SCHOOL *of* ENGINEERING

Department of Electrical & Computer Engineering

Lecture 2: OS Structure

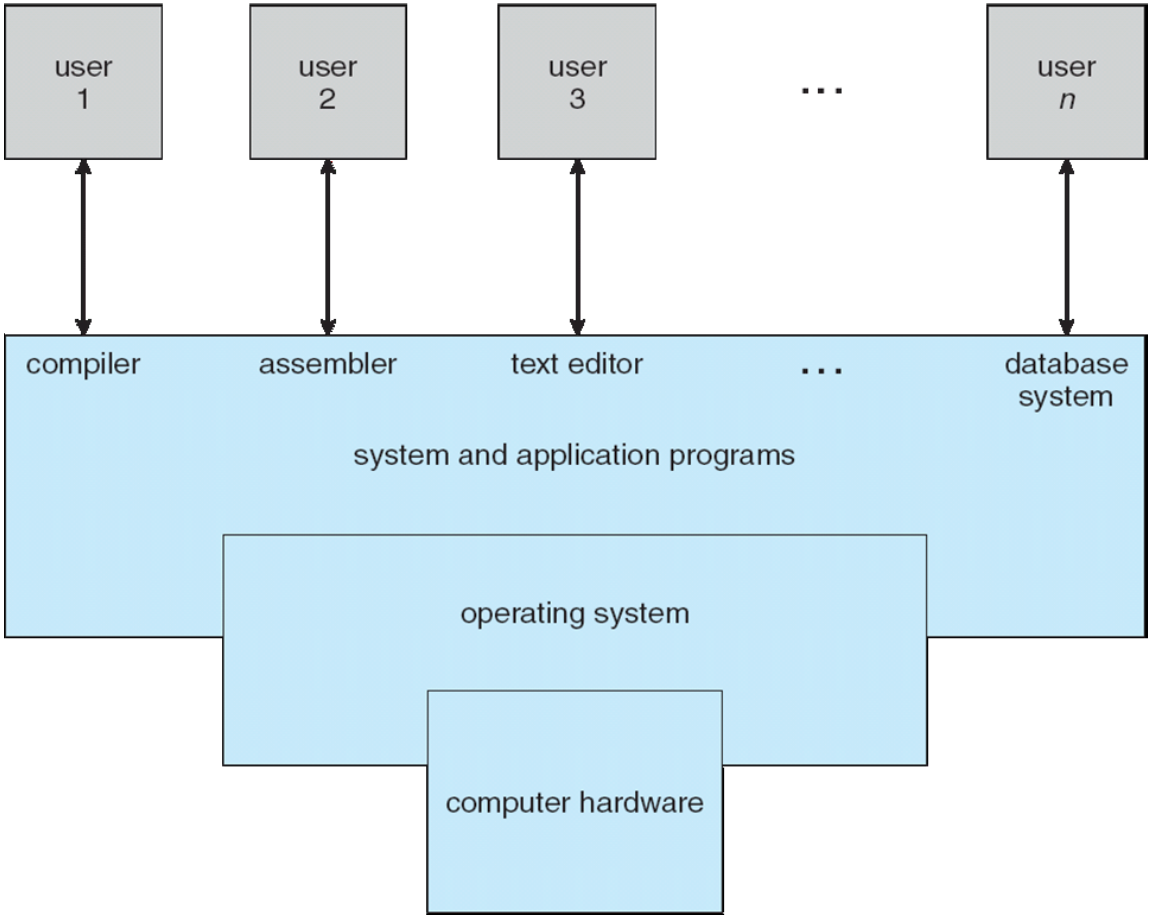
By

Dr. Edward Nava

Introduction

- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Linux
- Windows

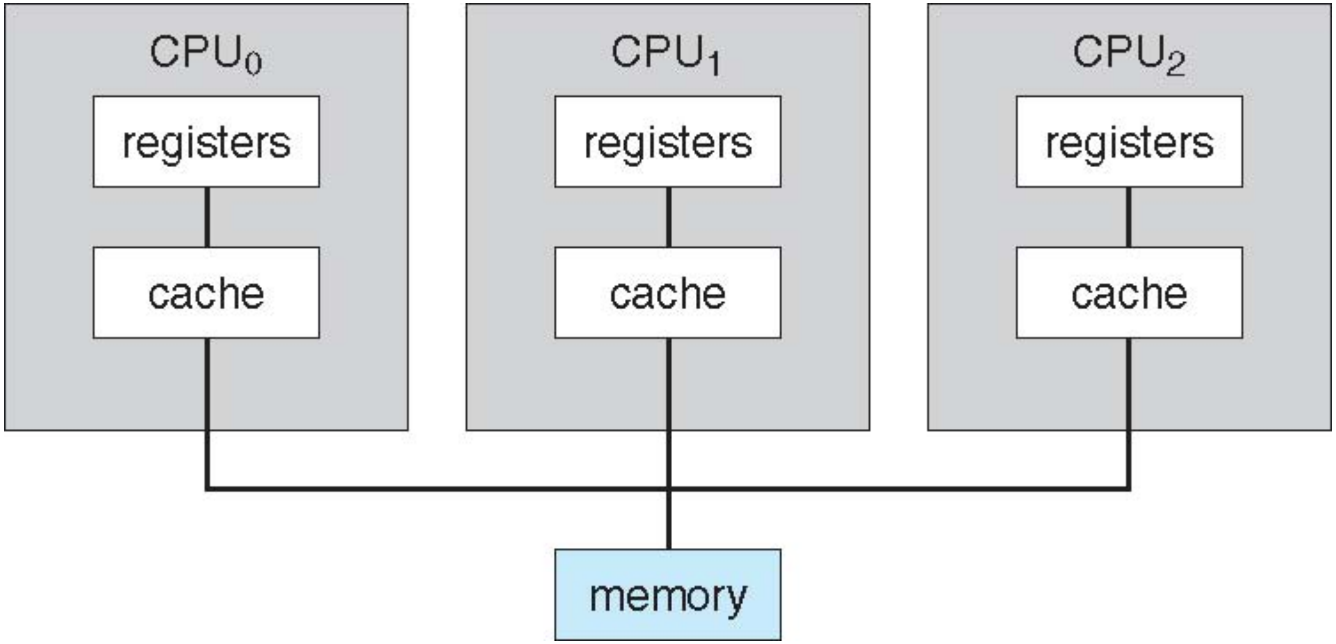
Four Components of a Computer System



Computer-System Architecture

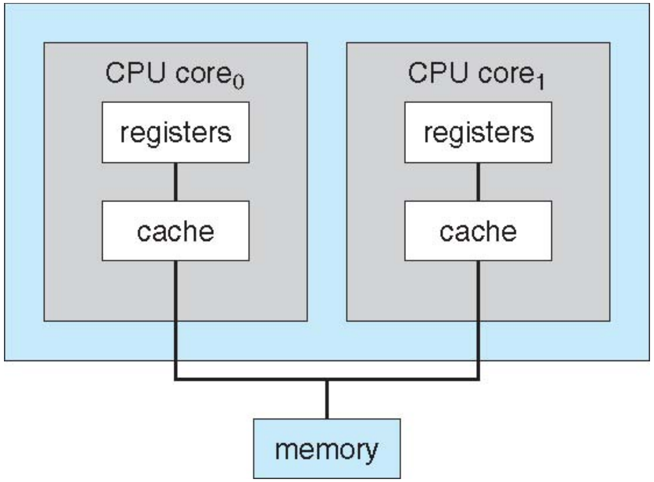
- Some systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - Advantages include
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance
  - Two types
    1. **Asymmetric Multiprocessing** – each processor is assigned a specific task.
    2. **Symmetric Multiprocessing** – each processor performs all tasks

Symmetric Multiprocessing Architecture



A Dual-Core Design

- Multi-chip and **multicore**
- Systems containing all chips
  - Chassis containing multiple separate systems



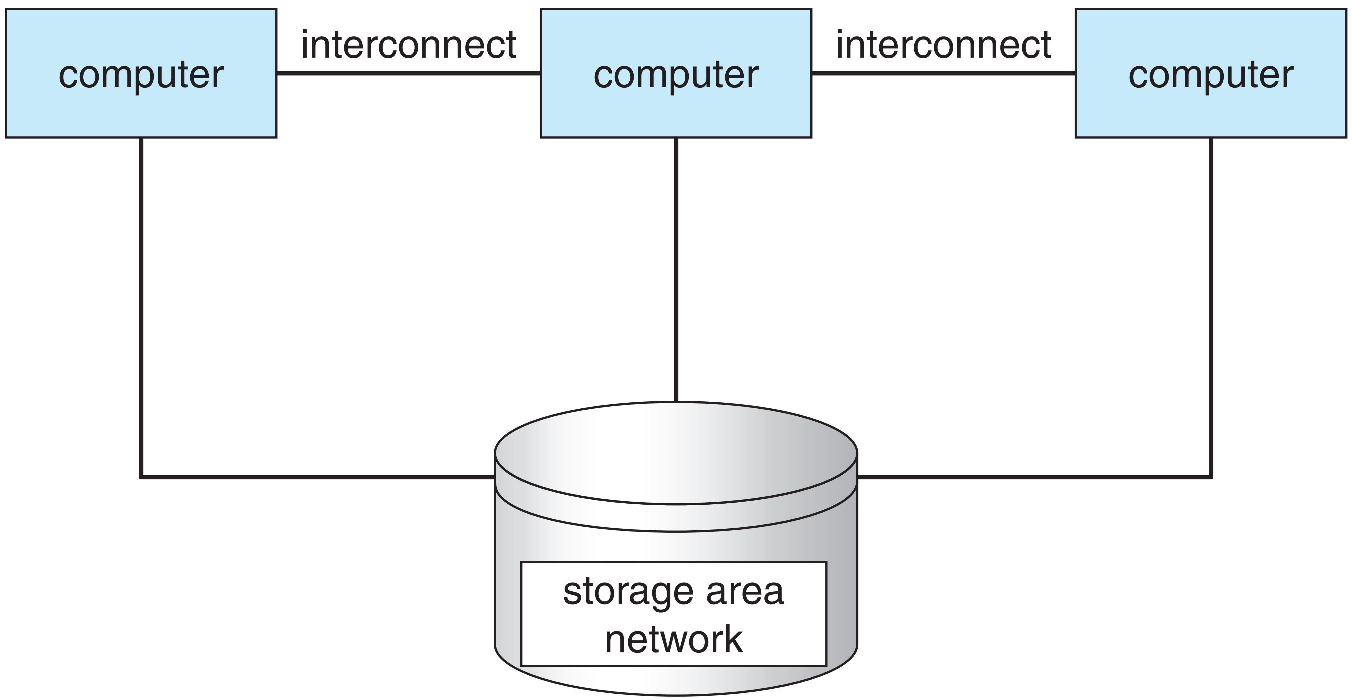
Operating System Definition

