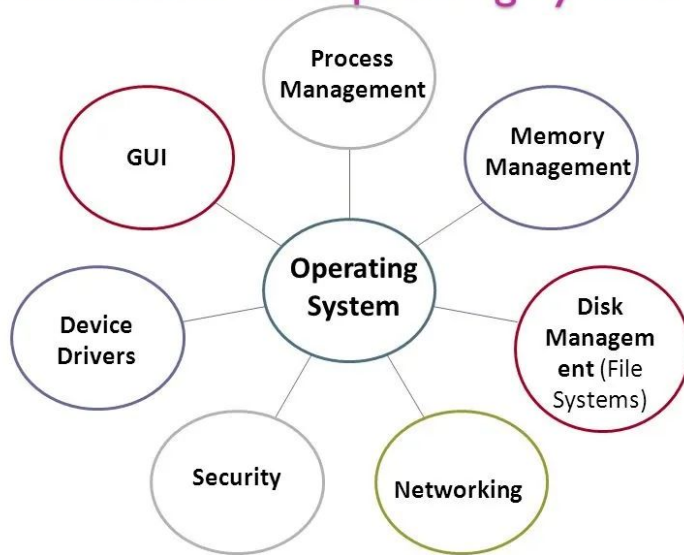


In General, What are the Parts of an OS?



More Technical Definition of OS Components

Functions of an Operating System



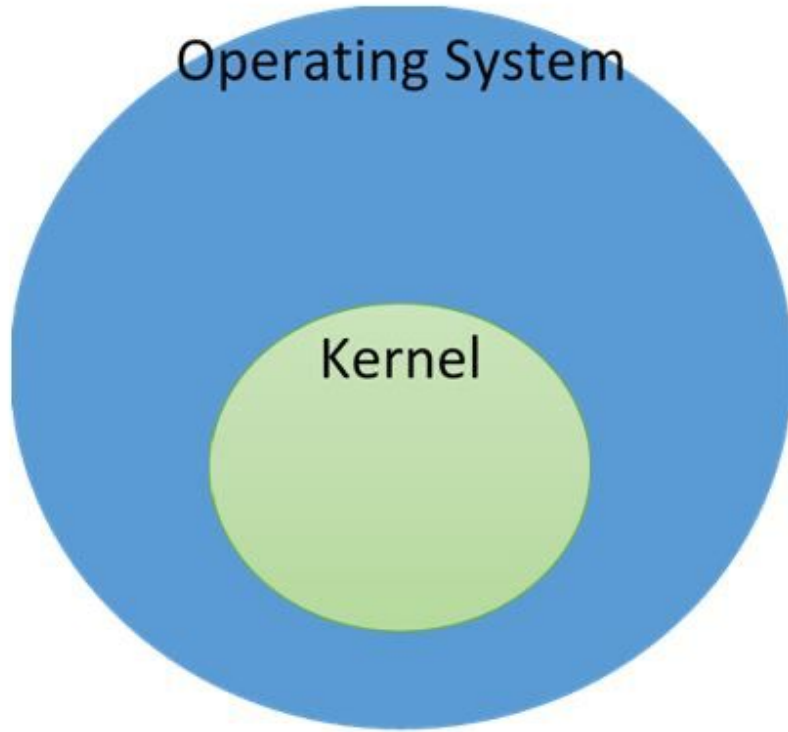
Modern Operating System Functionality

- **Process and Thread Management**
- **Concurrency:** Doing many things simultaneously (I/O, processing, multiple programs, etc.)
 - Several users work at the same time as if each has a private machine
 - Threads (unit of OS control) - one thread on the CPU at a time, but many threads active concurrently
- **I/O devices:** let the CPU work while a slow I/O device is working
- **Memory management:** OS coordinates allocation of memory and moving data between disk and main memory.
- **Files:** OS coordinates how disk space is used for files, in order to find files and to store multiple files
- **Distributed systems & networks:** allow a group of machines to work together on distributed hardware



^^ Slide from: Sean Barker, University of Massachusetts Amherst

Where Is the Kernel In This?



