1. Introduction

Sungyoung Lee

College of Engineering
KyungHee University

Contents

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 make solving user problems easier
  - Make the computer system convenient to use

- Use the computer hardware in an efficient manner

Computer System Components

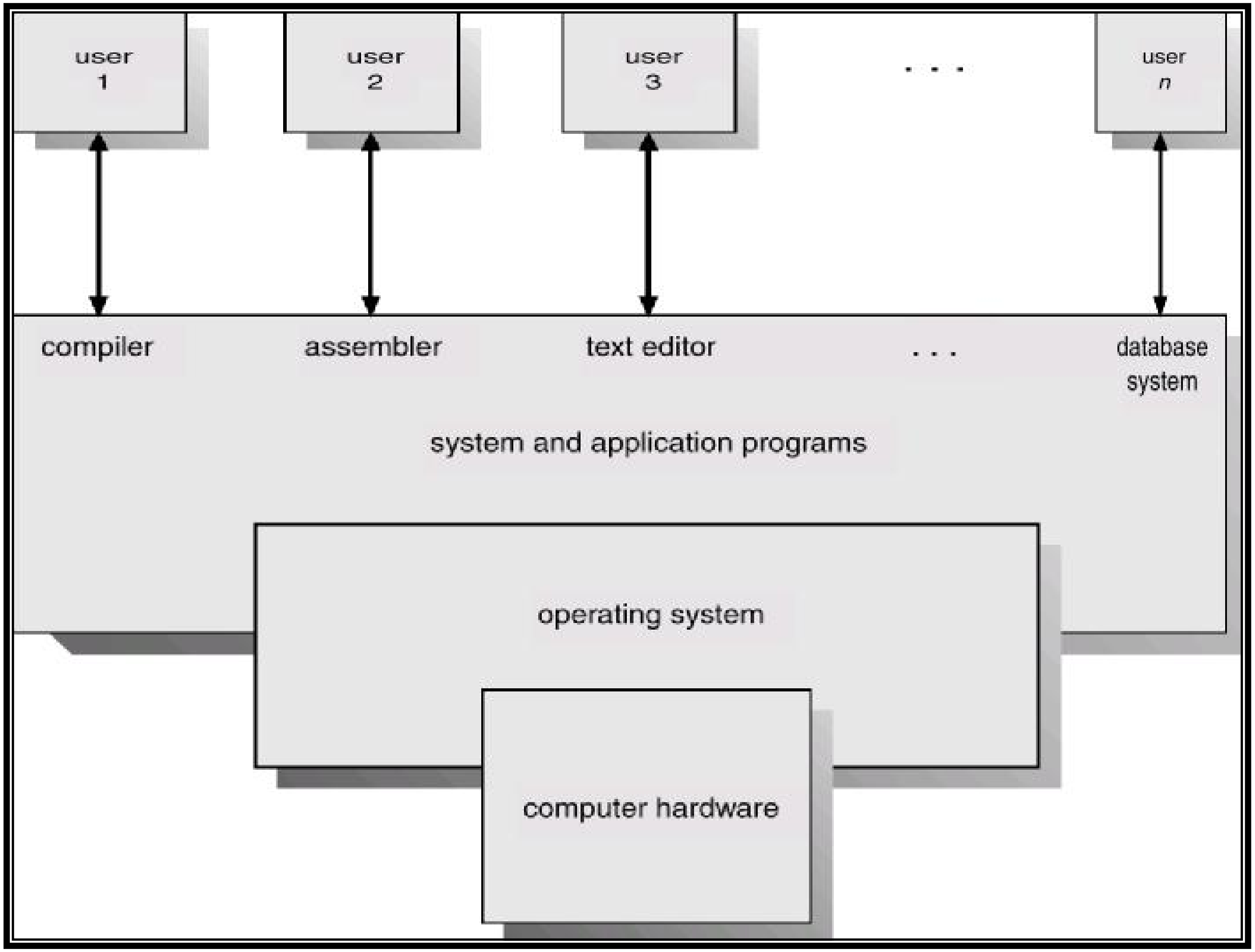
1. Hardware – provides basic computing resources (CPU, memory, I/O devices)

2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users

3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs)

4. Users (people, machines, other computers)

Abstract View of System Components



Operating System Definitions

- Resource allocator – manages and allocates resources

- Control program – controls the execution of user programs and operations of I/O devices

- Kernel – the one program running at all times (all else being application programs)

What is an OS?

