CS490 Windows Internals

Kenny Q. Zhu

Dept. of Computer Science

Shanghai Jiao Tong University

Kenny Zhu

Research Interests:

Programming Languages
- Data processing
- Concurrent programming

Database & Data mining
- Information extraction
- Knowledge discovery

Degrees: *National University of Singapore (NUS)*
Postdoc: *Princeton University*
Experiences: *Microsoft Redmond, USA*
Microsoft Research Asia
Faculty at *SJTU* since 2009

Administrative Info (I)

- Lecturer:
  - Kenny Zhu, SEIEE #03-524, kzhu@cs.sjtu.edu.cn
  - Office hours: by appointment or after class
- Teaching Assistant:
  - Jack Sun, SEIEE #03-341, jacksunwei@gmail.com
  - Office hours: Monday 4-6 PM
- Textbook: Windows Internals (5th ed.) By Mark E. Russinovich, *et al.*
- Course Web Page (definitive source!): http://www.cs.sjtu.edu.cn/~kzhu/cs490/

Administrative Info (II)

- All-English Course: everything in English!
- Lectures:
  - Course material + lab demo + quizzes
  - Lecture slides on course web site after class
- Assignments and Labs:
  - Released (usually) on Fridays
  - Assignments due on the following Friday
  - Submit *hard copies* to class or Jack's office SEIEE #03-342
  - Late submission: -30% of full score for each additional day
- Group Projects:
  - 2 persons per group

Administrative Info (III)

- 3-credit course
- Modes of Assessment:
  - Quizzes: 20%
  - Assignments: 40%
  - 2 Group Projects: 40%
- Email Jack the names and contact of your group by **this weekend**!

Course Overview

- Concepts and Tools
- Windows Structuring (Architecture)
- Core system mechanisms
- Concurrency and Windows Traps (interrupts/exceptions)
- Windows Synchronization
- Processes and thread, Scheduling
- Memory management
- I/O management
- File system
- Windows security (Pending)

Copyright Notice

- Much of the materials in this course are part of the *Windows Operating System Internals Curriculum Development Kit,* developed by David A. Solomon and Mark E. Russinovich with Andreas Polze
- Microsoft has licensed these materials from David Solomon Expert Seminars, Inc. for distribution to academic organizations solely for use in academic environments (and not for commercial use)

Roadmap For This Lecture

- Overview of Operating Systems
- Main Concepts in Operating Systems
- Structure of Operating Systems
- Micro-kernels
- History of Windows

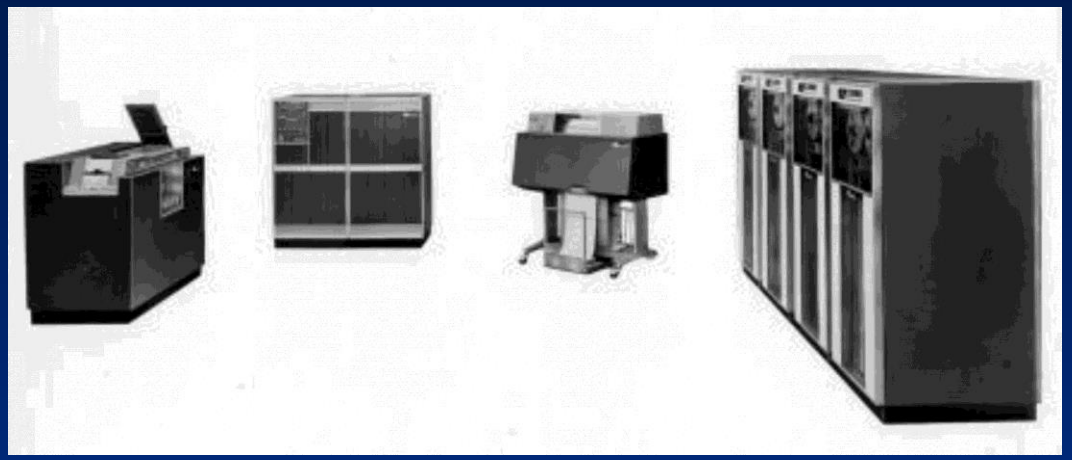
Operating Systems Concepts

- OS is *intermediary* between user and computer hardware
  - Ease of use (for the user)
  - Efficiency (for the hardware)
- Layered architectures

History of operating systems

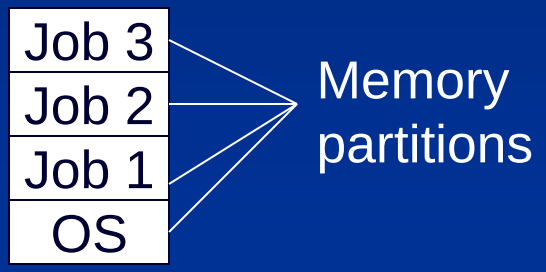
- Batch processing

The elements of the basic IBM 1401 system are the 1401 Processing Unit, 1402 Card Read-Punch, and 1403 Printer.



- Punching cards programming

Multiprocessing



The Evolution of Operating System Functionality

- Batch Job Processing
  - Linkage of library routines to programs
  - Management of files, I/O devices, secondary storage
- Multiprogramming
  - Resource management and sharing for multiple programs: spooling
  - Quasi-simultaneous program execution
  - Single user
- Multiuser/Timesharing Systems
  - Management of multiple simultaneous users interconnected via terminals
  - Fair resource management: CPU scheduling, spooling, mutual exclusion
- Real-Time Systems (process control systems)
  - Management of time-critical processes
  - High requirements with respect to reliability and availability

Tasks of an Operating System

- Processor management - Scheduling
  - Fairness
  - Non-blocking behavior
  - Priorities
- Memory management
  - Virtual versus physical memory, memory hierarchy
  - Protection of competing/concurrent programs
- Storage management – File system
  - Access to external storage media
- Device management
  - Hiding of hardware dependencies
  - Management of concurrent accesses
- Batch processing
  - Definition of an execution order; throughput maximization