- **Kernel:** Application & hardware interface
 - Manages:
 - Memory
 - Processes
 - System calls
 - Device drivers
- **OS:** User & hardware interface
 - Includes the above plus:
 - GUI
 - Tools

GNU/Linux

- “GNU/Linux operating system”
- Linux kernel
- GNU utilities
 - Text editor (nano; Emacs)
 - Shell (command line)

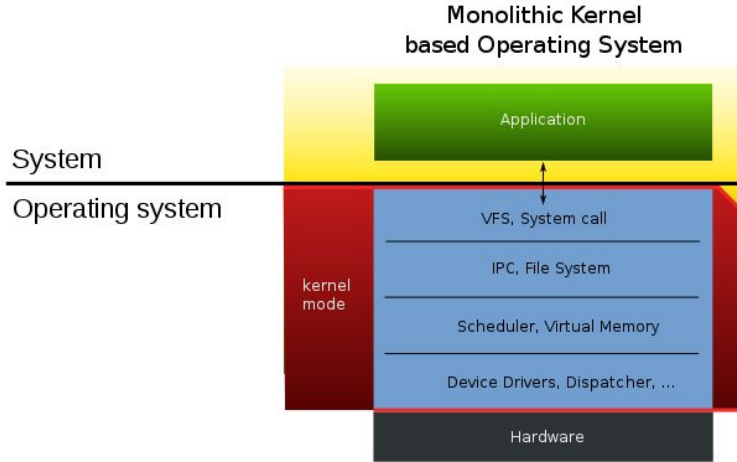


<https://nixcraft.tumblr.com/post/183844164917/rms-approve-d-linux-meme-as-i-have-recently-taken>

Motivation for Making Distinction of a Kernel

- Consider a bad user application
- For example, a poorly-written web browser
- What if this application could write to any part of memory?
 - Goodbye other applications
 - Goodbye kernel
- **Memory protection** is important!

What Is The Kernel? It Depends - Consider a Monolithic Kernel

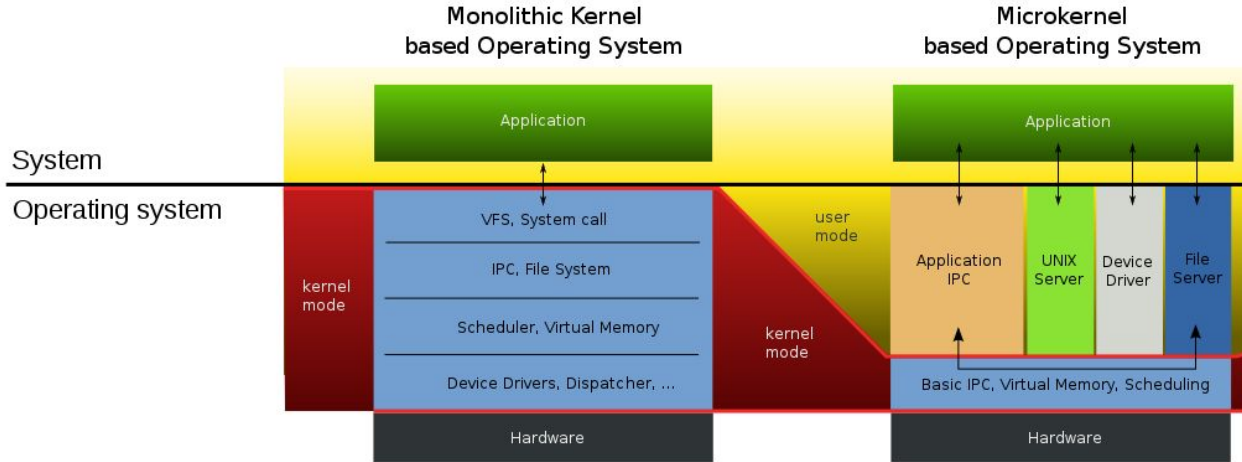


- VFS - Virtual file system
- System call
- IPC - Inter-process communication
- File system
- Scheduler
- Virtual memory
- Device Drivers

Any kernel before 1980s was monolithic

Robert Love, *Linux Kernel Development*, 3rd Edition

What is the Kernel? It depends - Consider a Microkernel



<https://commons.wikimedia.org/wiki/File:OS-structure2.svg>

- Not one process
- Roles distributed among **servers** (processes)
- When necessary, servers exist in kernel
- Unique address spaces for servers
- Requires **message passing**
- Robust to faults


Monolithic Kernel vs. Microkernel

- Monolithic:
 - Simpler
 - Single process, single address space
 - Higher performance
- Microkernel:
 - Modularization
 - Message passing
 - Higher reliability

Basis for Comparison	Microkernel	Monolithic Kernel
Size	Microkernel is smaller in size	It is larger than microkernel
Execution	Slow Execution	Fast Execution
Extendible	It is easily extendible	It is hard to extend
Security	If a service crashes, it does effects on working on the microkernel	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel more code is required	To write a monolithic kernel less code is required
Example	QNX, Symbian, L4Linux etc.	Linux,BSDs(FreeBSD,OpenBSD,NetBSD)etc.