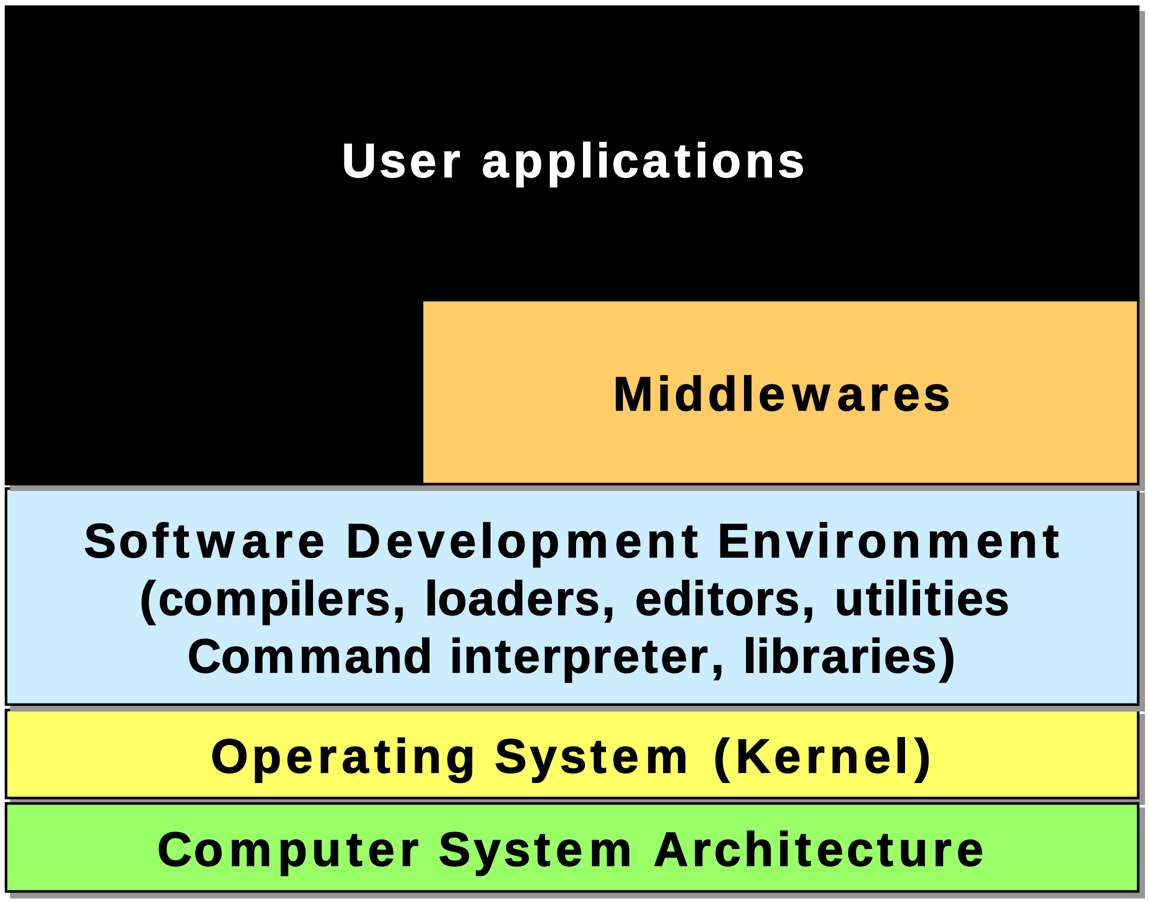
- OS is a resource manager
  - Abstraction
  - Sharing
    - Time multiplexing
    - Space multiplexing
  - Protection
  - Fairness
  - Performance

Resources

- **CPU**
- **Memory**
- **I/O devices**
- ...