Kernel- and User Mode Programs

Typical functionality implemented in either mode:

Kernel Space:

- Privileged mode
- Strict assumptions about reliability/security of code
- Memory resident
  - CPU-, memory-, Input/Output managment
  - Multiprocessor management, diagnosis, test
  - Parts of file system and of the networking interface

User Space:

- More flexible
- Simpler maintenance and debugging
  - Compiler, assembler, interpreter, linker/loader
  - File system management, telecommunication, network management
  - Editors, spreadsheets, user applications

Layered Model of Operating System Concepts

nr	name	typical objects	typical operations
1	Integrated circuits	register, gate, bus	Nand, Nor, Exor
2	Machine language	instruction counter, ALU	Add, Move, Load, Store
3	Subroutine linkage	procedure block	Stack Call, JSR, RTS
4	Interrupts	interrupt handlers	Bus error, Reset
5	Simple processes	process, semaphore	wait, ready, execute
6	Local memory	data block, I/O channel	read, write, open, close
7	Virtual model	page, frame	read, write, swap
8	Process communication	channel (pipe), message	read, write, open
9	File management	files	read, write, open, copy
10	Device management	ext.memory, terminals	read, write
11	I/O data streams	data streams	open, close, read, write
12	User processes	user processes	login, logout, fork
13	Directory management	internal tables	create, delete, modify
14	Graphical user interface	window, menu, icon	OS system calls

Layered Model of Operating System Concepts

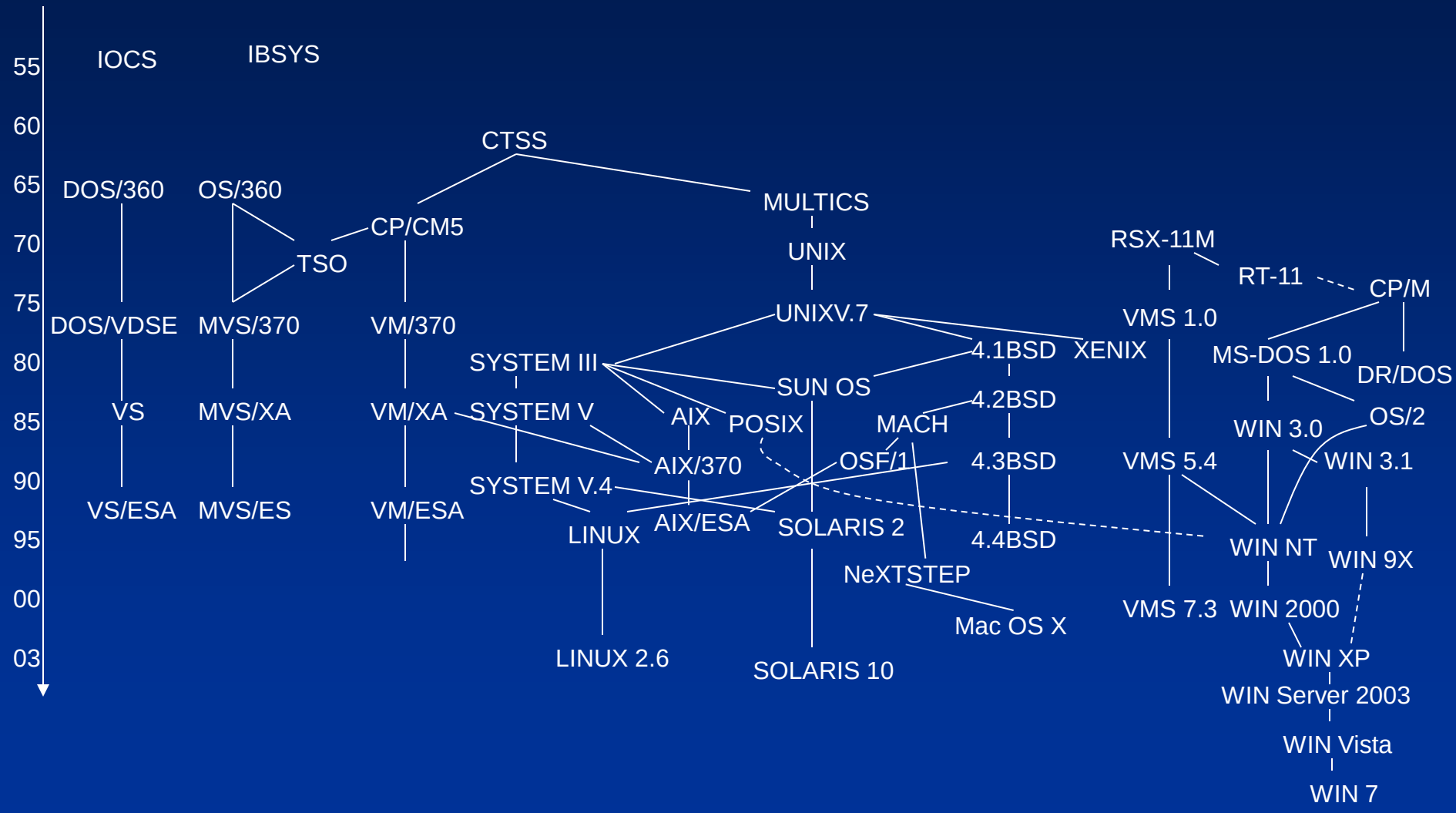
nr	name	typical objects	typical operations
1	Integrated circuits	register, gate, bus	Nand, Nor, Exor
2	Machine language	instruction counter, ALU	Add, Move, Load, Store
3	Subroutine linkage	procedure block	Stack Call, JSR, RTS
4	**Interrupts**	**interrupt handlers**	**Bus error, Reset**
5	Simple processes	process, semaphore	wait, ready, execute
6	**Local memory**	**data block, I/O channel**	**read, write, open, close**
7	**Virtual model**	**page, frame**	**read, write, swap**
8	**Process communication**	**channel (pipe), message**	**read, write, open**
9	**File management**	**files**	**read, write, open, copy**
10	Device management	ext.memory, terminals	read, write
11	I/O data streams	data streams	open, close, read, write
12	**User processes**	**user processes**	**login, logout, fork**
13	**Directory management**	**internal tables**	**create, delete, modify**
14	Graphical user interface	window, menu, icon	OS system calls