<https://www.geeksforgeeks.org/monolithic-kernel-and-key-differences-from-microkernel/>

This Is Not Without Controversy



WIKIPEDIA
The Free Encyclopedia

- Main page
- Contents
- Current events
- Random article
- About Wikipedia
- Contact us
- Donate

Contribute

- Help
- Learn to edit
- Community portal
- Recent changes
- Upload file

Tools

- What links here
- Related changes
- Special pages
- Permanent link
- Page information
- Cite this page
- Wikidata item

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)


Article [Talk](#)

Read [Edit](#) [View history](#)

Search Wikipedia

Tanenbaum–Torvalds debate

From Wikipedia, the free encyclopedia



This article **needs additional citations for verification**. Please help [improve this article](#) by [adding citations to reliable sources](#). Unsourced material may be challenged and removed.



Find sources: "Tanenbaum–Torvalds debate" – news · newspapers · books · scholar · JSTOR (July 2013) [\(Learn how and when to remove this template message\)](#)

The **Tanenbaum–Torvalds debate** was a written [debate](#) between [Andrew S. Tanenbaum](#) and Linus Torvalds, regarding the Linux kernel and kernel architecture in general. Tanenbaum, the creator of Minix, began the debate in 1992 on the Usenet discussion group [comp.os.minix](#), arguing that [microkernels](#) are superior to [monolithic kernels](#) and therefore Linux was, even in 1992, [obsolete](#).^[1]

The debate has sometimes been considered a [flame war](#).^[2]

Contents [hide]

- The debate



Andrew S. Tanenbaum Linus Torvalds

Let's Hear From Our Best Friend: Linus Torvalds



From YouTube channel Computer History Museum. URL: <https://youtu.be/WVTWCPoUt8w?si=hMJvw7yDFwNp10pE&t=1619>

Let's Hear From Our Best Friend's "Best Friend"

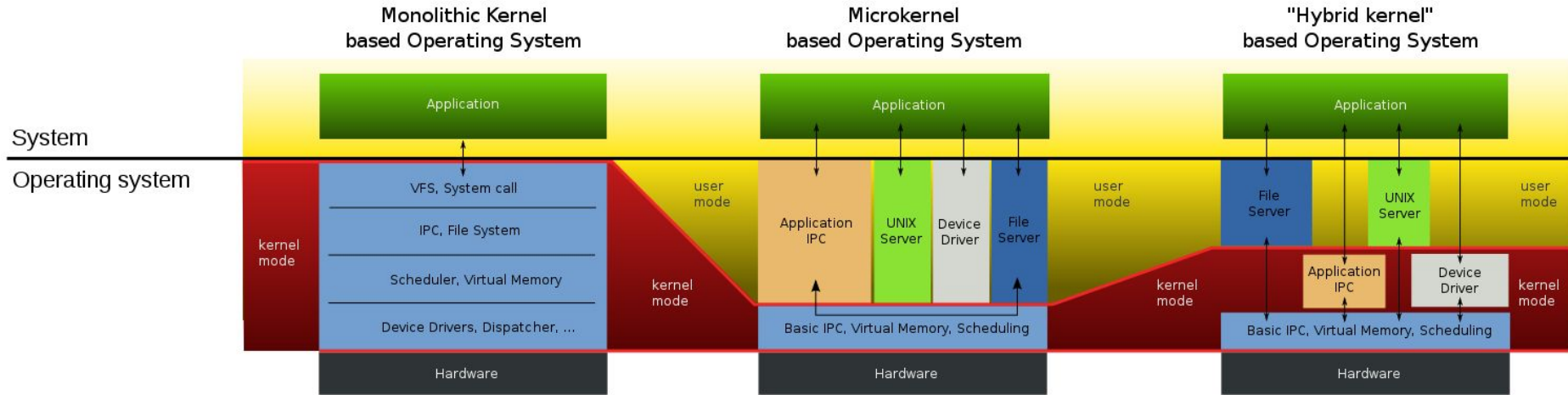


Is the Microkernel Really Implemented in Practice & At Scale?

“Neither Windows NT nor Mac OS X run any microkernel servers in user-space in their latest iteration, defeating the primary purpose of microkernel design altogether.”