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a **storage-area network (SAN)**
  - Provides a **high-availability** service which survives failures
    - **Asymmetric clustering** has one machine in hot-standby mode
    - **Symmetric clustering** has multiple nodes running applications, monitoring each other
  - Some clusters are for **high-performance computing (HPC)**
    - Applications must be written to use **parallelization**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

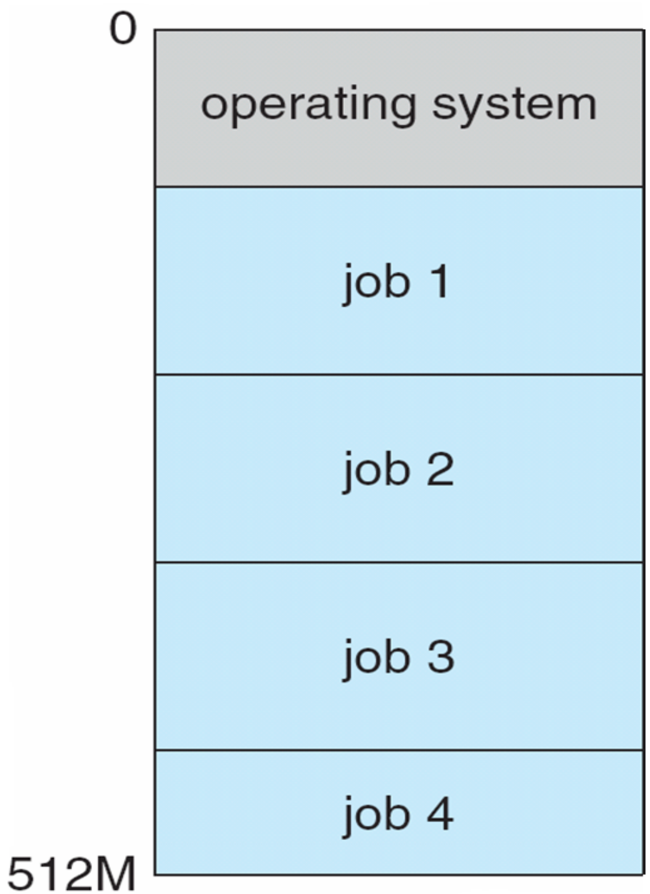
Clustered Systems



Operating System Structure

- **Multiprogramming** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via **job scheduling**
  - When it has to wait (for I/O for example), OS switches to another job
- **Timesharing (multitasking)** is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory ⇨**process**
  - If several jobs ready to run at the same time ⇨ **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory

Memory Layout for Multi-programmed System



Operating-System Operations

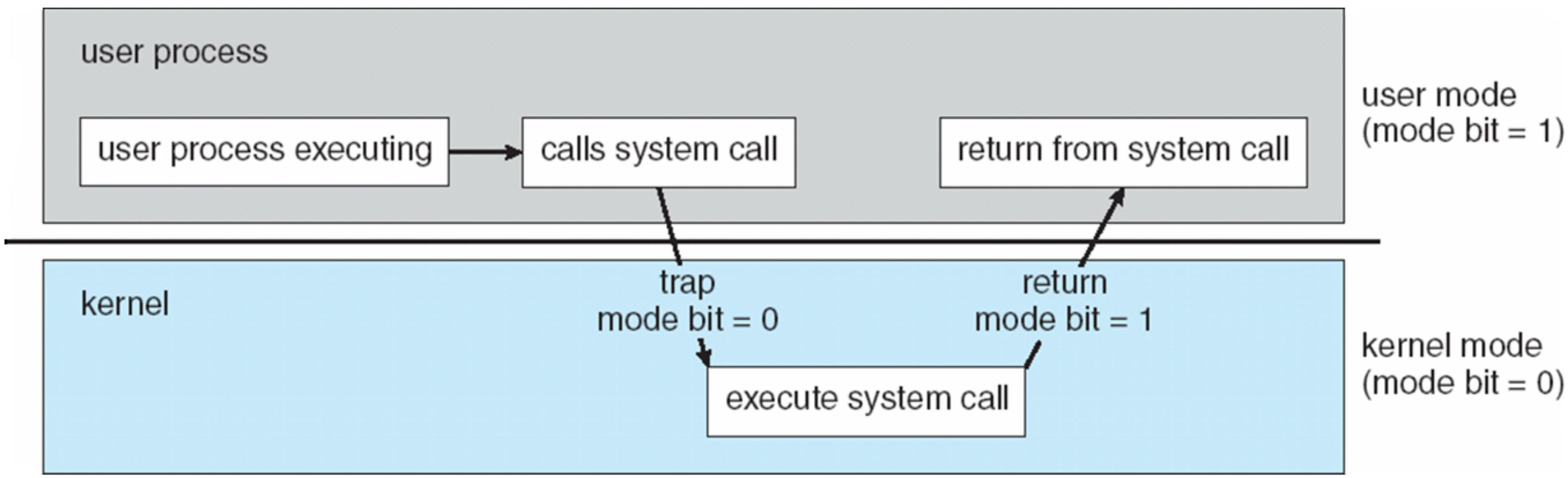
- **Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap):**
    - Request for operating system service
    - Software error (e.g., division by zero)
    - Other process problems include infinite loop, processes modifying each other or the operating system

Operating-System Operations (cont.)

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock.
  - Operating system set the counter (privileged instruction)
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a ***passive entity***, process is an ***active entity***.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

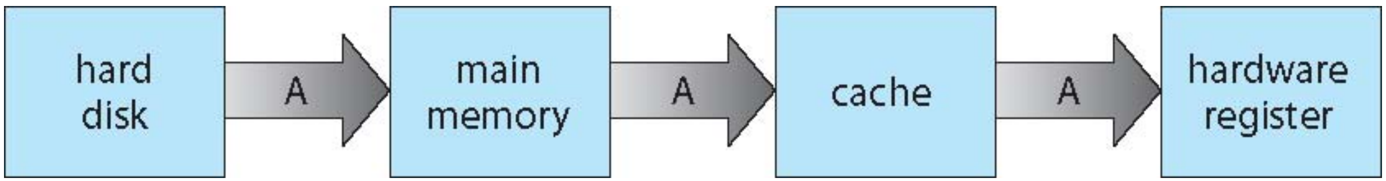
Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit

Migration of data "A" from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy

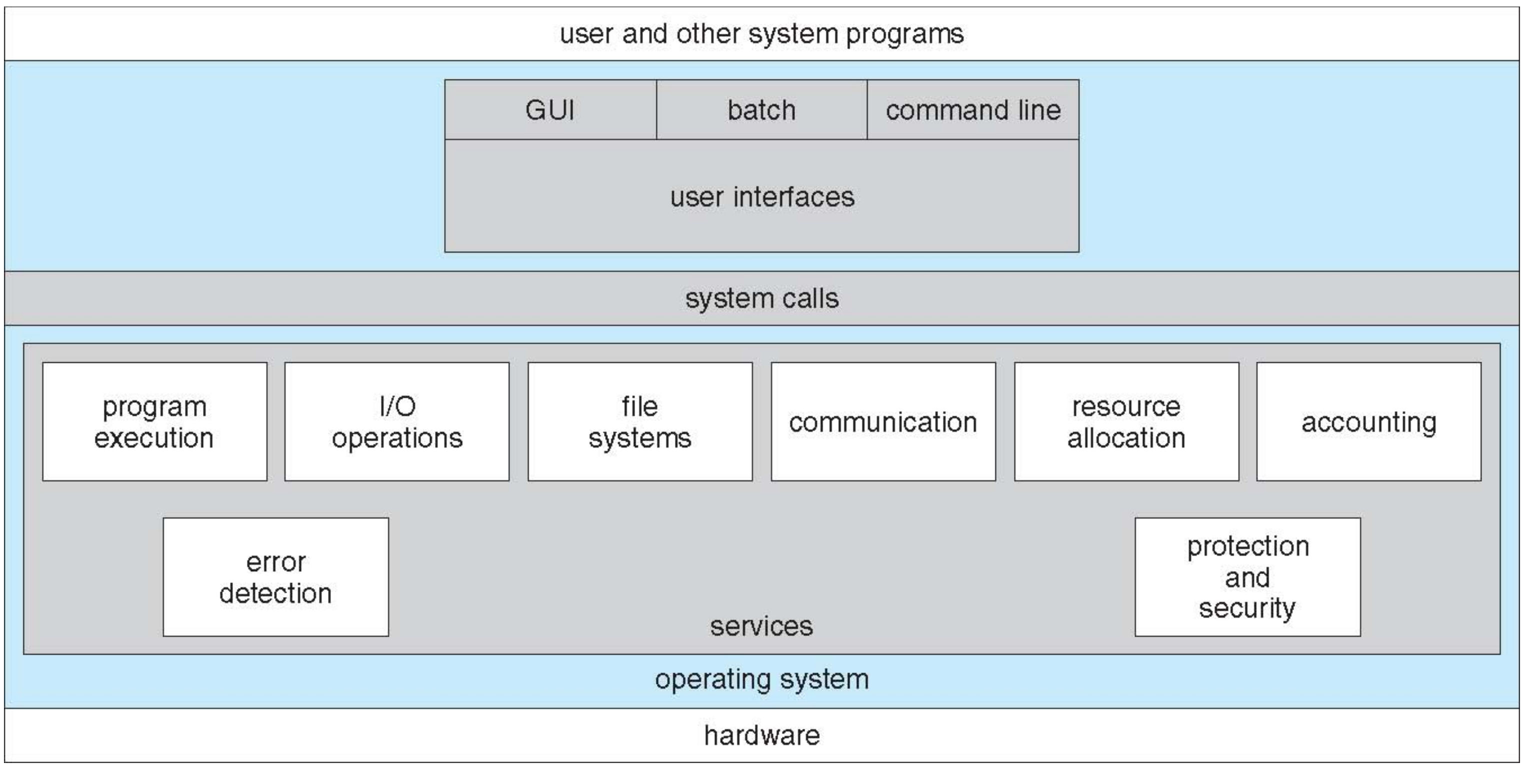


- Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recent value in their cache
- Distributed environment situation even more complex
  - Several copies of a datum can exist
  - Various solutions covered in Chapter 17

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices

A View of Operating System Services



I/O Structure

- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- After I/O starts, control returns to user program without waiting for I/O completion
  - **System call** – request to the operating system to allow user to wait for I/O completion
  - **Device-status table** contains entry for each I/O device indicating its type, address, and state
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt

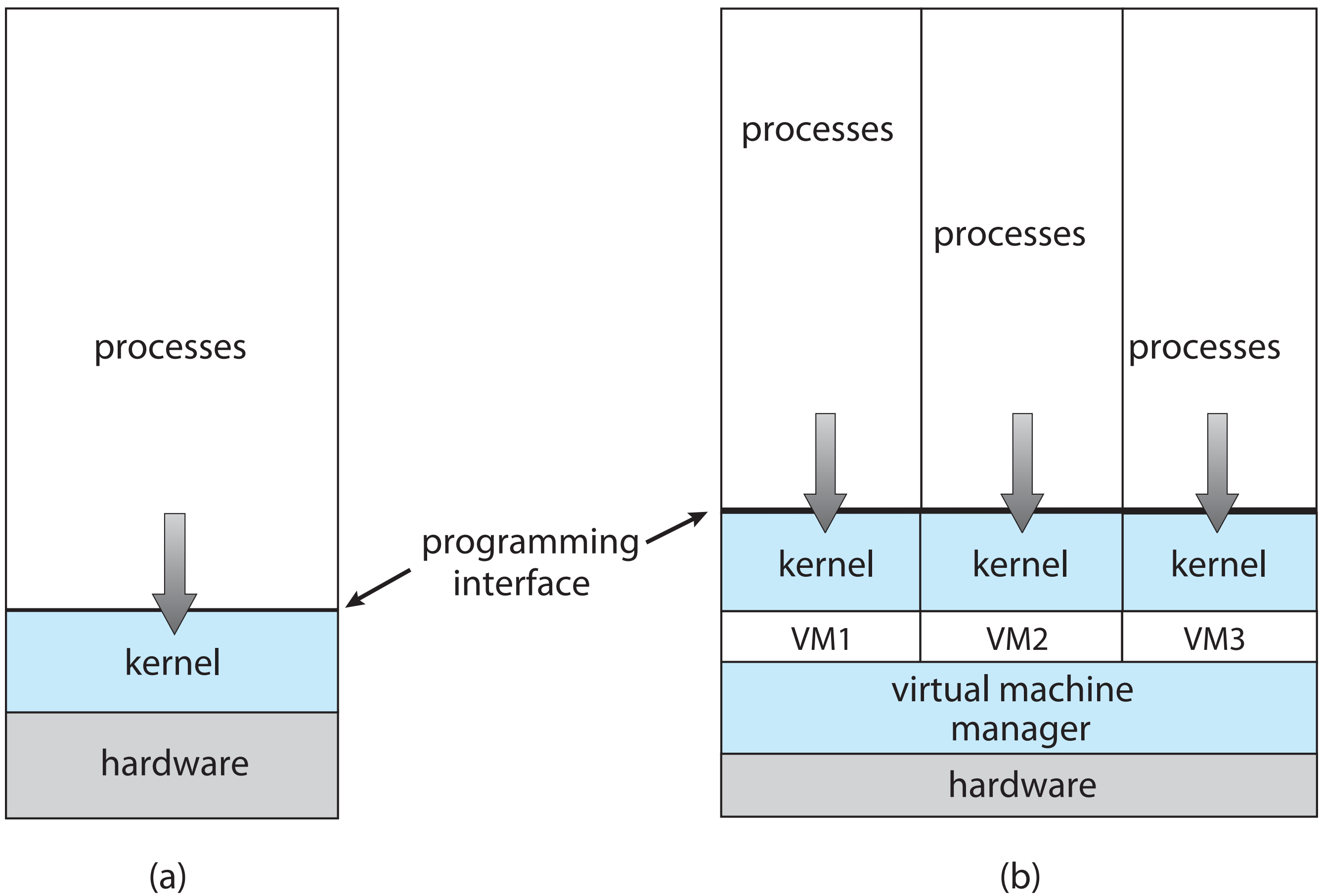
Computing Environments - Virtualization

- Allows operating systems to run applications within other OSes
  - Vast and growing industry
- **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
  - Generally slowest method
  - When computer language not compiled to native code – **Interpretation**
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - **VMM** (virtual machine Manager) provides virtualization services

Computing Environments - Virtualization

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - Apple laptop running Mac OS X host, Windows as a guest
  - Developing apps for multiple OSes without having multiple systems
  - QA testing applications without having multiple systems
  - Executing and managing compute environments within data centers
- VMM can run natively, in which case they are also the host
  - There is no general purpose host then (VMware ESX and Citrix XenServer)

Computing Environments - Virtualization



Origins of UNIX

- **1962,** CTSS: MIT project
  - **CTSS = Compatible Time-Sharing System**
  - implemented on IBM 7090
  - lead by Corbato, extremely successful, and continued to be used as late as 1972.