- OS provides the program execution environment

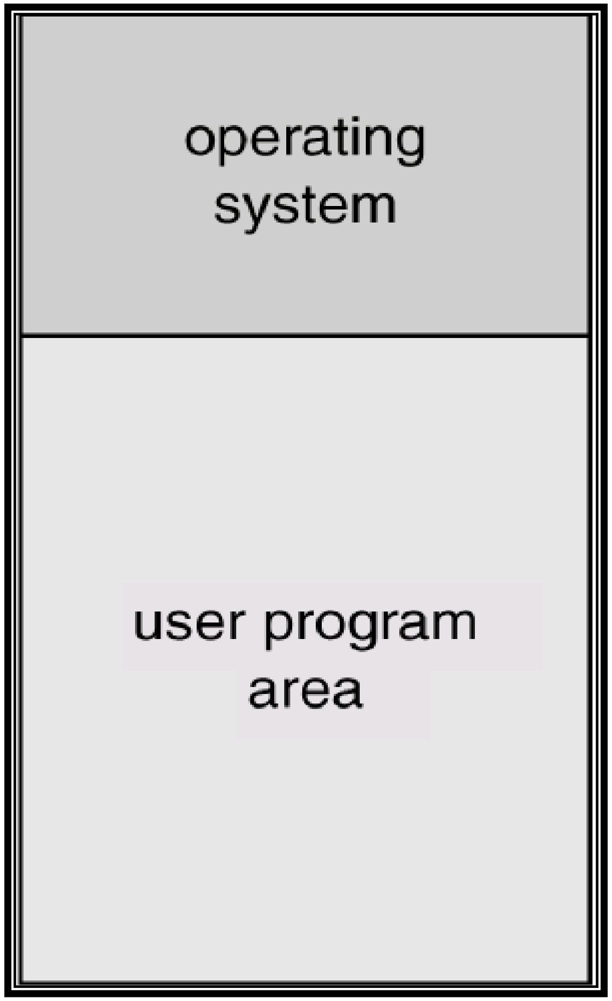
System Software Layers



Mainframe Systems

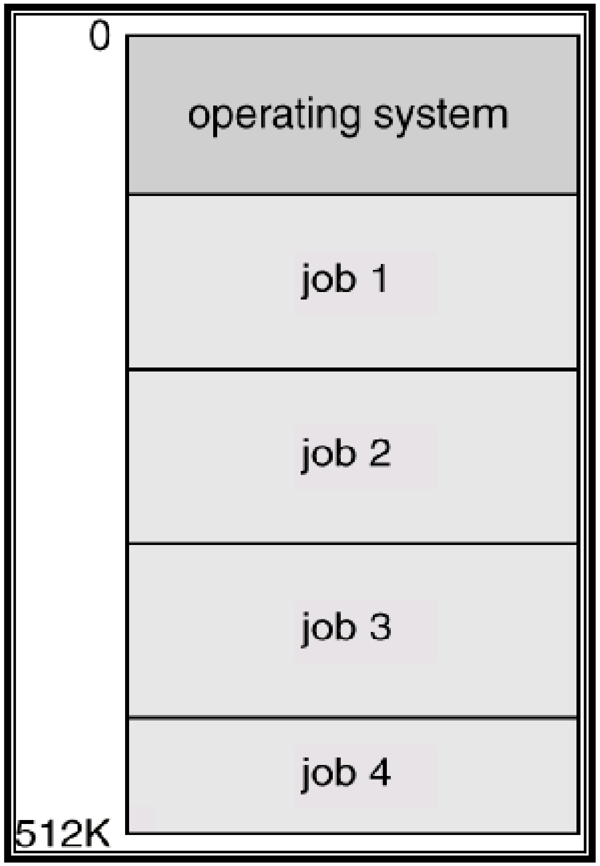
- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers pack to monitor

Memory Layout for a Simple Batch System



Multiprogrammed Batch Systems

Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.



OS Features Needed for Multiprogramming

- I/O routine supplied by the system
- Memory management – the system must allocate the memory to several jobs
- CPU scheduling – the system must choose among several jobs ready to run
- Allocation of devices

Time-Sharing Systems–Interactive Computing

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory)
- A job swapped in and out of memory to the disk
- On-line communication between the user and the system is provided; when the operating system finishes the execution of one command, it seeks the next “control statement” from the user’s keyboard
- On-line system must be available for users to access data and code

Terminology

- Batch, Multiprogramming, Time-sharing(or Multitasking)
  - von Neumann architecture
- Job scheduling vs. CPU scheduling
- Job, Task, Process
- Concurrent, Simultaneous, Parallel

Desktop Systems

- *Personal computers* – computer system dedicated to a single user
- I/O devices – keyboards, mice, display screens, small printers
- User convenience and responsiveness
- Can adopt technology developed for larger operating system' often
- Individuals have sole use of computer and do not need advanced CPU utilization of protection features
- May run several different types of operating systems (Windows, MacOS, UNIX, Linux)