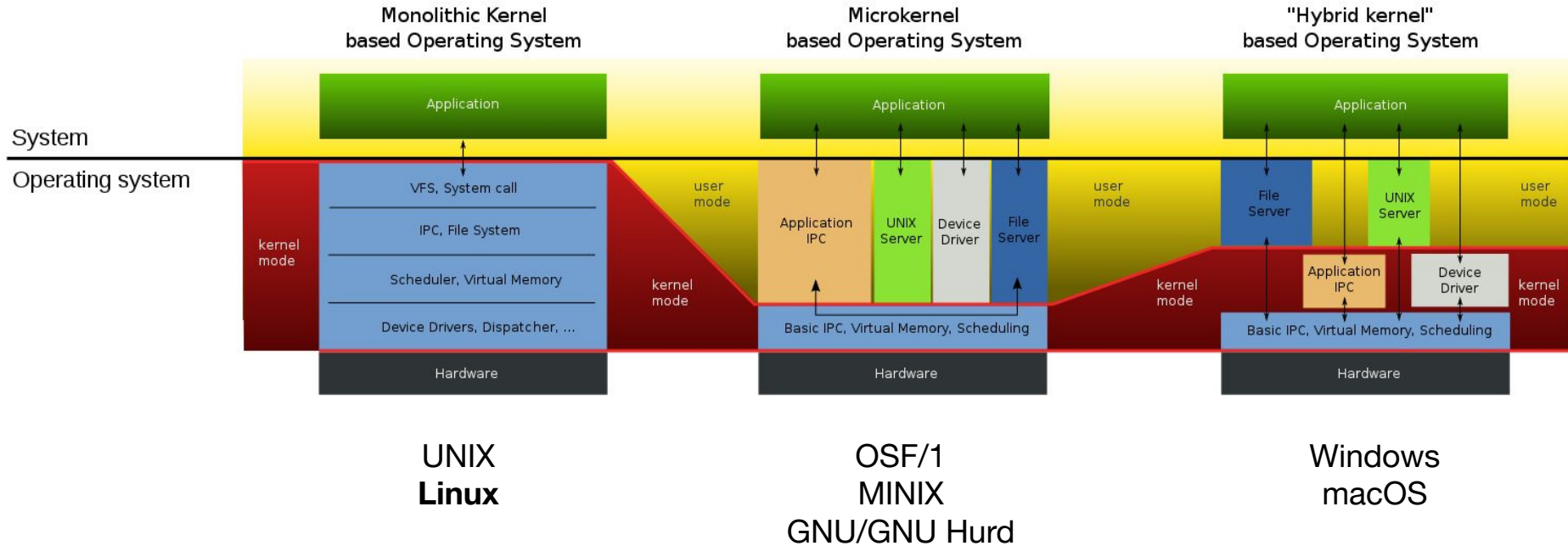
Robert Love, *Linux Kernel Development*, 3rd Edition

There's a Third Way: The Hybrid Kernel

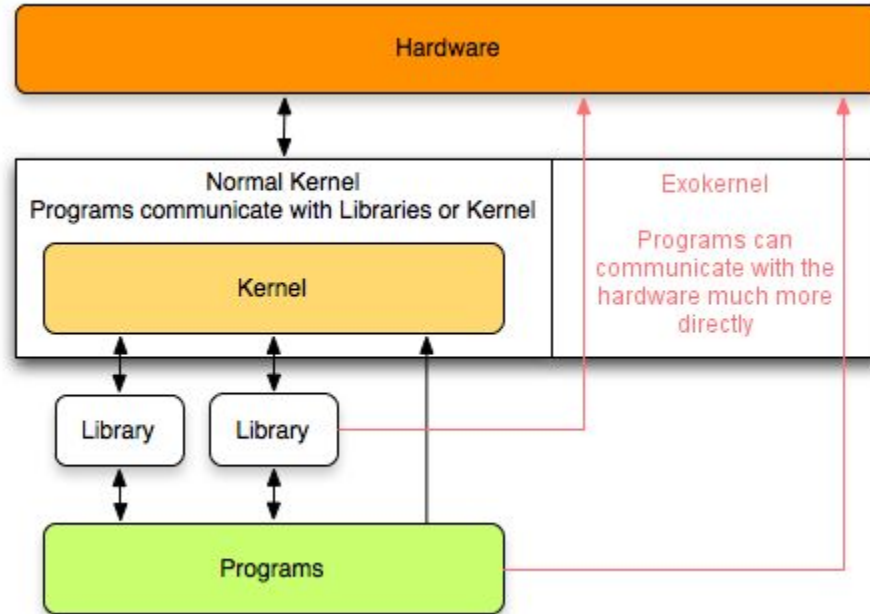


‘As to the whole "hybrid kernel" thing - it's just marketing. It's "oh, those microkernels had good PR, how can we try to get good PR for our working kernel? Oh, I know, let's use a cool name and try to imply that it has all the PR advantages that that other system has" - Linus Torvalds

Who Uses What Kernel Type?



