Unit OS1:
Overview of Operating Systems

1.2. The Evolution of Operating Systems

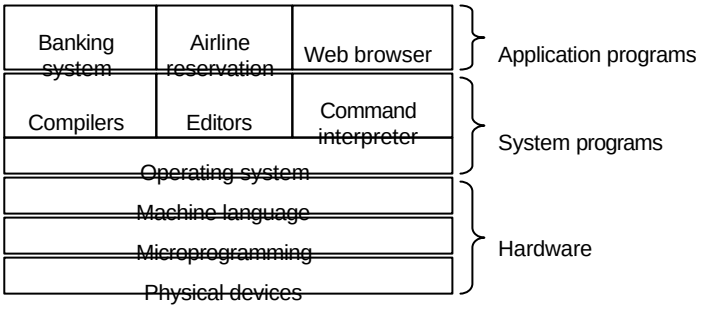
Copyright Notice

Roadmap for Section 1.2.

- History of Operating Systems
- Tasks of an Operating System
- OS as extension of the hardware
- Main concepts: processes, files, system calls
- Operating system structuring

Operating Systems Concepts

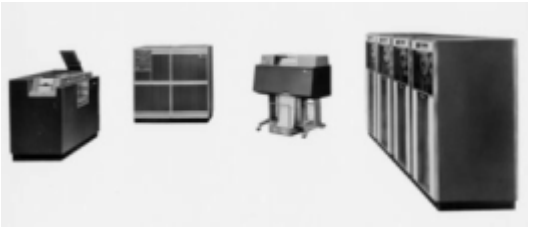
- System software manages resources
- OS hides complexity of underlying hardware
- Layered architectures



History of operating systems

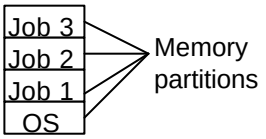
- Batch processing

  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 managment and sharing for multiple programs
  - 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/conurrent 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:

- 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

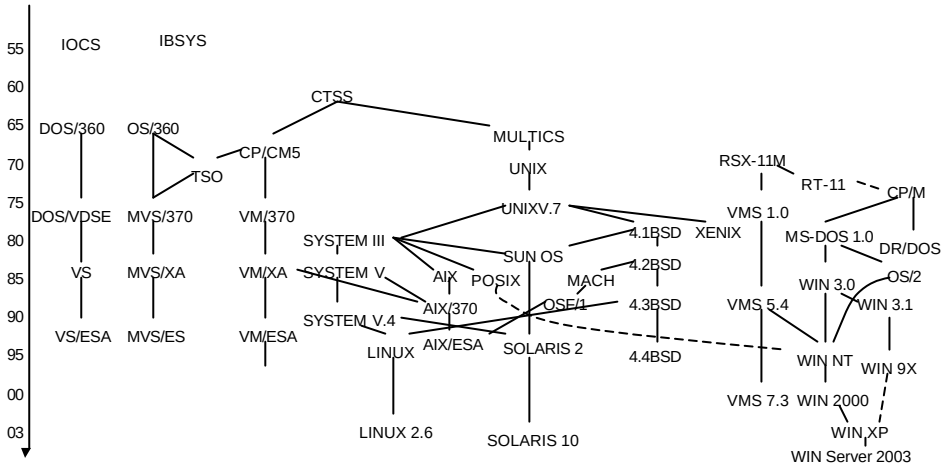
Layered Model of Operating System Concepts

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

OS acts as Extension of Hardware

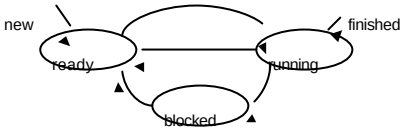
- System view: layered model of OS
  - Implementation details on one layer are hidden from higher layers
- Same machine, different operating systems:
  - IBM PC: DOS, Linux, NeXTSTEP, Windows NT, SCO Unix
  - DEC VAX: VMS, Ultrix-32, 4.3 BSD UNIX
- Same OS, different machines : UNIX
  - PC (XENIX 286, APPLE A/UX)
  - CRAY-Y/MP (UNICOS - AT&T Sys V)
  - IBM 360/370 (Amdahl UNIX UTS/580, IBM UNIX AIX/ESA)
- Windows XP (or Windows NT/2000)
  - Intel i386 (i486 an NT 4.0), Alpha, PowerPC, MIPS, Itanium