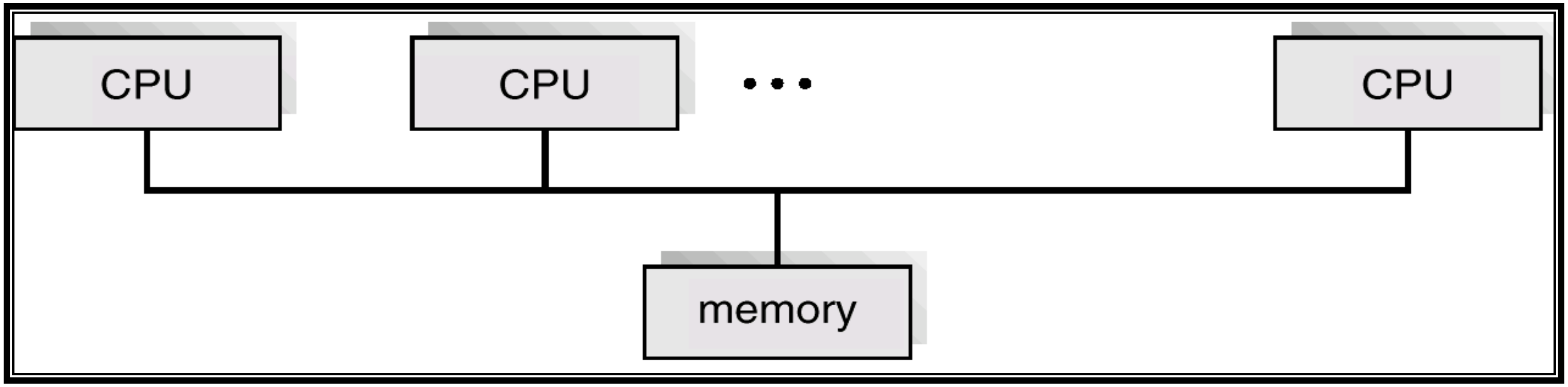
Parallel Systems

- Multiprocessor systems with more than one CPU in close communication
- *Tightly coupled system* – processors share memory and a clock; communication usually takes place through the shared memory
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - graceful degradation
    - fail-soft systems (fault-tolerant systems)
- *Cf) Co-processor or controller*

Parallel Systems (Cont'd)

- *Symmetric multiprocessing (SMP)*
  - Each processor runs and identical copy of the operating system
  - Many processes can run at once without performance deterioration
  - Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - Each processor is assigned a specific task; master processor schedules and allocated work to slave processors
  - More common in extremely large systems

Symmetric Multiprocessing Architecture



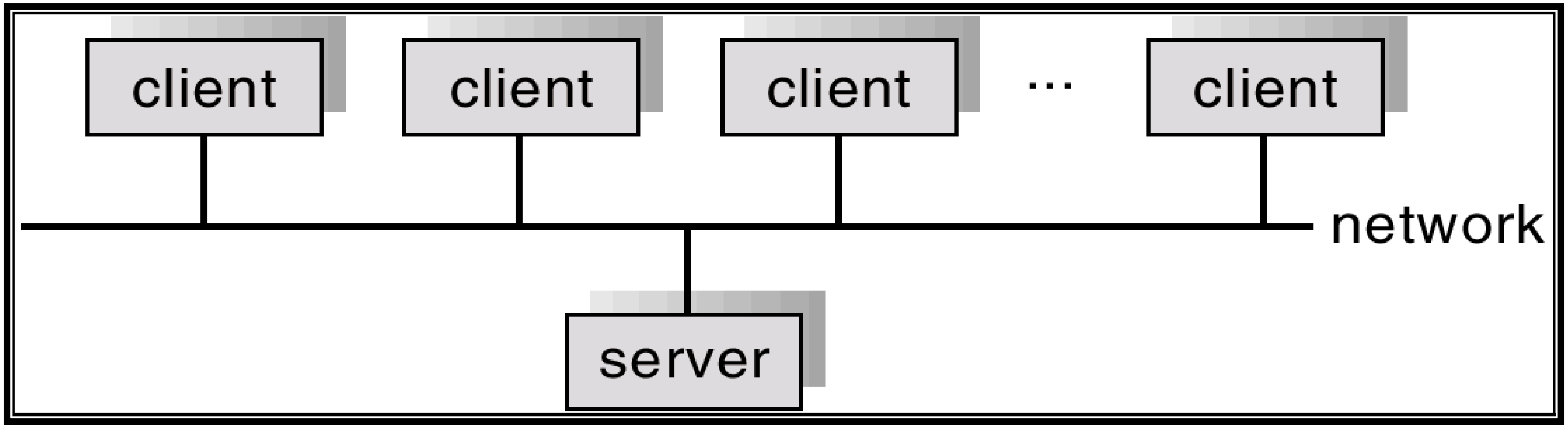
Distributed Systems

- Distribute the computation among several physical processors
- *Loosely coupled system* – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines
- Advantages of distributed systems
  - Resources Sharing
  - Computation speed up – load sharing
  - Reliability
  - Communications

Distributed Systems (Cont'd)

- Requires networking infrastructure
- Local area networks (LAN) or Wide area networks (WAN)
- May be either client-server or peer-to-peer systems

General Structure of Client-Server



Clustered Systems

- Clustering allows two or more systems to share storage
- Provides high reliability (or high availability)
- *Asymmetric clustering*: one server runs the application while other servers standby
- *Symmetric clustering*: all N hosts are running the application
- Cf) Grid Computing
- Cf) Distributed Lock Manager (DLM), Storage Area Network (SAN)

Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems
- Well-defined fixed-time constraints
- Real-Time systems may be either *hard* or *soft* real-time

Real-Time Systems (Cont'd)