CTSS, Fernando Corbató, MIT

- **1965,** Multics: Bell + GE + MIT's join project
  - Multics = MULTiplexed Information and Computing Service
  - **Ken Thompson**, one of PIs; Corbato too.
  - a natural extension of CTSS
  - Bell Labs withdrew from the project in 1969
  - System was used until 2000

Origins of UNIX

1970

- a simple file system on PDP-7 to support a Space Travel game at Bell Lab

  → **System V file system (s5fs)**

- added a process subsystem, a simple command interpreter (shell, late as Bourne shell) and a small set of utilities.

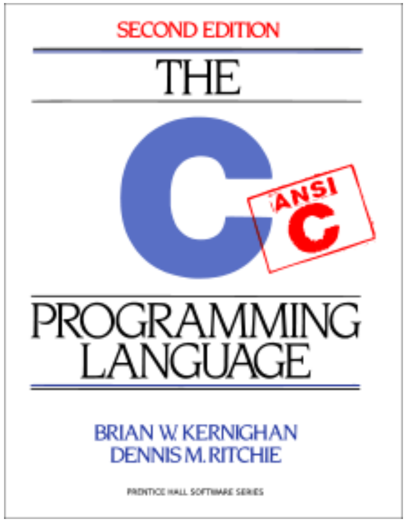
  → a new system **UNIX (1970)**

  UNIX=UNiplexed Information and Computing Service,
  or a pun on Multics

- ported to PDP-11
- added the ed editor
- added an interpretive language B
- **Dennis Richie** evolved B into **the C language** and cc

Origins of UNIX

- **1973**
  - UNIX Version 3 written in assembly language (2/73)
  - UNIX Version 4 written in **C** (11/73)



- 1974
  - the landmark paper:

    Thompson and Ritchie "The UNIX Time Sharing System" in the Communication of ACM, July 1974.
  - ACM Turing Award 1984

Evolution of UNIX

- **1974**
  - **UC Berkeley** obtained the UNIX system and added:
    - an ex editor (later known as vi)
    - C shell
    - a Pascal compiler
- **1978, two branches**
  - Branch 1 --- UNIX Version 7, (1/79), truly portable
  - System III in 1982
  - System V Release 1 in 1983
  - System V Release 2 in 1984, SVR2
  - System V Release 3 in 1987, SVR3 with ipc
  - **System V Release 4** in 1989, **SVR4** joined by SUN (Solaris) mixed with SVR3, 4BSD, SunOS, XENIX, added threading

Evolution of UNIX

1978, two branches

- Branch 2 --- **Berkeley Software Distribution (BSD) 2BSD**
  - distributed applications and utilities
  - ported to VAX-11, The first UNIX on 32bit architecture
- involved into Berkeley's **3BSD** in 1979
  - 4G address space, but only 2M physical memory ==> virtual memory
- DARPA fund BSD as part of **TCP/IP project**
  - 4.0BSD in 1980 **paging**
  - 4.1BSD in 1981 **networking**
  - **4.2BSD** in 1983 **fast file system (ffs)**
  - 4.3BSD in 1986
  - **4.4BSD** in 1993 replace the original vm with a new version based on MACH, add **log-structured file system**
  - 1993 the Berkeley Computer Systems Research Group exhausted their last source of funding for their continuing work, and the group was dissolved.
  - Some of the Berkeley researchers start to create an inexpensive operating system that runs on PCs, NetBSD, **FreeBSD**, and **OpenBSD**

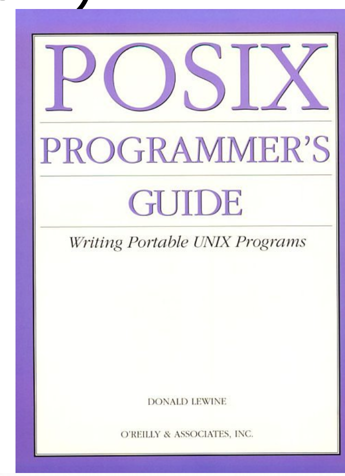
Evolution of UNIX

- **1980** Mach by CMU
  - Mach 3.0 --- microkernel implementation
- **1984 X window System**
  - developed at MIT
  - character-based → **graphical user interfaces (GUI)**

- allowed a user to run a program on a remote computer and display it in a window on the user's local computer
- X Consortium formed in 1988

Evolution of UNIX

- **1989** IBM + DEC + HP ==> Open Software Foundation
  - System V + 4.3BSD ==> **POSIX** (IEEE 1003.x)
  - (**P**ortable **O**perating **S**ystem **I**nterface)
- Commercial UNIX
  - XENIX on Intel 8086 by Microsoft (1978)
  - SunOS (4.2BSD), **Soloris (SVR4)** by Sun Microsystems with **NFS**,**vfs**
  - Irix by Silicon Graphics
  - AIX by IBM
  - HP/UX by HP
  - Ultrix,OSF/1 by DEC with multiprocessor support



IEEE

GNU & Free Software

FREE SOFTWARE FOUNDATION

- **1984 GNU**
  - "GNU's Not Unix"
  - GNU OS:

GNU's design is Unix-like, but differs from Unix by being free software and containing no Unix code

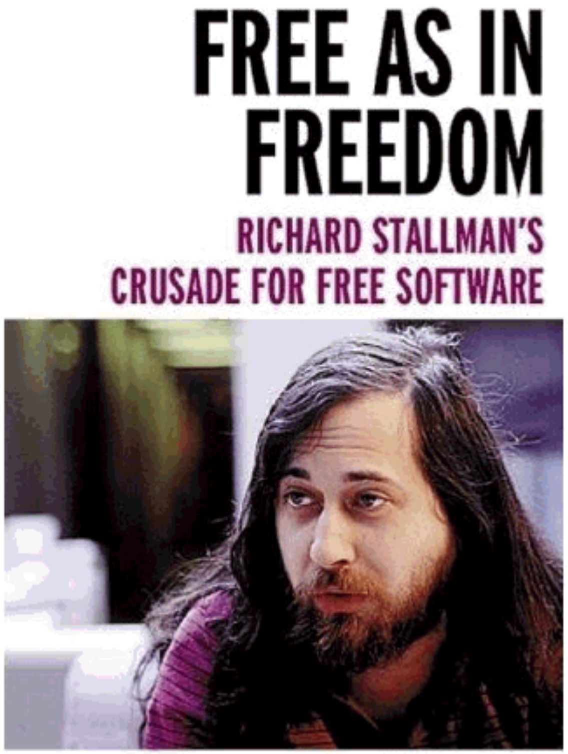
- **1985** FSF

*"**Free software**" means software that respects users' freedom and community. Roughly, **the users have the freedom to run, copy, distribute, study, change and improve the software**. With these freedoms, the users (both individually and collectively) control the program and what it does for them*

- Free Software Foundation (FSF) founded by **Richard Stallman**.
  - Initially to raise funds to help develop GNU
  - Create the tools needed to make a UNIX compatible OS

GNU & Free Software

- **GNU & FSF**
  - “Free” in “free software” pertains to freedom, not price.