But Wait, There's More! The Exokernel



Linux Overview

- Unix-like
- Monolithic kernel
 - Technically Linux is only the kernel
- Many distributions
 - Ubuntu
 - Red Hat
 - Debian
 - Fedora
 - Arch
- Open source - [link](#)

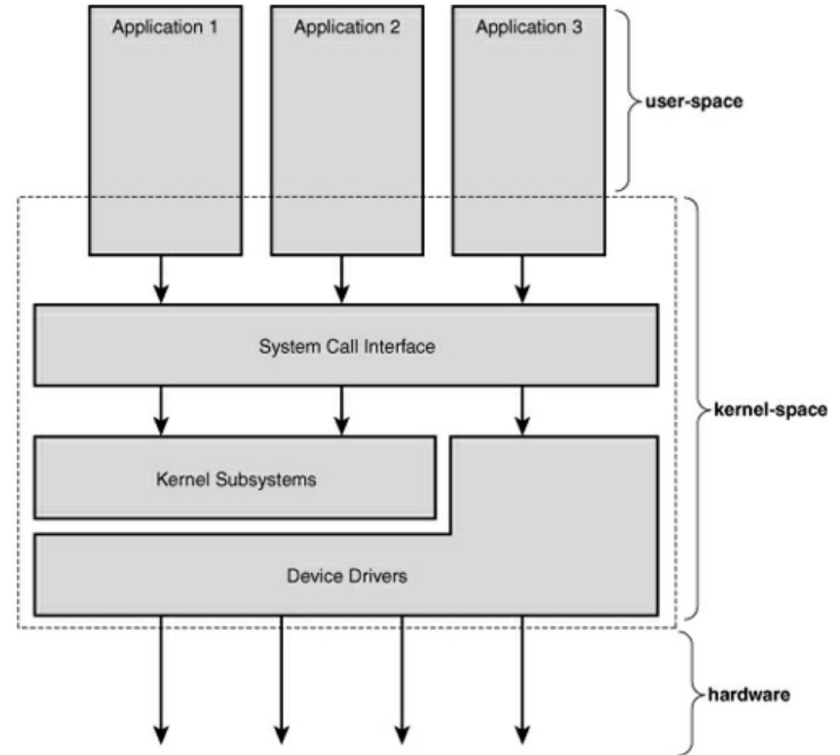
Linux Kernel Source Tree

Table 2.1. Directories in the Root of the Kernel Source Tree

Directory	Description
arch	Architecture-specific source
block	Block I/O layer
crypto	Crypto API
Documentation	Kernel source documentation
drivers	Device drivers
firmware	Device firmware needed to use certain drivers
fs	The VFS and the individual filesystems
include	Kernel headers
init	Kernel boot and initialization
ipc	Interprocess communication code
kernel	Core subsystems, such as the scheduler
lib	Helper routines
mm	Memory management subsystem and the VM
net	Networking subsystem
samples	Sample, demonstrative code
scripts	Scripts used to build the kernel
security	Linux Security Module
sound	Sound subsystem
usr	Early user-space code (called initramfs)
tools	Tools helpful for developing Linux
virt	Virtualization infrastructure

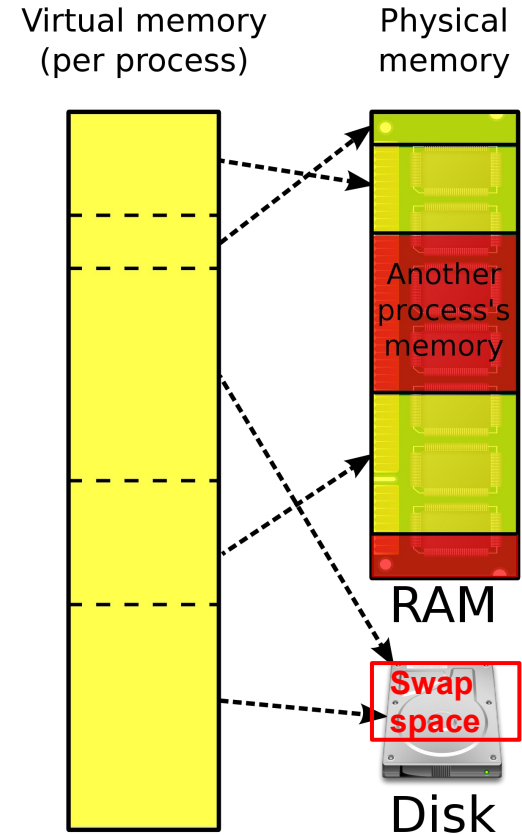
User Space Vs. Kernel Space

- Applications run in user space
 - Includes bash, nano, Firefox, GUI (e.g., GNOME)
 - Applications have individual memory spaces
 - Cannot view other apps' memory
- Kernel code runs in kernel space
 - One memory space for kernel code
 - Kernel can access hardware without restrictions
 - Device drivers
 - macOS has gone a different route



