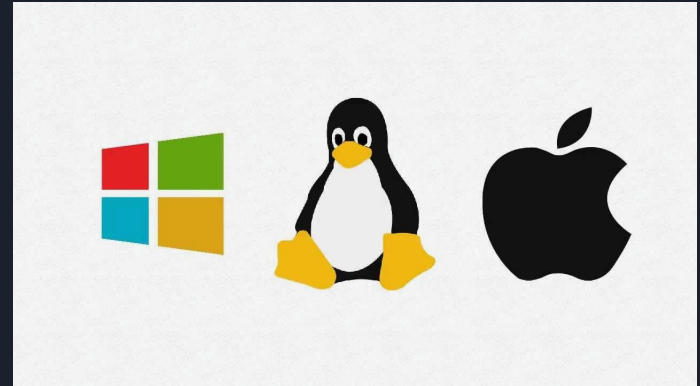
A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Deep Dive into Operating System Kernels and Evolution

By Christian Marquardt, Michael Ressler, Khbindar Arumugam, Eric Kempf

Project Overview

- ❖ Looking into the history of different operating systems and how they came to be
- ❖ Looking in depth into how the kernels work on different OS
- ❖ Looking at similarities/differences between OS and why to choose one over the other

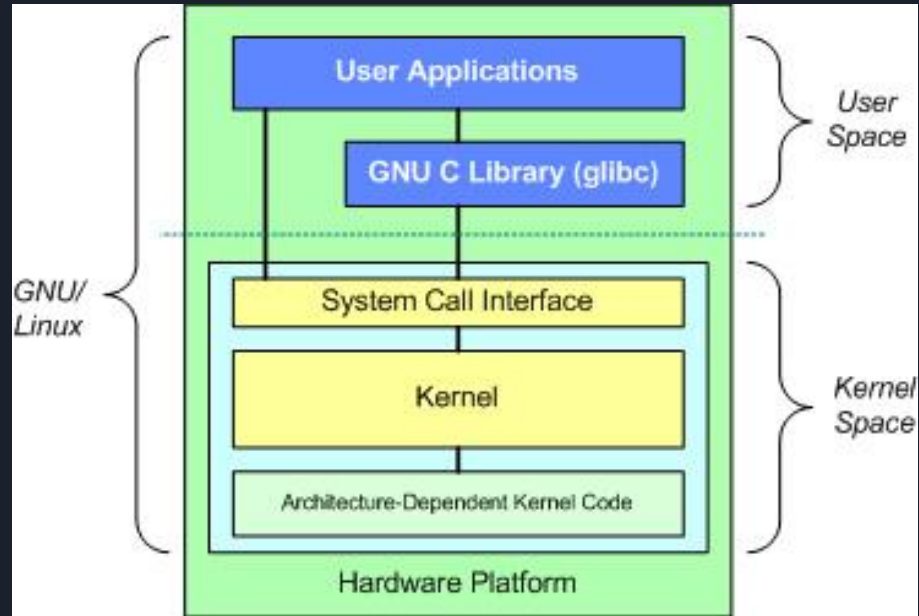


Linux



- Linux was developed by Linus Torvalds and the Free Software Foundation.
- The first official Linux version was version 0.02
- New versions of Linux are created because of evolution.
- Linux operating systems are not produced by one organization or person.
- For the past few years, the rate at which Linux released was 2 $\frac{3}{4}$ months.
- A Linux distribution does all the work for their users, making it possible to use the system right away.
 - RedHat Enterprise Linux, Fedora, Ubuntu, Debian, and Linux Mint are a few examples of the most popular Linux distributions available.

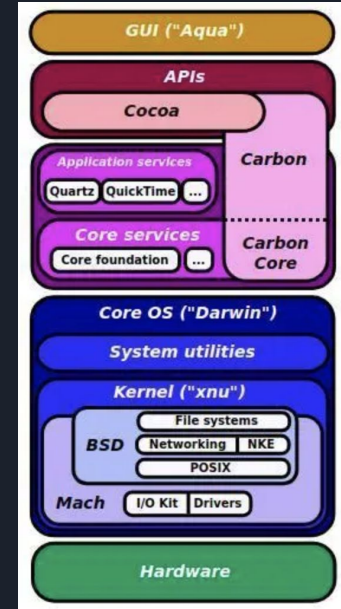
Linux cont.



MacOS



- Hybrid Kernel Approach
 - “XNU is Not Linux”
 - Monolithic && Micro
- Kernel Environment
 - Mach and FreeBSD Architecture
 - Two address spaces
 - User and Kernel space
- No Completely Closed Source
 - Kernel source code is open source
 - APIs are closed
- PureDarwin
 - Utilizes Darwin
 - Increased use and accessibility



Windows



- ❖ Before Windows: MS-DOS > NT kernel
- ❖ Hybrid Architecture: Kernel > Executive > Supervisor > User
 - Split up duties for the kernel to separate workload
- ❖ Scheduler: Assigns threads different priority and determines which should run
 - Has improved in efficiency over time (DFSS, similar to CFS Linux)
- ❖ Processor groups > CPU sets
 - Allow process to take control of block of CPUs
- ❖ Dispatcher database lock: Bottlenecked efficiency of multiprocessor/thread system
 - Changed to per object locks > increased performance by 290%
- ❖ Security: Kernel Data Protection, Hardware Enforced Stack Protection
 - Utilizes virtualization to make the kernel more secure

Focus on code reusability and versatility is a major design philosophy of windows. Sharing of the kernel between different architectures is what makes Windows such a powerful OS.



Key Takeaways

- ❖ Operating systems have come a long way since they were made
 - Performance, security, stability
- ❖ Something out there for everyone
 - MacOS - Reliable, tailored to the hardware
 - Windows - Versatile kernel, running on many architectures
 - Linux - A distribution for everyone
- ❖ The best OS?