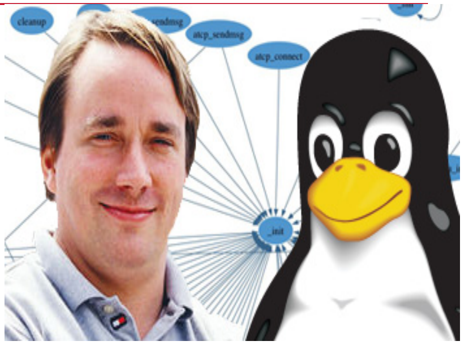


4 specific freedoms in using software:

1) To run the program as you wish
2) To copy the program and give it away to your friends and co-workers;
3) To change the program as you wish, by having full access to source code;
4) To distribute an improved version and thus help build the community.

GNU General Public License (GNU GPL or simply GPL) is the widely used free software license

UNIX/LINUX



1991 LINUX

- started by **Linus Torvalds** in 1991, a student at the University of Helsinki in Finland.
  - Built on the concepts in MINIX, OS textbook by Tannenbaum
  - Respond to Richard Stallman GNU software calling for a free kernel.
- GNU/Linux: a FREE OS, compatible with the POSIX 1003.1 standard, including the functions of UNIX System V and BSD 4.3
- Linux Kernel combined with GNU utilities and libraries

UNIX/LINUX

- LINUX Distributions --- GNU/Linux bundled with other applications
  - **Red Hat** (Enterprise) Linux : the largest commercial Linux vendor. **CentOS**, a free version of Red Hat
  - **Fedora (**Red Hat sponsors a community version) which undergoes a more rapid development cycle.
  - **Debian** & KNOPPIX – The first Live CD version. KNOPPIX is based on Debian Linux. Many vendors now offer Live versions, no installation.
  - **Ubuntu**: most popular desktop version.

UNIX/LINUX

- **1994** Linux 1.0 release
  - TCP/IP support
  - VM subsystem
  - a complete file system
- **2011** Linux 3.0 release
  - As of 2012, the Linux 3.2 release had about 15M lines of code
- **2015** Linux 4.0 release

- **1996** Linux 2.0 release
  - support for multiprocessor architecture
  - support for heterogeneous architectures
    - 64-bit DEC Alpha
    - Motorola 68000-series
    - SUN's SPARC
    - PowerMac

Open source vs free software

- Is Free Software too strong?
  - Replaced by **Open Source Software (OSS)** , 1998

    less ambiguous and more comfortable for the corporate world

- Publish software with an **open source license**,
  - anybody may understand its internal functioning.
  - anyone is allowed to create its **modifications**, **port** it to new OSs and architectures

open source™

nearly all free software is open source, and
nearly all open source software is free. --- by FSF

Open source vs free software

- **Typical Open Source Software**
  - Apache-- the most popular HTTP server software
  - The GIMP - Adobe Photoshop clone
  - Ghostscript - Postscript interpreter
  - Open Office - Office suite
  - Mozilla Firefox - Web browser
  - Mozilla - Thunderbird E-mail Client
  - VLC - Media player
  - MySQL – Database

http://www.digitaltrends.com/

References: UNIX/LINUX history

- http://cm.bell-labs.com/cm/cs/who/dmr/hist.html, Dennis M. Ritchie, "The evolution of the Unix time-sharing system," AT&T Bell Laboratories Technical Journal, vol. 63 No.6, Part 2, pp. 1577-1593, Oct 1984,
- http://www.levenez.com/unix/ “UNIX Timeline”

Microsoft and Albuquerque

- The January 1975 issue of *Popular Electronics* featured an article on **Altair 8800**, the 1st PC, for sale to public.

MITS® Micro Instrumentation & Telemetry Systems, Inc.
5404 Coal Ave., S. E., Albuquerque, New Mexico 87108

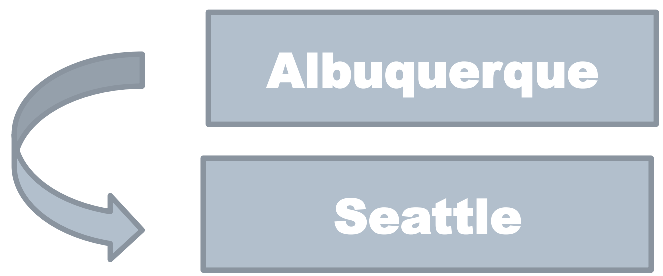
- the Altair 8800 was made by MITS (Micro Instrumentation and Telemetry System), 5404 Coal Ave, SE, within 1 mile of UNM!
- **Bill Gates and Paul Allen** saw the magazine and called to say they had a program ready for Altair 8800.
- Gates worked the next 8 weeks non-stop at the Harvard computer center to complete a BASIC programming language interpreter for Altair 8800.

Microsoft and Albuquerque

- April 1975, Gates and Allen formed the business partnership, called **Micro-Soft**, registered with the state of New Mexico.

They licensed their program to MITS' owner, Ed Roberts, and the program was named as ***MS-BASIC***.

- In 1979, Gates and Allen decided to move Micro-Soft with a dozen of employees from Albuquerque to Seattle, and change their name from Micro-Soft to **Microsoft**.



Microsoft → MS-DOS, MS Windows

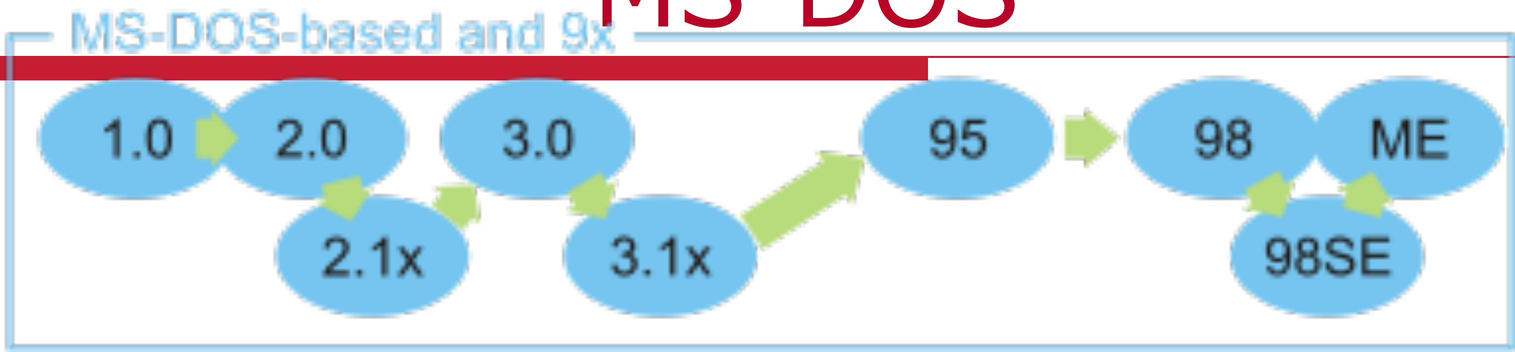
- In **1980**, IBM started IBM PC, and wanted a product (model 5150) ready in a short time.