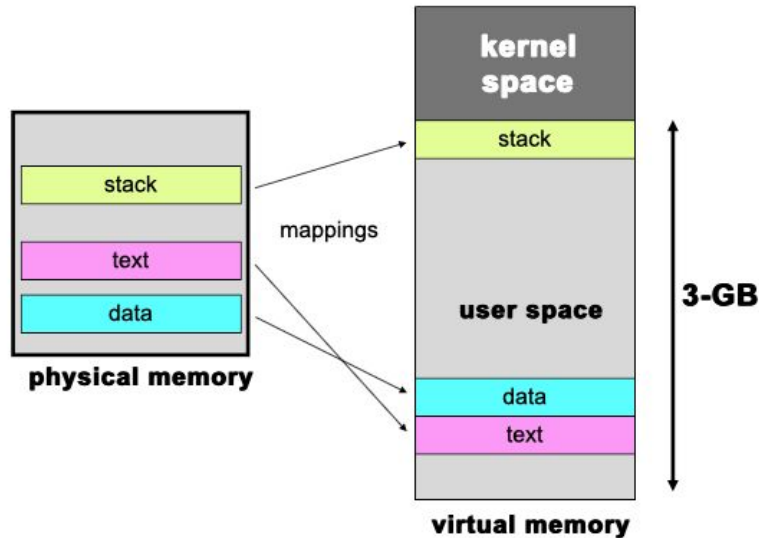
Virtual Memory

- **Virtual memory** means we create addresses that are abstracted from the physical memory (RAM)
- Virtual memory can therefore exceed the actual size of physical memory (RAM)
- Use disk (slower) for excess storage
- Virtual memory abstraction & management makes it so we don't notice where data resides (RAM vs. disk)
- Instead, it generally feels as though everything is in RAM



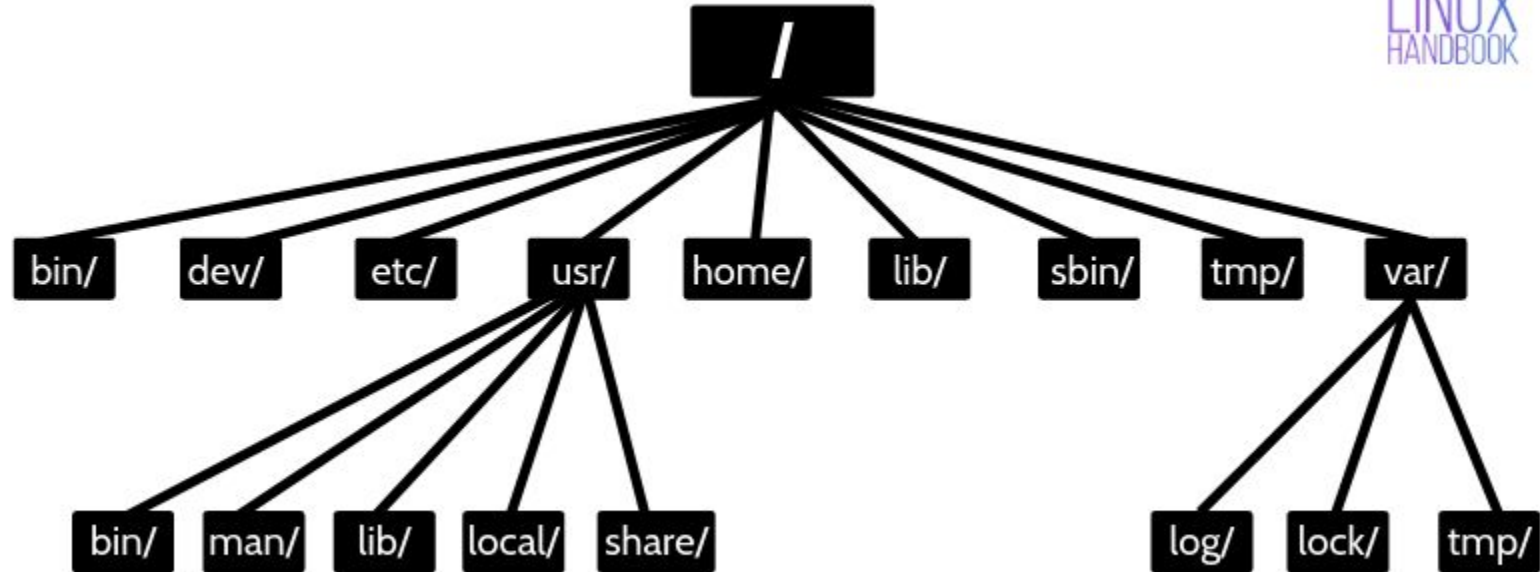
User Space & Kernel Space In Physical & Virtual Memory