- Hard real-time:
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems

- Soft real-time
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features

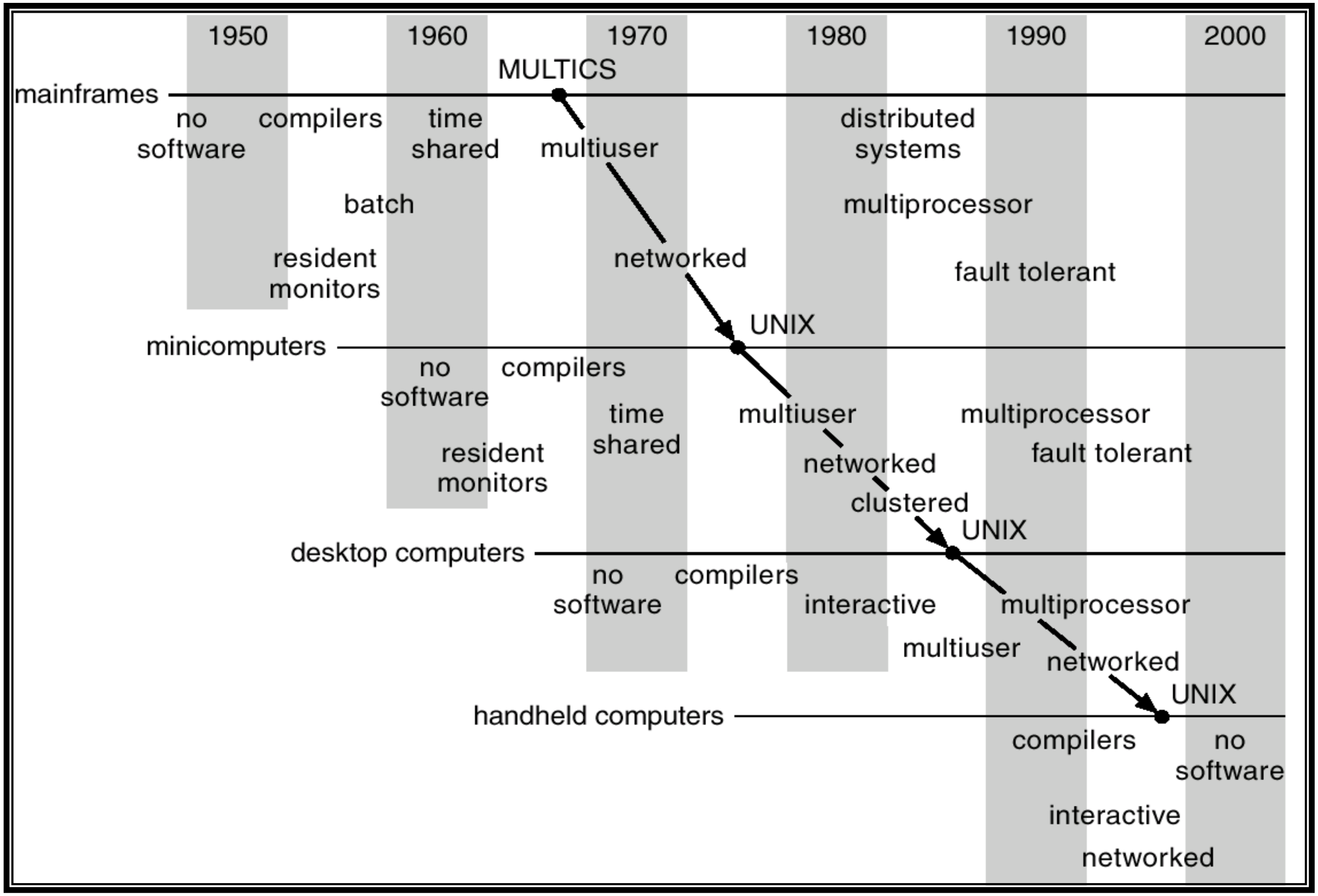
Handheld Systems

- Personal Digital Assistants (PDAs)

- Cellular telephones

- Issues:
  - Limited memory
  - Slow processors
  - Small display screens

Migration of Operating-System Concepts and Features



Computing Environments

- Traditional computing
- Web-based computing
- Embedded computing

1st Generation (1945-55)

- Vacuum Tubes and Plugboards

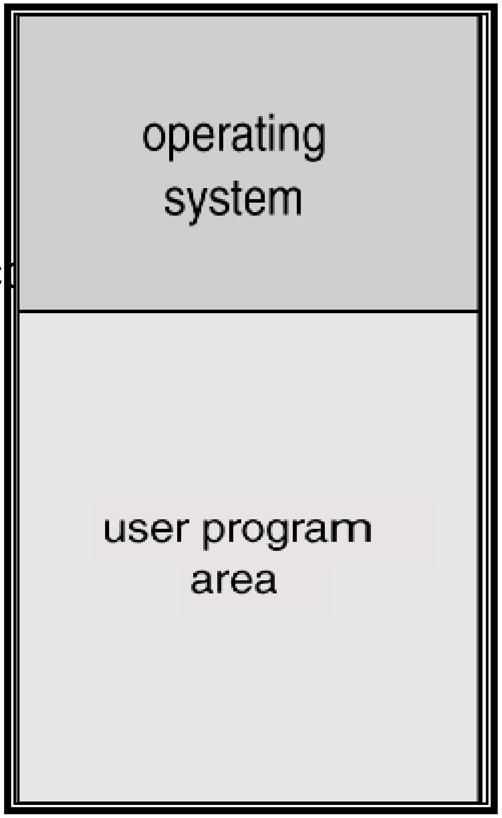
- No OS
- No Programming Languages
- No Assembly Languages