Then, Microsoft was approached by IBM.

- In **1981**, Gates licensed **MS-DOS (Microsoft Disk Operating System)** to IBM.

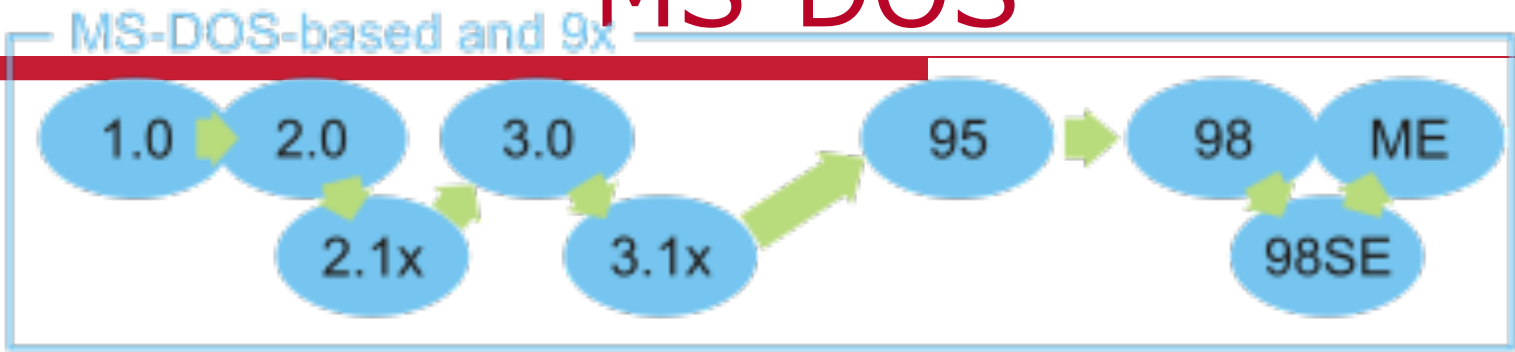
- In November **1983**, Microsoft Windows 1.0 was launched.
- Windows 2.0 is designed for the Intel 286 processor, by **1987**

MS-DOS



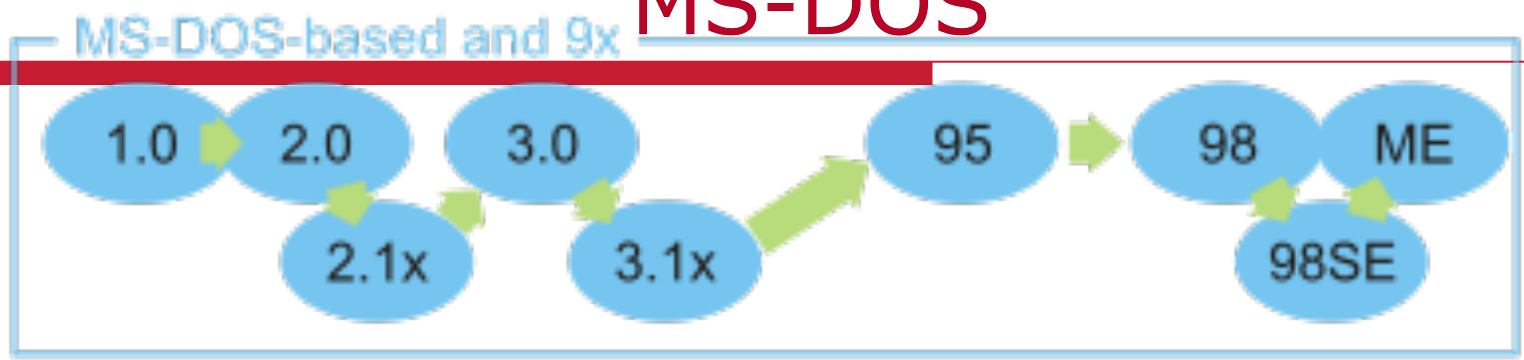
- MS-DOS
  - Time frame: the 1980s to the mid 1990s
  - Platform: x86 –based personal computers
- MS-DOS 1.0, 1981, on Intel 8086
  - 16-bit, **single-user, command-line oriented**
  - 4K lines code (assembly)
  - memory requirement: 8KBytes in memory OS

MS-DOS



- MS-DOS
  - Time frame: the 1980s to the mid 1990s
  - Platform: x86 –based personal computers
- MS-DOS 2.0, 1983, on Intel 286
  - 16-bit, 24-bit segmented
  - memory requirement: 24KBytes in memory OS
  - **MMU, virtual memory**, 30-bit address space
  - tree-structure file/directory system

MS-DOS



- MS-DOS
  - Time frame: the 1980s to the mid 1990s
  - Platform: x86 –based personal computers
- MS-DOS 3.0, 1984, on Intel 80286
  - 16-bit, 24-bit segmented
  - memory requirement: 36KBytes in memory OS
  - **Networking**

Windows to Windows NT

MICROSOFT® WINDOWS

- Time frame: the 1985 to 2001
- A front-end to the MS-DOS on Intel 286/386
  - 16-bit graphical operating environment
  - Limited multitasking of existing MS-DOS programs
  - Included original device drivers for I/O