Application memory-mapping



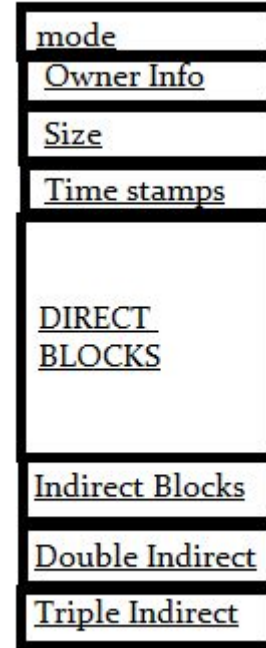
Linux Directory Structure (File System)

LINUX
HANDBOOK



File System Details

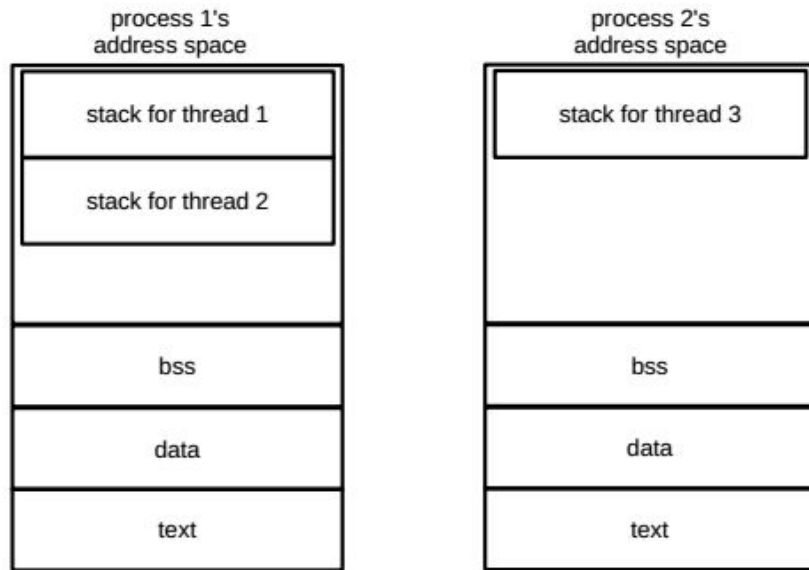
- **inodes** are the core data structures
- Files & directories are denoted by inodes
- **inode table** connects files & directories to inode data structures



inode for file
<https://www.slashroot.in/inode-and-its-structure-linux>

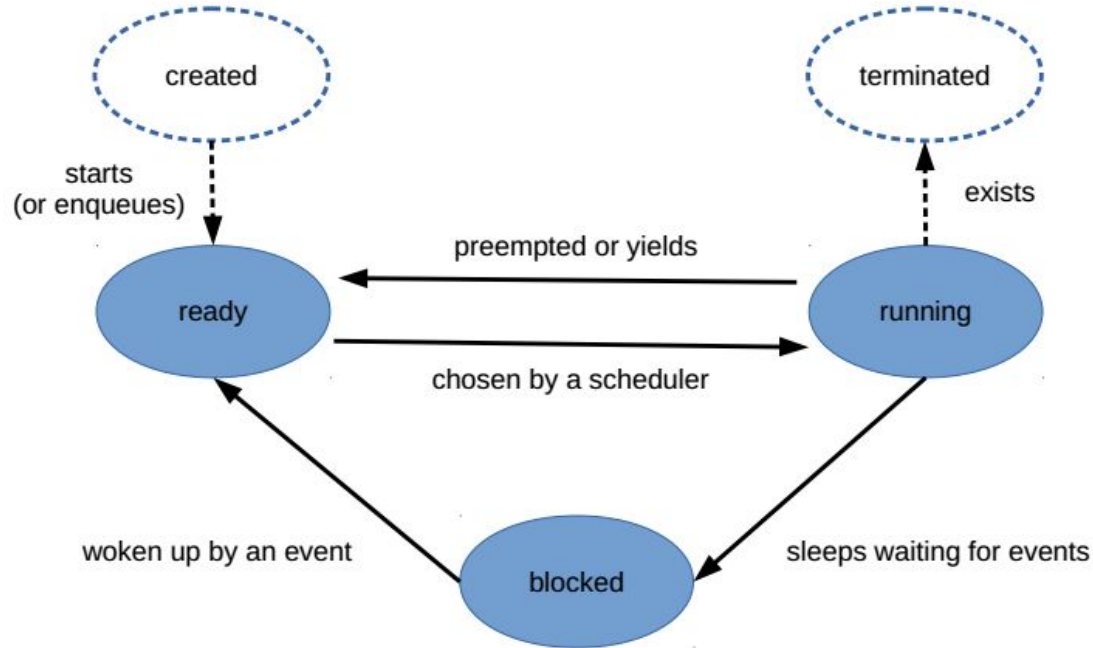
Processes

- **Processes** are realizations of programs
- Get private address space
- Processes can spawn multiple threads
 - Threads share resources between each other (e.g., memory space) and the process



