CSCI 350
Ch. 1 – Introduction to OS

Mark Redekopp
Ramesh Govindan and Michael Shindler

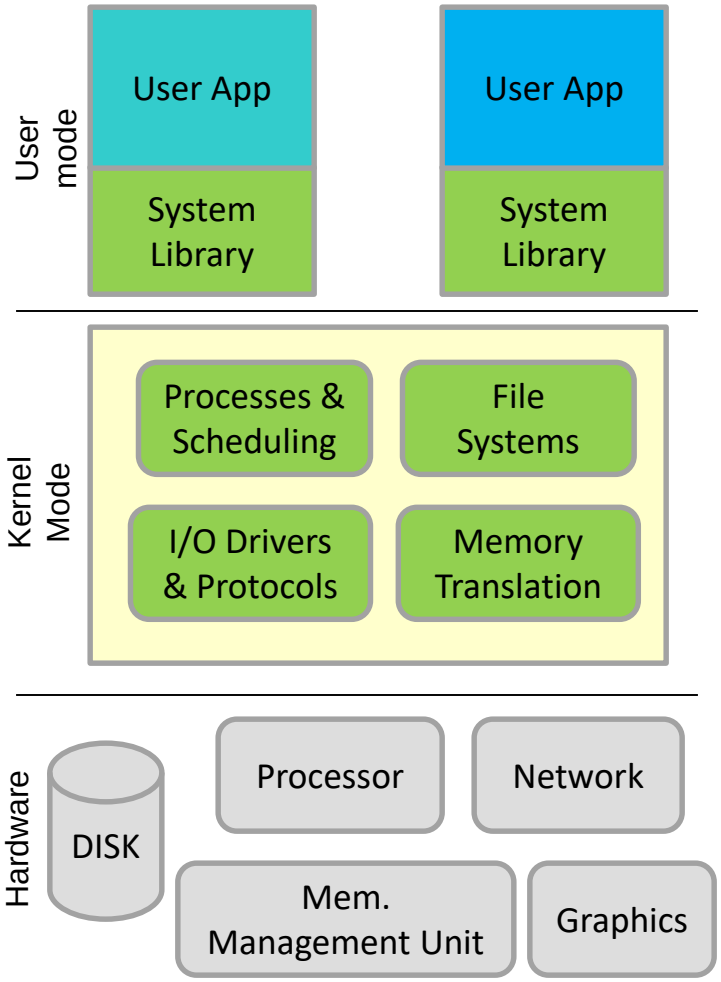
Syllabus

- Website (http://bits.usc.edu/cs350)
- People
- Projects
  - PintOS
  - Policies
- Grading & Exams

WHAT IS AN OPERATING SYSTEM

Definition

- A piece of software that manages a computer's resources
- What resources need managing?
  - CPU (threads and processes)
  - Memory (Virtual memory, protection)
  - I/O (Abstraction, interrupts, protection)



HISTORY OF OPERATING SYSTEMS

Evolution

- Single job systems
  - https://www.youtube.com/watch?v=XetHU_pAWOo
- Batch systems: OS little more than a library for I/O
  - Execute the next program while outputting results from the previous
  - Better than before since computer wasn't idle when processing
- Multi-tasking: DMA, direct I/O access, virtual machines
  - Avoided I/O waiting, multiplex CPU and I/O
  - Virtual machines were developed around this time
- Time sharing: Support interactive use of computers
  - Isolation (privileges), the process abstraction, memory hardware, virtual memory
- PCs: bitmapped displays, window management
- Mobile: SSDs, sensors/GPS, touch screen
- Servers: Virtualization, optimize for specific tasks

As a result of Moore's Law, features available on higher end devices find their way into lower end devices over time. Supercomputer -> server -> desktop -> mobile

Kinds of Operating Systems

- General Purpose
  - Laptop, Desktop
- Server
  - User and other access policies
  - Supporting virtualization (hypervisor)
- Mobile
- Embedded
  - Hard and Soft Real-time
    - Hard Real-time: Missing a deadline results in failure
    - Soft Real-time: Performance/usefulness degrades if deadlines missed
    - Scheduling, priority, and deterministic latency