OS acts as Extension of Hardware

- System view: layered model of OS
  - Implementation details on one layer are hidden from higher layers
- Same machine, different operating systems:
  - IBM PC: DOS, Linux, NeXTSTEP, Windows, SCO Unix
  - DEC VAX: VMS, Ultrix-32, 4.3 BSD UNIX
- Same OS, different machines: UNIX
  - PC (XENIX 286, APPLE A/UX)
  - Mac (OS X)
  - CRAY-Y/MP (UNICOS - AT&T Sys V)
  - IBM 360/370 (Amdahl UNIX UTS/580, IBM UNIX AIX/ESA)
  - Windows NT, XP, 2000, 2003, ...
    - Intel x86, Alpha, PowerPC, MIPS, Itanium

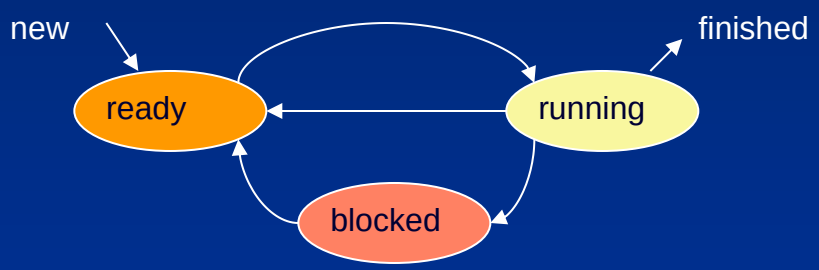
Operating Systems Evolution

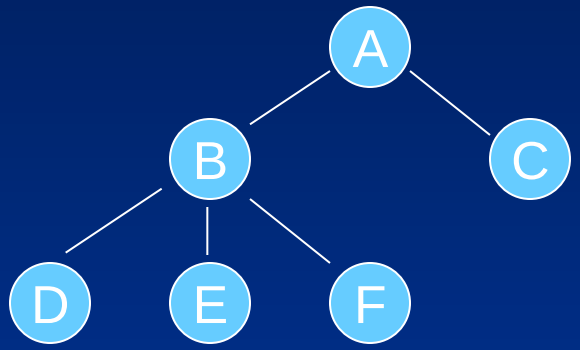
(more at http://www.levnez.com)



Main Concepts: processes

- Processes is runtime context of a program
- Process table, core image
- Command interpreter, shell
- Child processes



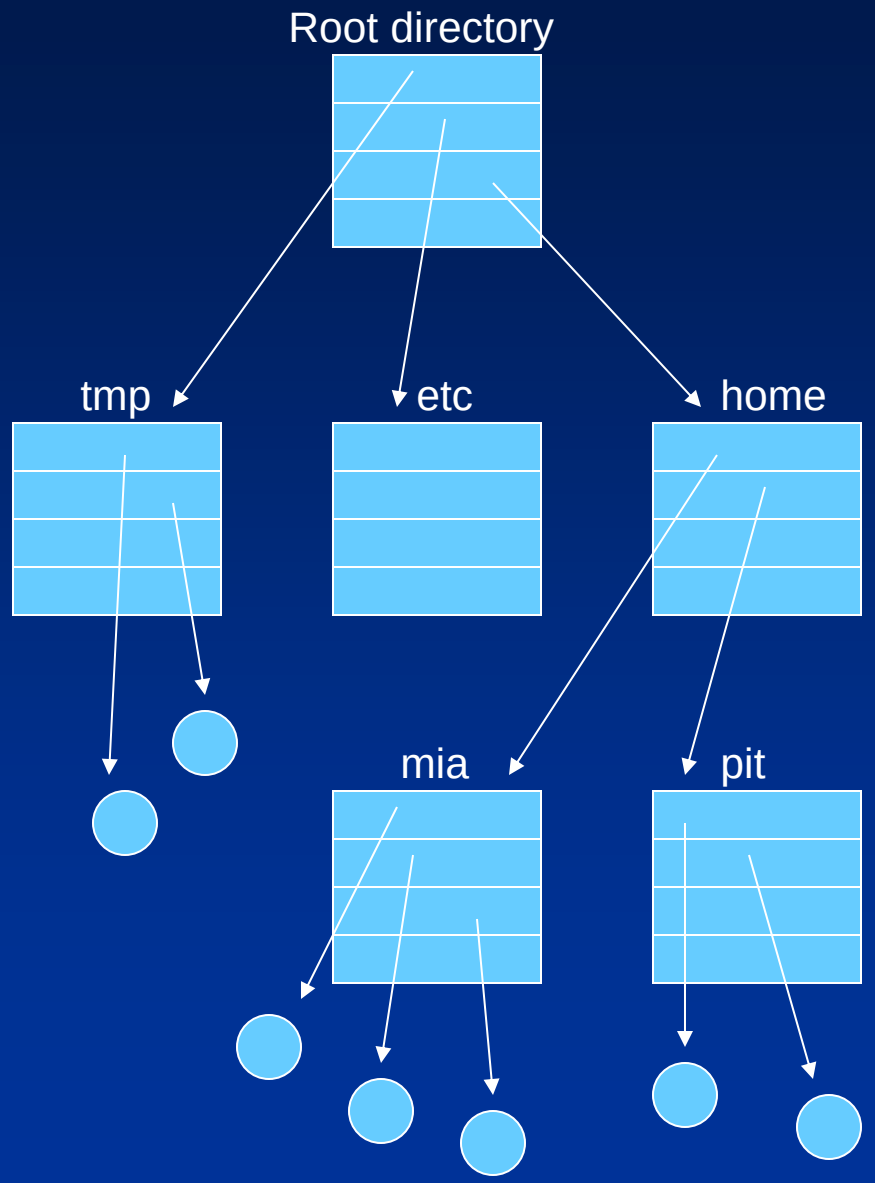


Process tree

- Scheduling, signals
- User identification, group identification

Main Concepts: Files

- Files, directories, root
- Path, working directory
- Protection, rwx bits
- File descriptor, handle
- Special files, I/O devices
- Block I/O, character I/O
- Standard input/output/error
- pipes