2nd Generation (1955-65)

- Transistors and Mainframes

- Batch systems
  - One job at a time
  - Card readers, tape drives, line printers

  - OS is always resident in memory and merely transfers a c
  - CPU is underutilized due to the bottleneck in I/O

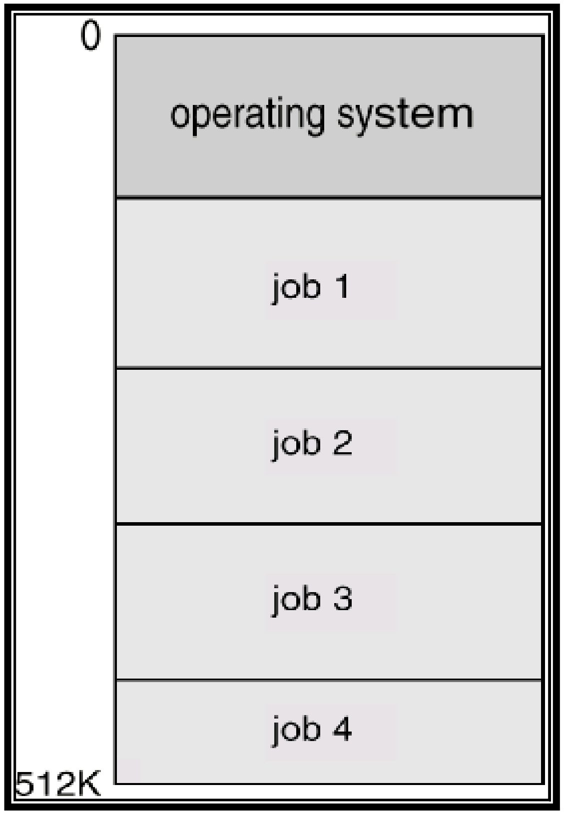


3rd Generation (1965-80)

- Integrated Circuits (ICs)
- Architectural Advances
  - Using ICs: better price/performance
  - Disk drives
  - On-line terminals
  - The notion of “Computer Architecture”
    - IBM System/360 family

3rd Generation (1965-80)

- Multiprogrammed Systems
  - Increase CPU utilization
  - OS features
    - Job scheduling
    - Memory management
    - CPU scheduling
    - Protection
  - Spooling (Simultaneous
  - Peripheral Operation On-Line)



3rd Generation (1965-80)

- Time-sharing Systems
  - Improve response time
  - OS features
    - Swapping
    - Virtual memory
    - File system
    - Sophisticated CPU scheduling
    - Synchronization
    - Interprocess communication
    - Interactive shell
    - More protection, …

4th Generation (1980-)

- LSIs & VLSIs

- Architectural Advances
  - Microprocessors: smaller and faster
  - Storages: larger and faster
  - Personal computers
  - CPU work is offloaded to I/O devices

- Modern OS Features
  - GUI (Graphical User Interface)
  - Multimedia
  - Internet & Web
  - Networked / Distributed, etc.

OS History

A long time ago,
in a galaxy far, far away, …

- IBM OS/360: Multiprogramming
- MIT CTSS (Compatible Time-Sharing System)
- MIT, Bell Labs, GE, MULTICS
  (MULTiplexed Information and Computing Service)

And UNIX was born in 1969

OS History: UNIX (1969-85)