Genealogy of Operating Systems

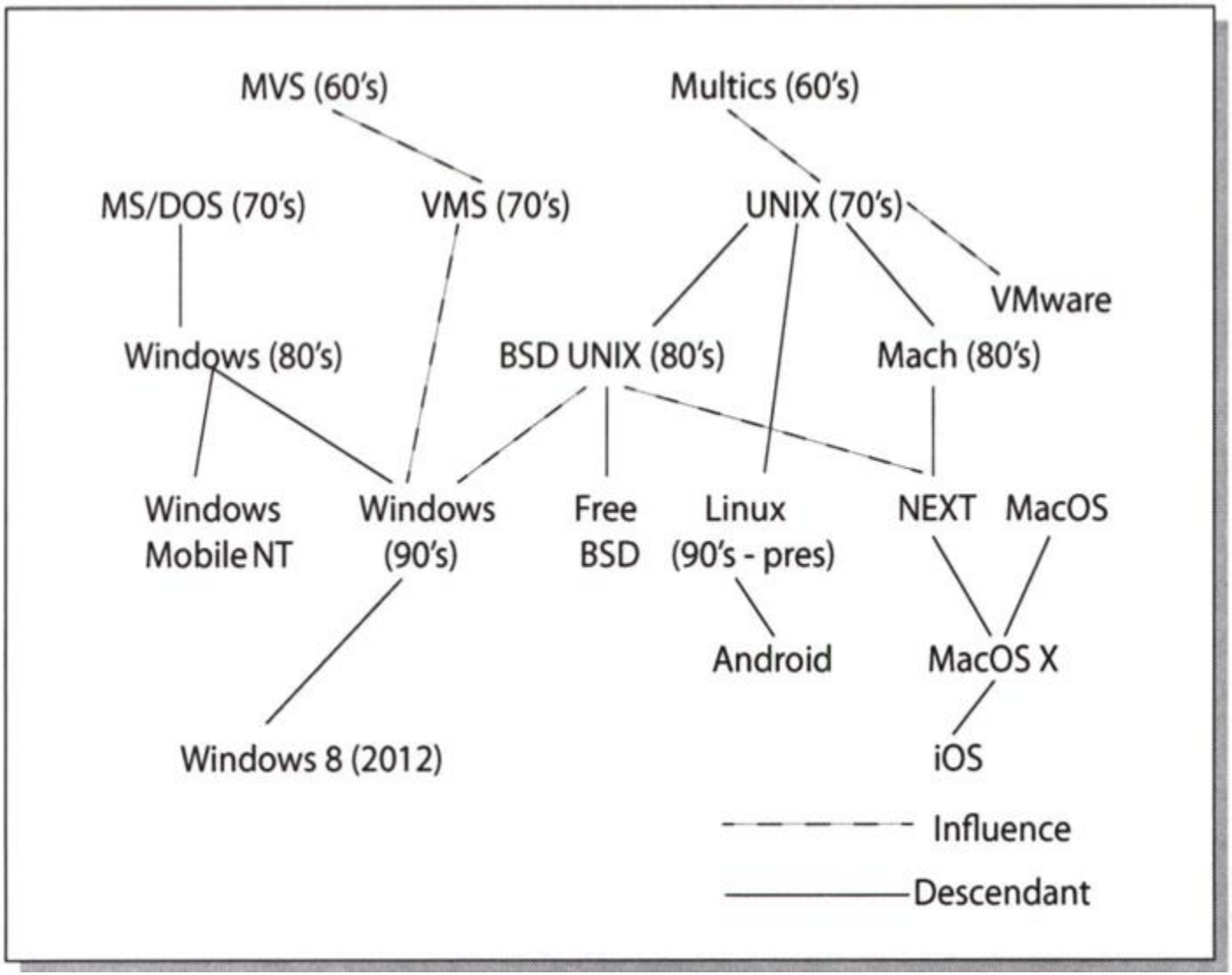


Figure 1.9, OS:PP 2nd Ed.

Roles

- Textbook regularly revisits 3 roles an OS plays:
- Referee
  - Protection against other applications
  - Enforce fair resource sharing
    - Why doesn't an infinite loop require a reboot?
- Illusionist (Virtualization)
  - Each program thinks it is running separately
  - Each program thinks it has full access to computer's resources (or unlimited resources)
- Glue
  - Common services (such as copy/paste)
  - Files can be read by any application
  - UI routines for look & feel
  - Separate applications from hardware
    - so you don’t need to know which keyboard, disk drive, etc

Case Study of Roles

- What roles {Referee, Illusionist, Glue} would be relevant and important in the following systems?
- Web browser
- Cloud computing
- Multiuser database
- The Internet

Case Study of Roles

- Web browser
  - Referee (execution of scripted code), glue (display and network functionality)
  - Illusionist (handling of network errors)
- Cloud computing
  - Illusionist (no need to know where data is stored) and glue (to provide APIs and access) to data
  - Referee and Illusionist (virtualization, separation)
- Multiuser Database
  - Illusionist (replication and transactions)
  - Referee (access rights)
- Internet
  - Illusionist (DNS translation, IP routing, Reliability)
  - Referee (Quality-of-Service, congestion, denial of service)
  - Glue (Common API, standards, etc.)

OS Design Criteria (Metrics)

- Reliability (availability)
- Security/Privacy
- Performance
- Portability & Adoption

Reliability vs. Availability

- Reliable: Outputs are as intended for the given set of inputs
  - If outputs are not as intended we say the system has failed
- Availability: System can do work
- Available does not imply reliable
  - System can be available but not reliable (bugs!)
  - System can be reliable yet not available
    - DoS, Crash after every instruction but saves results

Security

- Security means avoiding compromised states
  - Example: Execution of malware/virus
- Privacy implies only authorized users can access certain data
- Security is hard to achieve without sacrificing other important aspects like performance (e.g. encryption, translation, etc.)

Performance

- Scheduling and Fairness
  - Predictability
- Throughput vs. response time

Portability

- OS can abstract applications from various hardware changes
  - OS should play middle man and provide an API or AMI (Abstact Machine Interface)
- Can OS itself be abstracted from hardware changes (i.e. easily ported to another HW platform)
  - Many OSs use a HAL (hardware abstraction layer) so that the bulk of the code need not change as it is migrated to a new platform
- Adoption: The more devices that can be supported the greater the chance of adoption
  - Network effect: Adoption based on
  - If good apps are available, OS will be adopted. But good apps may not be written unless the OS is widely adopted.