Windows 1.0, 1985
Windows 2.0, 1987
Windows 3.0, 1990

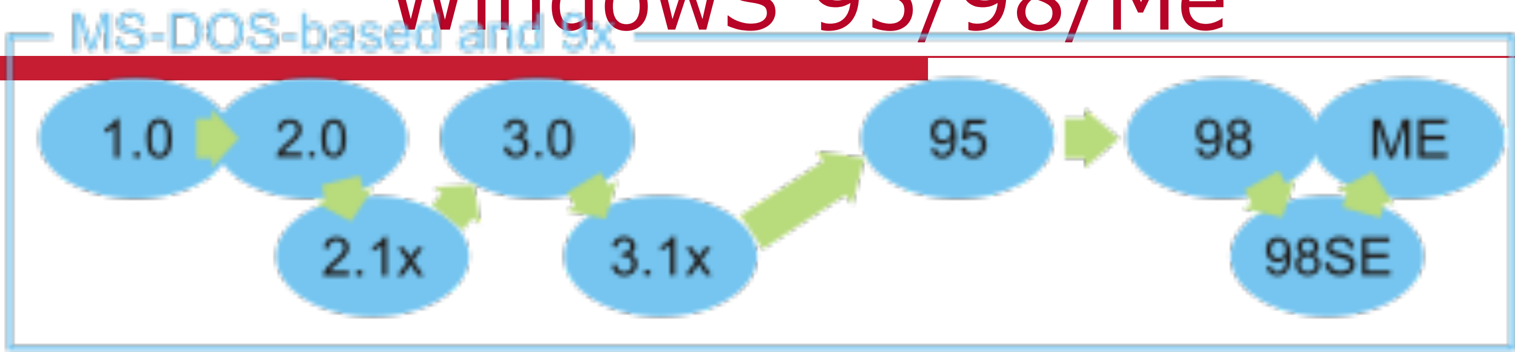
Windows to Windows NT

MICROSOFT® WINDOWS

16-bit → 32-bit
Building a modern 32-bit OS on top of the 16-bit MS-DOS was not working!

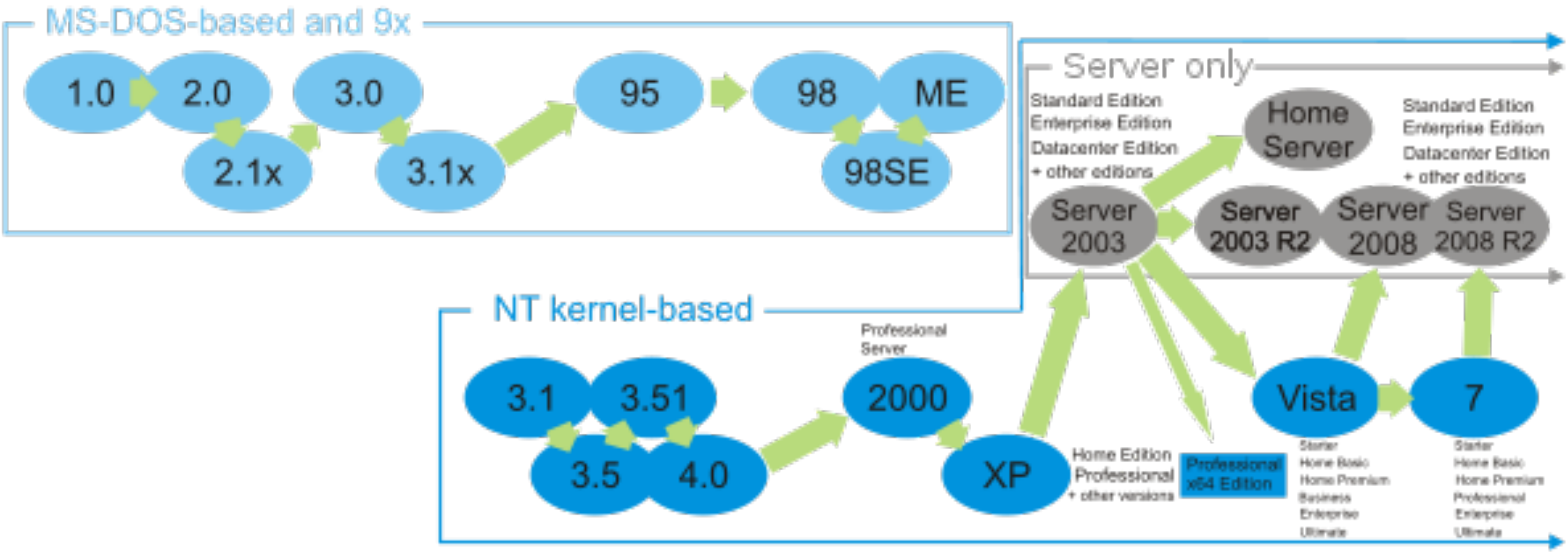
- Windows **3.1**, 1993, on Pentium
  - Not-truly 32-bit OS, still on 16-bit MS-DOS
  - mainly in desktop market
- Windows NT 3.1 (**Windows New Technolog**y)
  - real 32-bit OS, **no MS-DOS**
  - written mainly in C, initial release had 3M Lines of code.
  - mainly in server market
- Windows NT 4.0 in 1996
  - written mainly in C and C++, 16M Lines of code.
  - highly portable: Pentium. Alpha, MIPS, PowerPC, ...

WindowS 95/98/Me



- Windows 95, on MS-DOS 7.0
  - 16-bit assembly code and 32-bit code mixed together
- Windows 98
  - migrate more functionalities from MS-DOS to the Windows part.
  - although multiprogramming, **kernel itself was not reentrant**
  - with a 4-GB virtual address space and due to MS-DOS features:
    - 2 GB was private to the process
    - 2 GB was read-write shared among all processes as well as kernel structures. so a small bug in one program could wipe out the system.
- Windows Me (Windows Millennium Edition)
  - still a minor revision to Windows 98

Merge of Windows NT & Windows 9x



- Windows 2000 = Windows NT 5.0
  - Merge of two branches
    - MS-DOS based and Windows 9x
    - NT kernel-based

Merge of Windows NT & Windows 9x

- Windows 2000 (Windows NT 5.0a)
  - OS runs in kernel mode, user processes run in user mode.
  - support for running on symmetric multiprocessors (SMPs, <=32)

- No MS-DOS, instead cmd.exe is used.
- support Unicode, single binary to run everywhere in the world with different languages.
- come with several levels:
  - Professional --- 4GB RAM, 2 CPUs, optimized for response time
  - Sever --- 4GB, 4 CPUs, optimized for throughput
  - Advanced server --- 8GB, 8 CPUs, in a cluster of 2
  - Datacenter server --- 64GB, 32 CPUs, in a cluster of 4

Windows XP/Vista/7/8/10

- Windows XP, 2001 & Windows Server 2003
  - The name "XP" is short for "eXPerience", highlighting the enhanced user experience
  - built on the Windows NT kernel
- Windows VISTA and Windows Server 2008
  - Emphasizing Trustworthy Computing

- Windows 7 (2009)
- Window 8 (2012)
- Window 10 (2016)
- Moving towards 64-bit
- Moving towards tablets & mobile