Main concepts: system calls

- User programs access operating system services via system calls
- Parameter transmission via trap, register, stack

  count=read(file, buffer, nbytes);

- 5 general classes of system calls:
  - Process control
  - File manipulation
  - Device manipulation
  - Information maintenance
  - Communications

Main concepts: shell

- Command interpreter
- Displays prompt, implements input/output redirection
- Background processes, job control, pseudo terminals

$ date

$ date >file

$ sort <file1 >file2

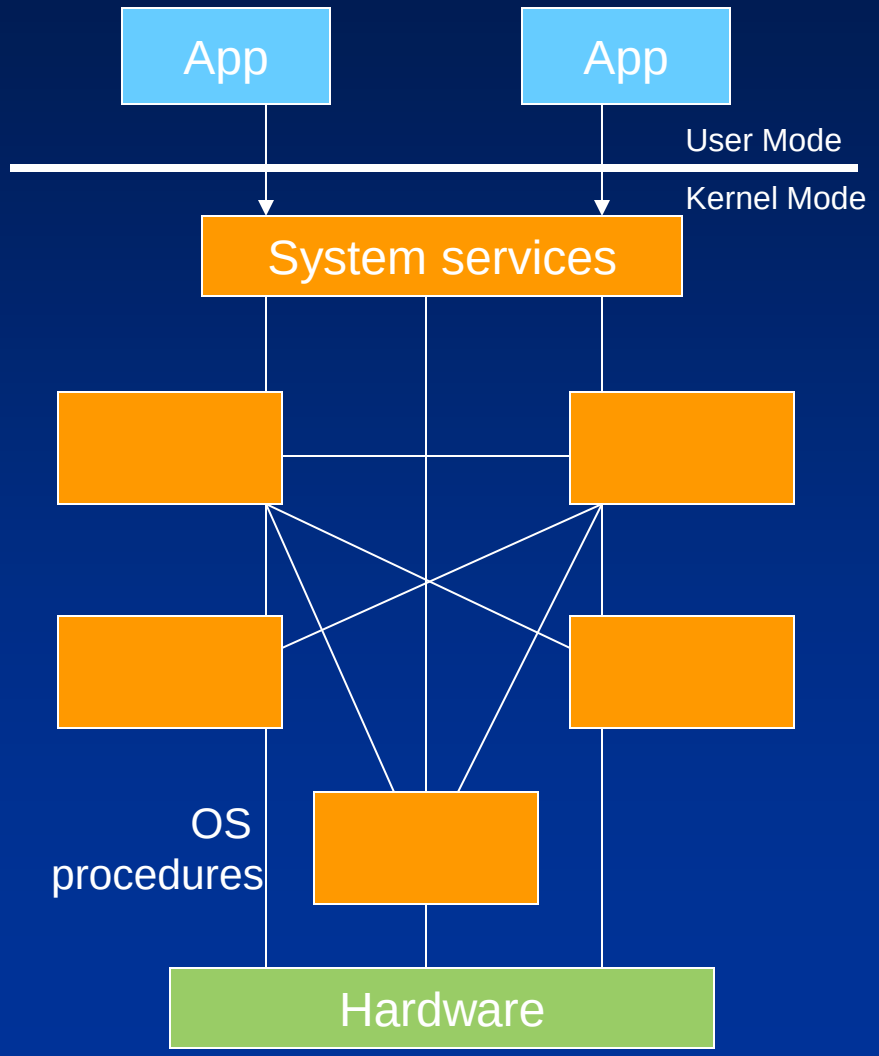
$ cat file1 file2 file3 > /dev/lp1