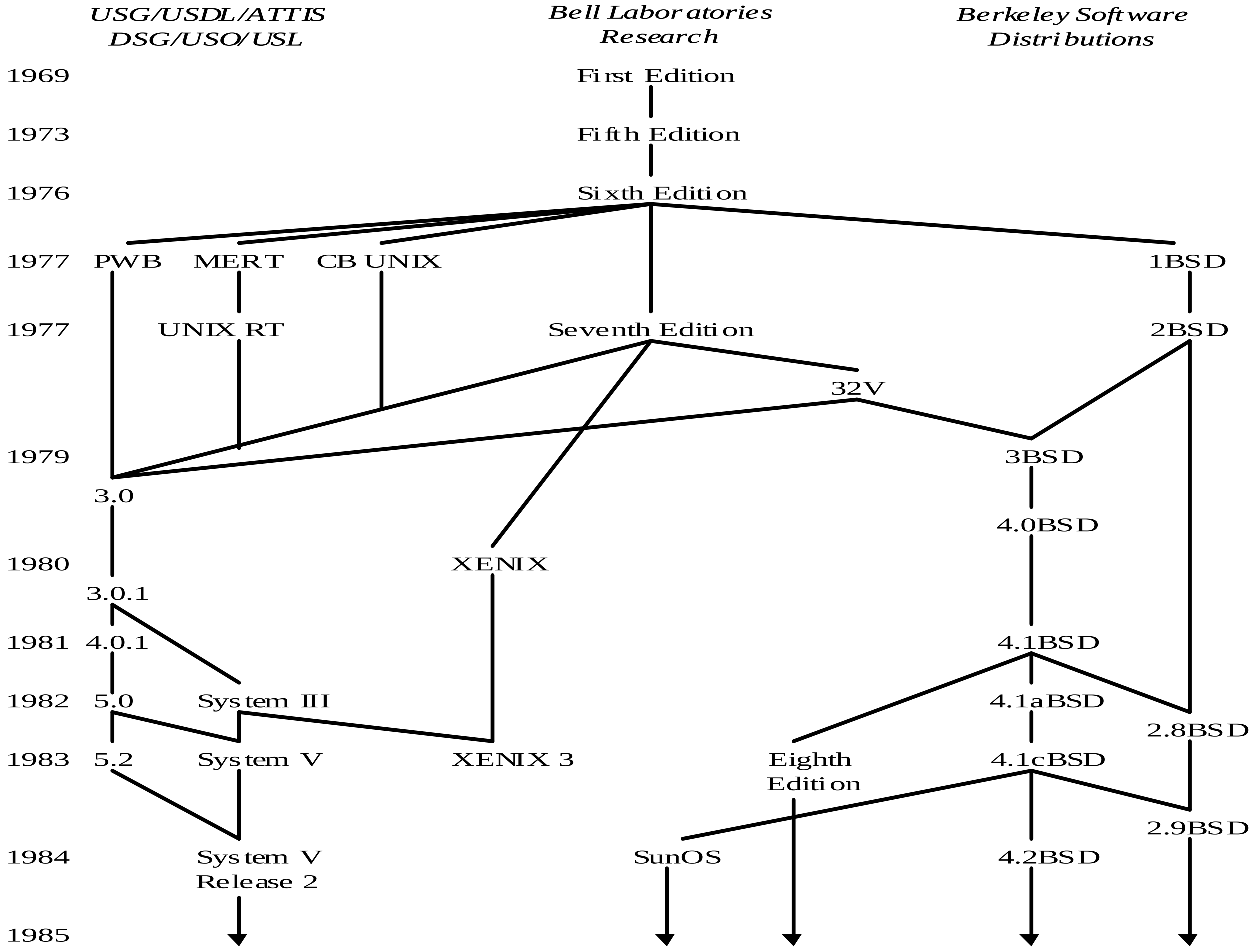


Figure 1.1 The UNIX system family tree, 1969-1985

OS History: UNIX (1985-96)