<https://helix979.github.io/jkoo/post/os-scheduler/>

Process Management



Interesting Linux Kernel Facts

- Can't use the C library, standard C headers
- Programmed using GNU C
 - Must use gcc, or another compiler that can support required features
- No memory protection
- Preemptive

No C Library or Standard Headers

- Speed concerns
- Size concerns
- Kernel contains recreations of common functions

GNU C - Inline Functions

- Function internals inserted
- Don't have to call and return
- Higher code size
- Higher performance
- When time is important

```
inline void displayNum(int num) {  
    cout << num << endl;  
}  
  
int main() {  
    displayNum(5);  
    displayNum(8);  
    displayNum(666);  
}
```

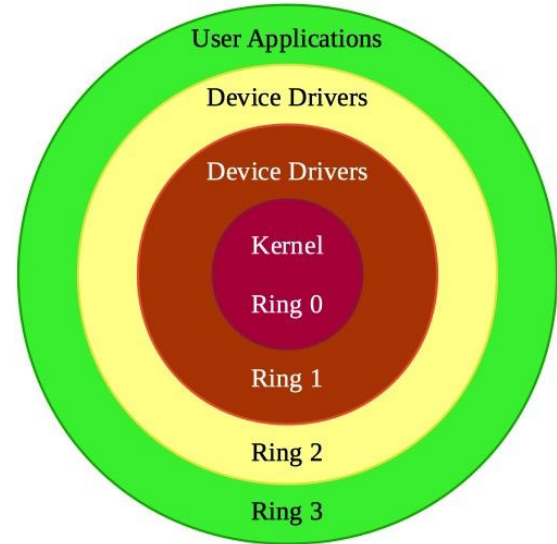
Compilation

```
inline void displayNum(int num) {  
    cout << num << endl;  
}  
  
int main() {  
    cout << 5 << endl;  
    cout << 8 << endl;  
    cout << 666 << endl;  
}
```

<https://www.programiz.com/cpp-programming/inline-function>

Lack of Memory Protection in Kernel

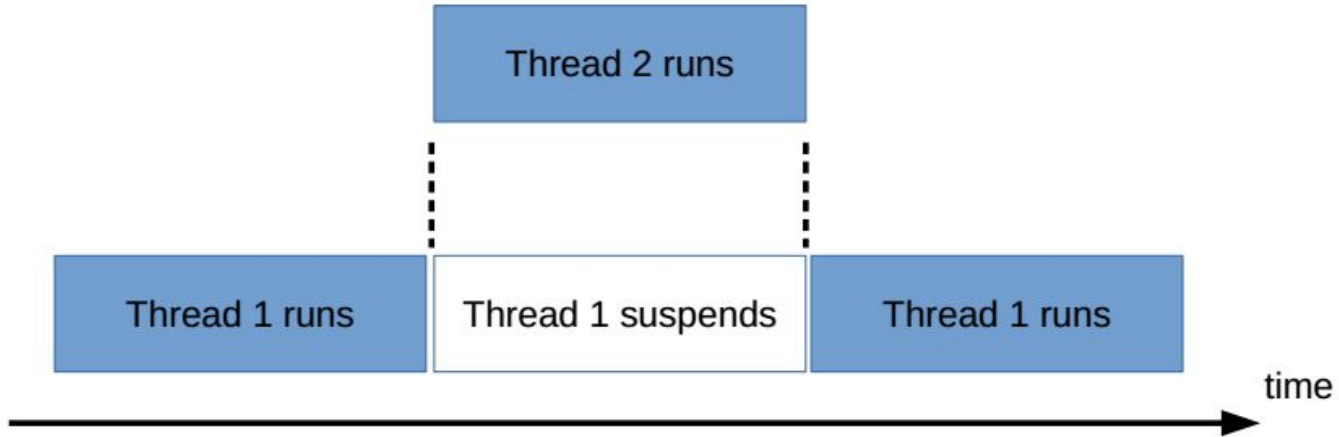
- User-space applications can't reach kernel memory
- However, kernel can access kernel space and user space
- What could go wrong?



Standard IA Protection Rings

<https://medium.com/swlh/negative-rings-in-intel-architecture-the-security-threats-youve-probably-never-heard-of-d725a4b6f831>

Kernel is Preemptive



Next Lecture:

How Does Linux Deal With Processes?