$ make all >log 2>&1 &

Note: 0: stdin, 1:stdout, 2:stderr

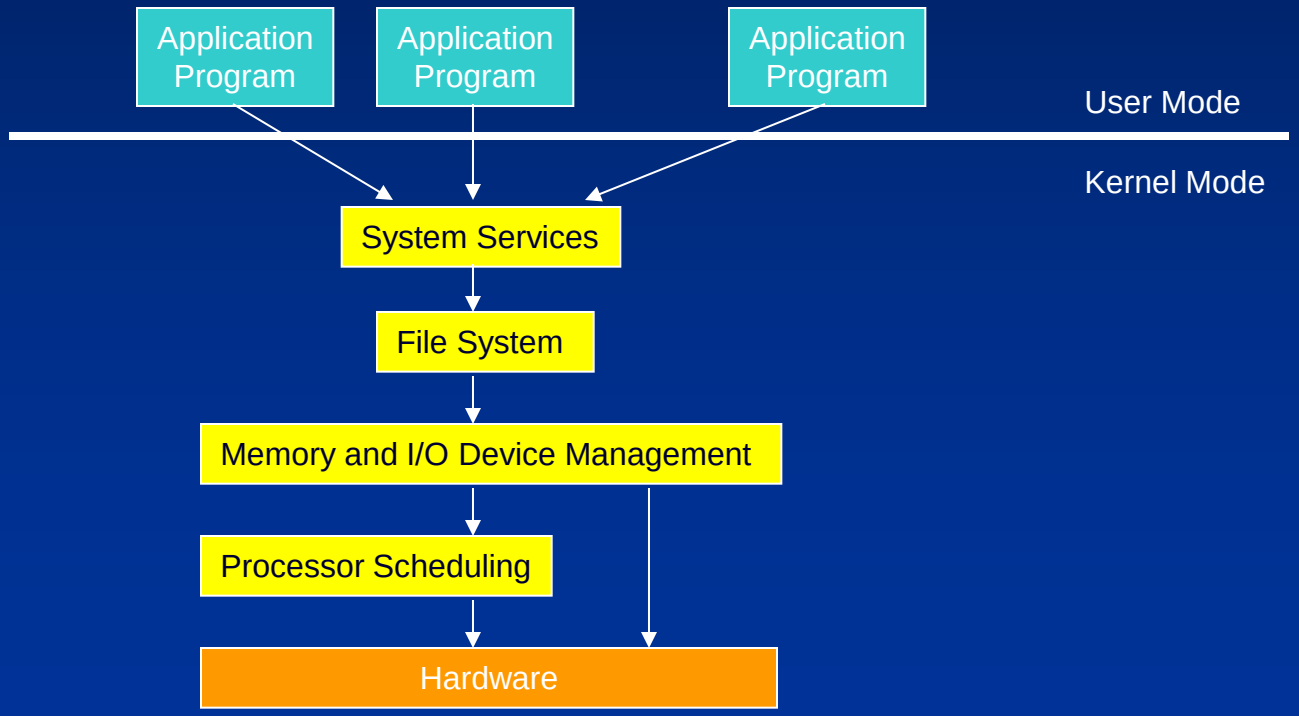
Structuring of Operating Systems

- **Monolithic systems**
  - Functionally *distinguishable* aspect not architecturally separate components, but all *interwoven*.
  - Unstructured
  - Supervisor/System call changes from user mode into kernel mode



Layered OS

- Each layer is given access only to lower-level interfaces

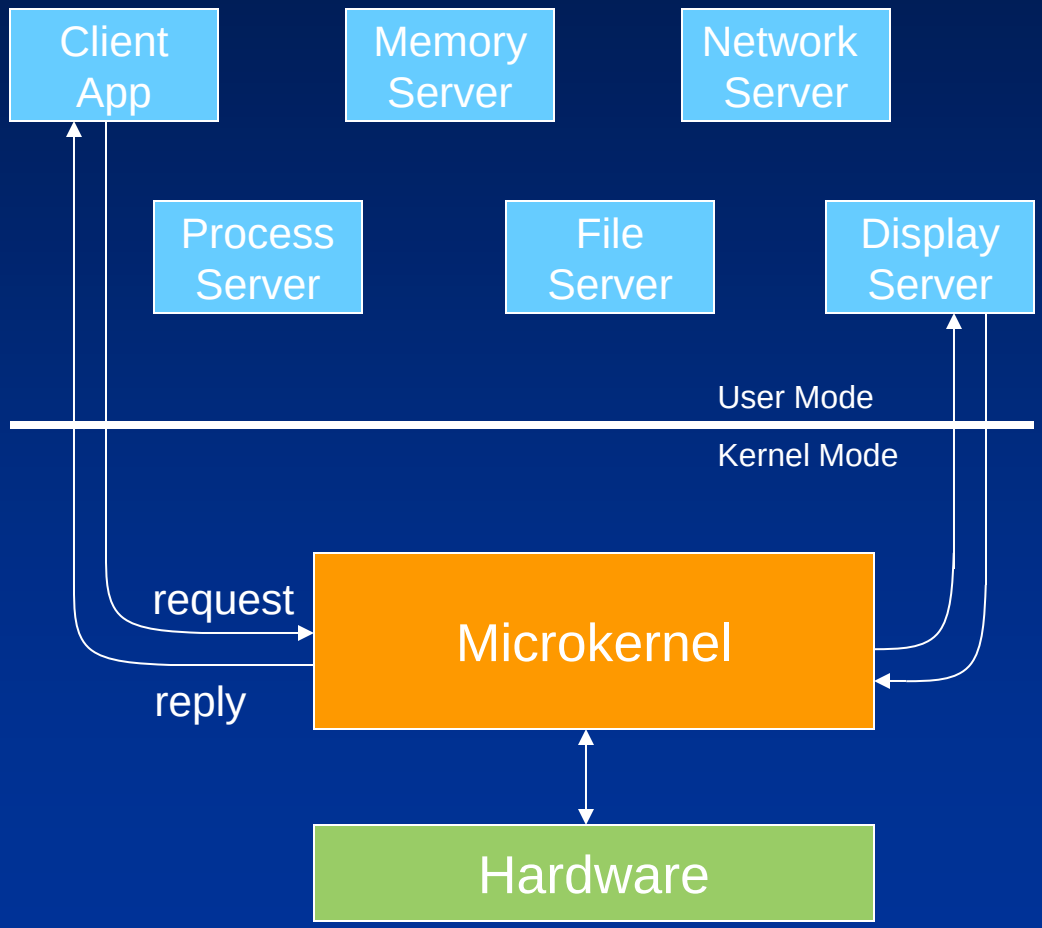


Microkernel OS (Client/server OS)

Kernel implements:

- Scheduling
- Memory Management
- Interprocess communication (IPC)