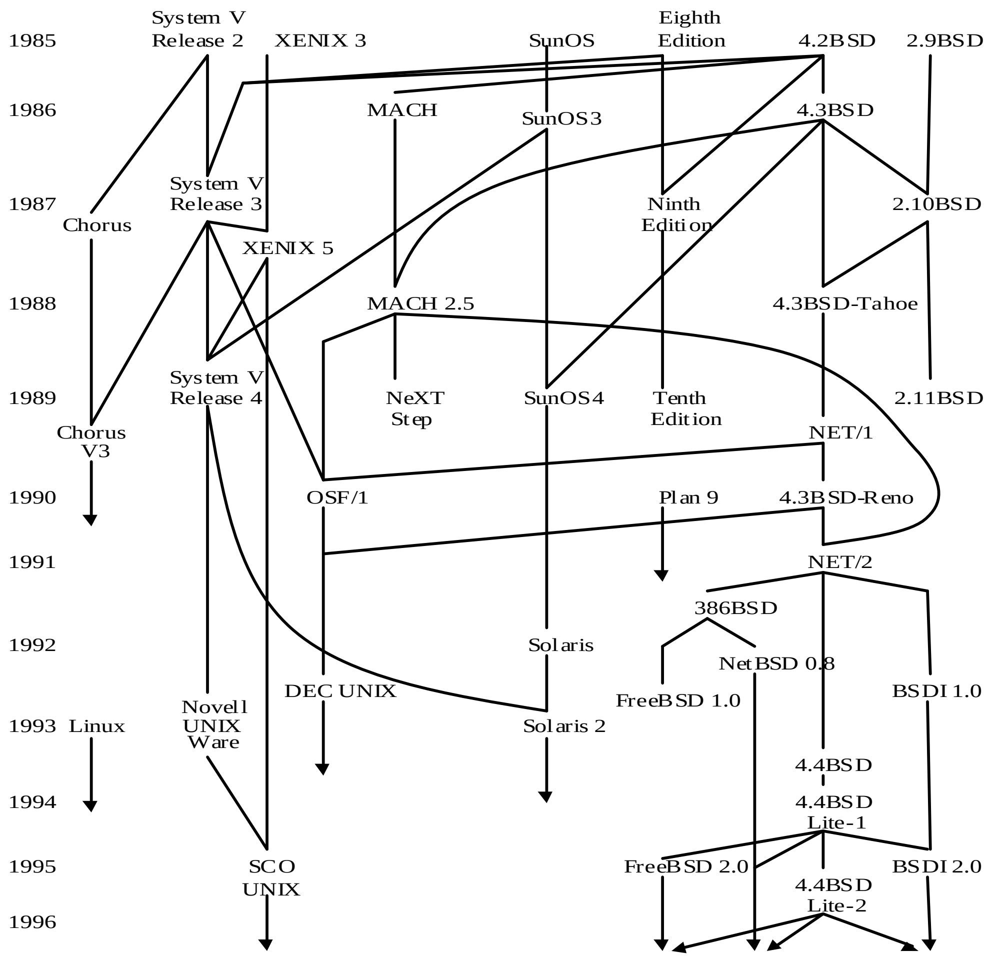


Figure 1.2 The UNIX system family tree, 1986-1996

OS History: UNIX (Current)

- Sun Solaris
- HP HP-UX
- IBM AIX
- Caldera (SCO) Unixware
- Compaq (Digital) Tru64
- SGI Irix
- Linux, FreeBSD, NetBSD
- Apple Mac OS X, etc.

- Cf) POSIX

OS History: Windows & Linux

1981 82 84 86 88 90 91 92 93 94 95 96 97 98 99 00 01

MS-DOS Family

1.0 2.0 3.0 3.2 4.0 5.0 6.0 7.0

Windows Family

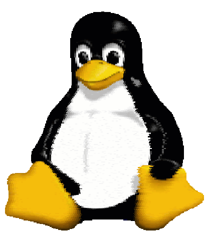
1.0 2.0 3.0 3.1 95 98 98 SE ME

Windows for WorkGroup Family

3.1 3.11

Windows NT Family

3.1 3.51 4.0 2K XP



Linux

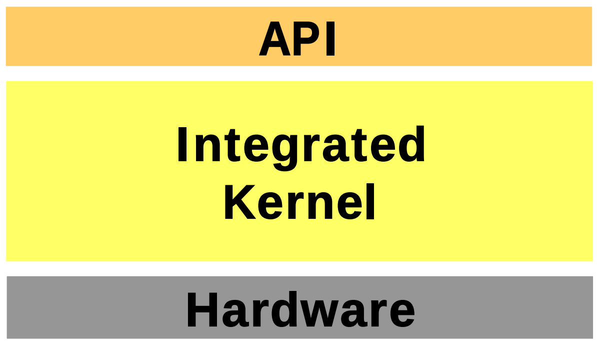
0.01 0.99 1.0 1.2 2.0 2.1 2.2 2.4

OS Classification (1)

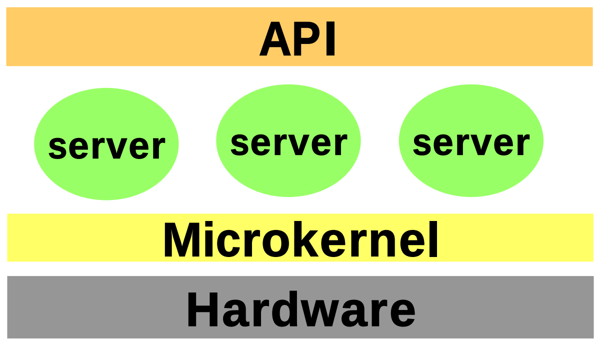
	MS-DOS	Windows 98	Windows 2000	Linux
Multi-user	X	X	O	O
Multi-task Multi-process	X	O	O	O
Multi-processor	X	X	O	O
Multi-thread	X	O	O	O

OS Classification (2)

- Monolithic Kernel
  - Function calls
  - Unixware, Solaris, AIX, HP-UX, Linux, etc.



- Micro kernel
  - Multiple servers
  - Message passing
  - Mach, Chorus, Linux mk, etc.



OS Classification (3)

- Mainframe systems
  - CTS, MULTICS, IBM MVS, VM
- Desktop systems
  - DOS, Windows, MacOS, Unix/Linux
- Multiprocessor systems
- Cluster systems
- Distributed systems
  - Amoeba(Vrije Univ.), Locus(UCLA), Grapevine(Xerox), V(Stanford), Eden(U. of Washington), Chorus/Nucleus(Inria)
- Embedded systems
  - Vertex, pSOS, VxWorks, OSE, Windows-CE, Embedded Linux
  - Company-proprietary OS (Cisco, Qualcomm, Palm, Cellvic)
- Real-time systems
  - Real-Time Linux, Spring(U. of Massachusetts), HARTS(U. of Michigan), MARUTI(U. of Maryland)