User-mode servers



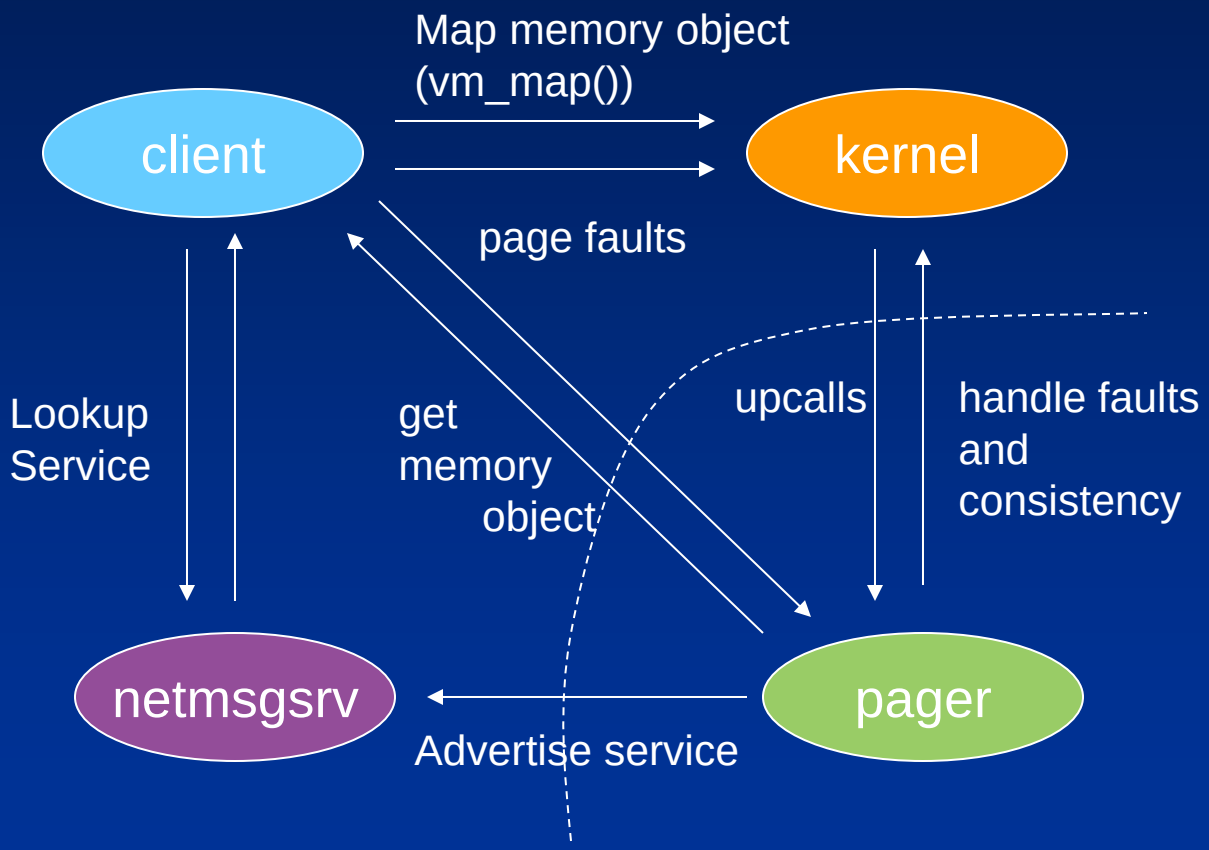
Microkernel OS?

- Is Windows a microkernel-based OS?
  - No – not using the academic definition (OS components and drivers run in their own private address spaces, layered on a primitive microkernel)
  - All kernel components live in a common shared address space
    - Therefore no protection between OS and drivers
- Why not pure microkernel?
  - Performance – separate address spaces would mean context switching to call basic OS services
  - Most other commercial OSs (Unix, Linux, VMS etc.) have the same design
- But it does have some attributes of a microkernel OS
  - OS personalities running in user space as separate processes
  - Kernel-mode components don't reach into one another's data structures
    - Use formal interfaces to pass parameters and access and/or modify data structures
  - Therefore the term "modified microkernel"

Mach Microkernel OS Extended Memory Managment

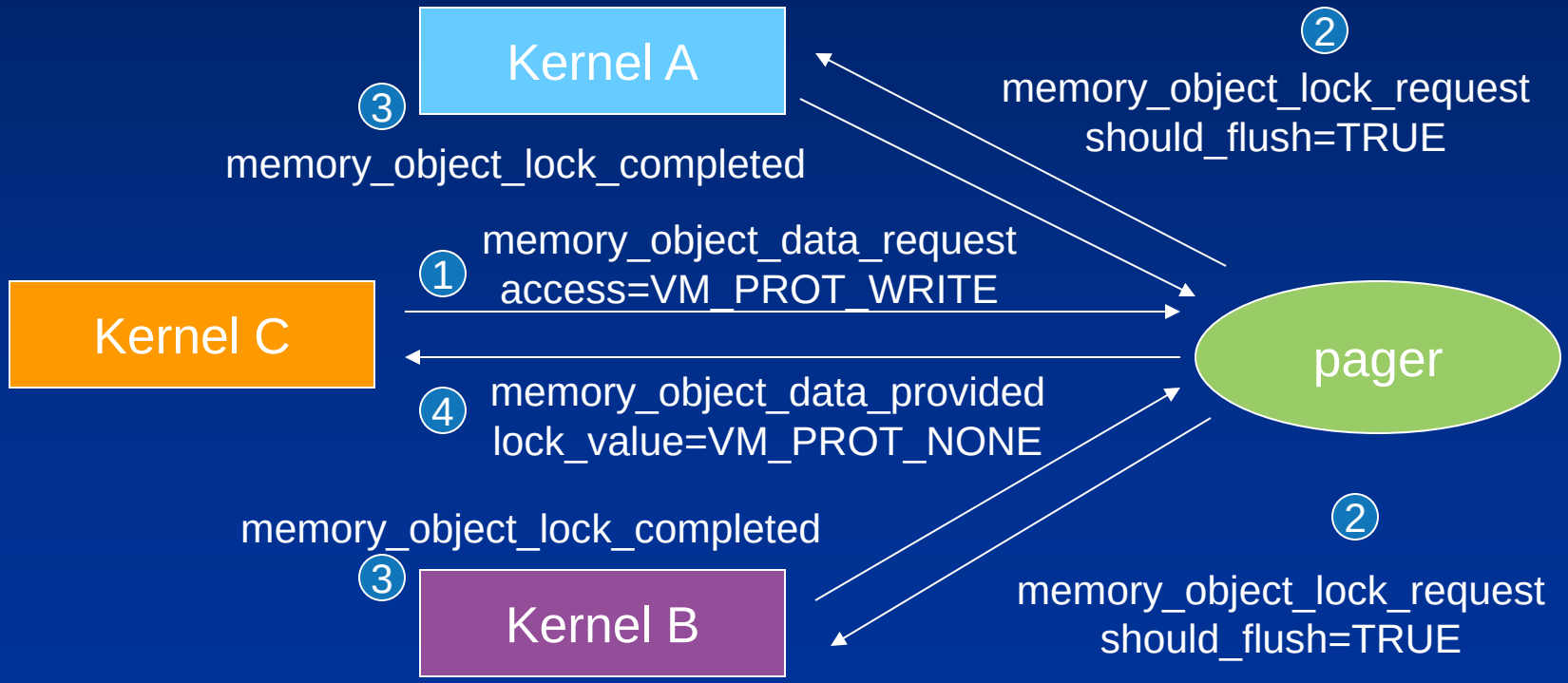
Paging handled by user-space server

Port: comm. endpoint, network-wide



Mach Microkernel OS Distributed Shared Memory System

- Access remote memories, port access rights – ACL
- Kernel C attempts to write to mem of A and B.



Windows NT Origins

- Design began in late 1988/early 1989 after Dave Cutler and a handful of Digital employees started at Microsoft
  - Dave Cutler—legend in the operating system world
    - Project leader for Digital's VMS (Virtual Memory System)
  - Internally, Windows NT has many similarities to Digital's VMS (scheduling, memory management, I/O and driver model)
  - VMS+1=WNT just a coincidence
- Original goal was replacement for OS/2
  - Later goal changed to be the replacement for Windows 3.0

Window NT Origins (Cont'd)

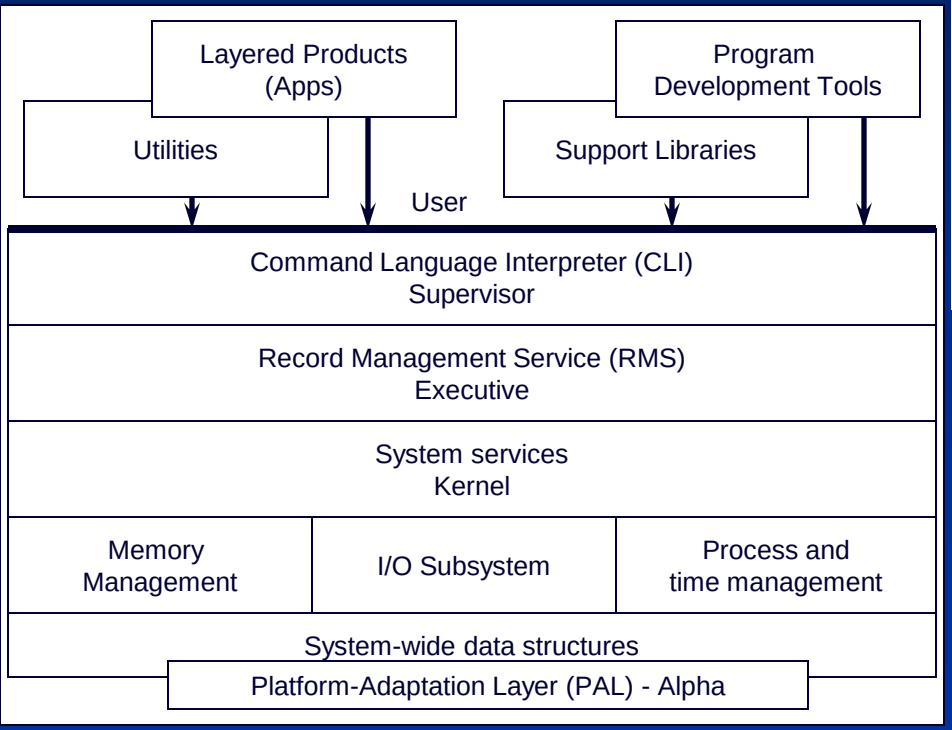
- The name "Windows NT" was chosen because
  - NT stands for New Technology
    - But at a high level, the architecture and user interface are not really that "new"
      (as compared to most 32-bit OS's)
  - Also, the i860 RISC CPU NT was originally targeted at was code named N-Ten
- Interesting book on the early years of NT:
  - Show-stopper!: The Breakneck Race to Create Windows NT and the Next Generation at Microsoft
  - By G. Pascal Zachary, ISBN: 0029356717

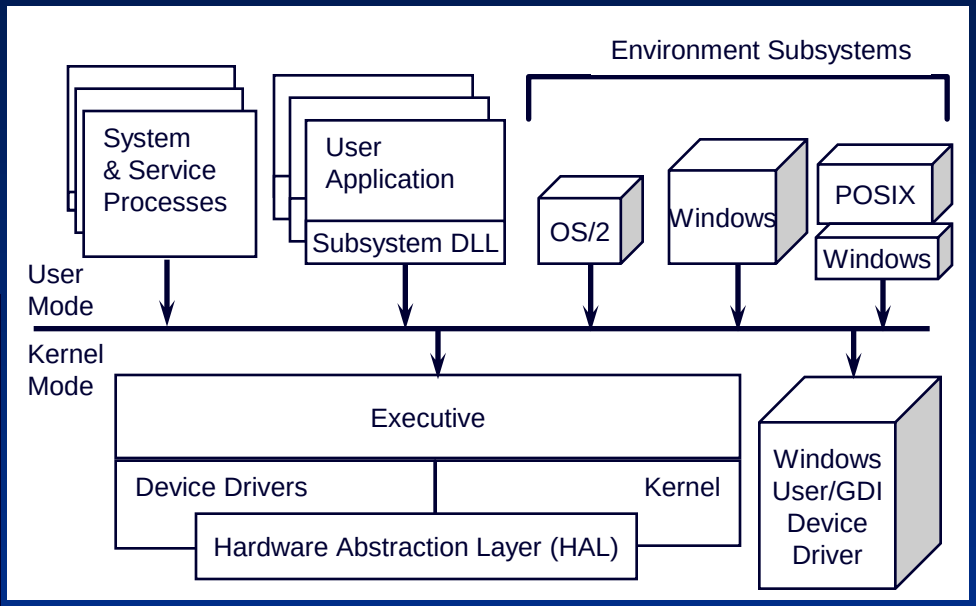
NOTE: In this course, "Windows" refers to Windows 2000, Windows XP, Windows Server 2003, Windows Vista and Windows 7

- Where there are specific differences, these will be noted

VMS and Windows
- a bird's-eye view on architectures

Layered design for VAX/VMS operating system





Windows high-level architecture

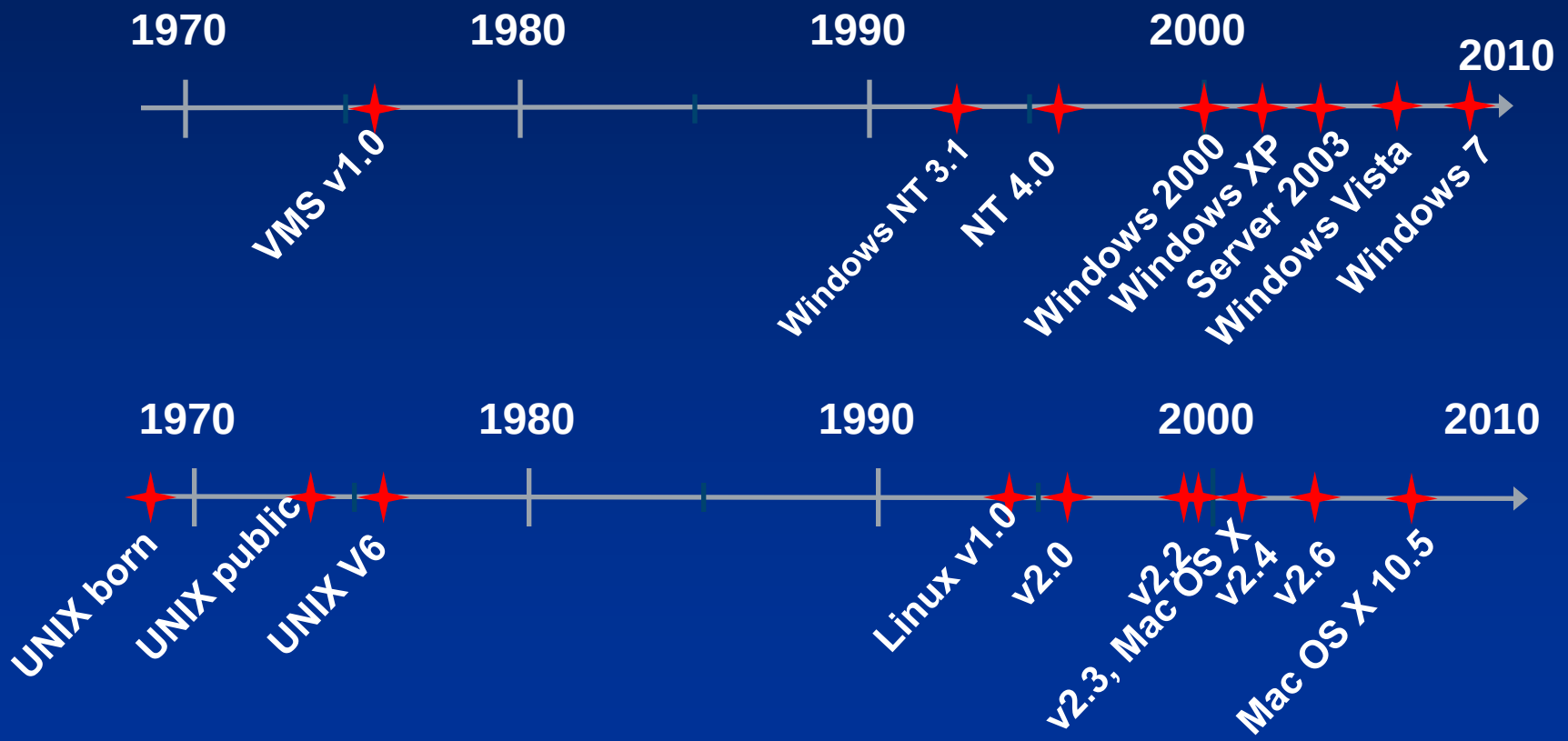
Release History

- Although product name has varied, internally, each version identified by a “build number”
  - Internal identification - increments each time NT is built from source (5-6 times a week)
  - Interesting timeline: http://windows2000.about.com/library/weekly/aa010218a.htm

Build#	Version	Date
297	PDC developer release	Jul 1992
511	NT 3.1	Jul 1993
807	NT 3.5	Sep 1994
1057	NT 3.51	May 1995
1381	NT 4.0	Jul 1996
2195	Windows 2000 (NT 5.0)	Dec 1999
2600	Windows XP (NT 5.1)	Aug 2001
3790	Windows Server 2003 (NT 5.2)	Mar 2003
6000	Windows Vista (Longhorn)	Nov 2006
6001	Windows Server 2008	Feb 2008
7600	Windows 7	Jul 2009

Windows And Unix Evolution

- Windows and Unix kernels are based on foundations developed in the mid-1970s



(see http://www.levenez.com for diagrams showing history of Windows & Unix)

Further Reading

- Dennis M. Ritchie, The Evolution of the Unix Time-sharing System,
  - in Proc. of Lang. Design and Programming Meth. Conf., Sydney, Australia, Sept 1979, Lecture Notes in Computer Science #79, Springer-Verlag, 1980.
- David Donald Miller, OpenVMS Operating System Concepts,
  - 2nd Ed., Digital Press, 1997.
  - History of Digital Operating Systems (from pp. 447)
- G. Pascal Zachary, Show Stopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft,
  - ISBN: 0029356717, Free Press, 1994.
- Mark E. Russinovich , *et al.*,
  Windows Internals,
  - 5th Edition, Microsoft Press, 2009.
  - Windows Versions (pp. 1)

Feedback?