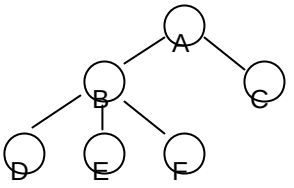
Operating Systems Evolution



Main Concepts: processes

- Processes, process table, core image
- Command interpreter, shell
- Child processes



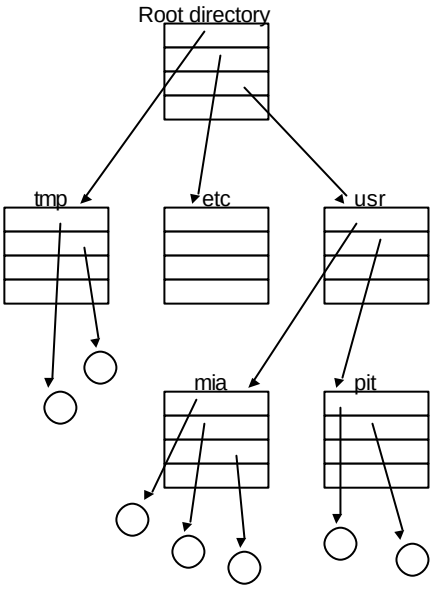


Process tree

- 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

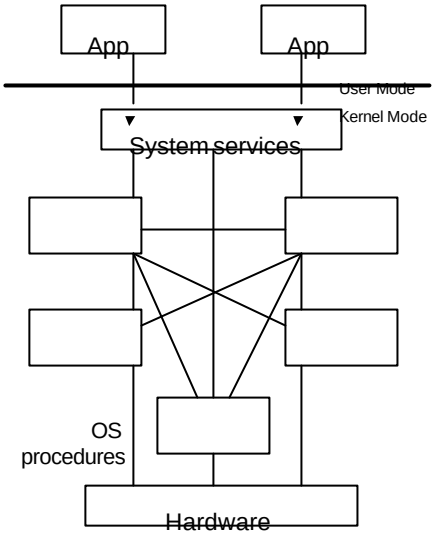
  $ sort <file1 >file2

  $ cat file1 file2 file3 > /dev/lp1

  $ make all >log 2>&1 &

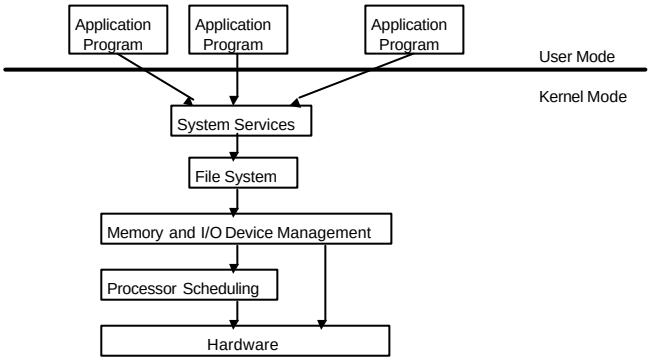
Structuring of Operating Systems

- **Monolithical systems**
- Unstructured
- Supervisor call changes from user mode into kernel mode



Layered OS

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

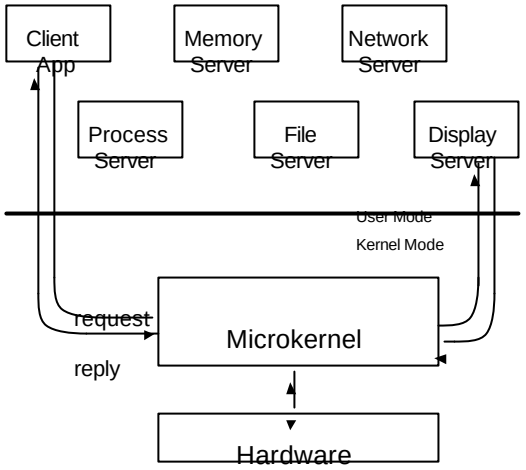


Microkernel OS (Client/server OS)

Kernel implements:

- Scheduling
- Memory Management
- Interprocess communication (IPC)

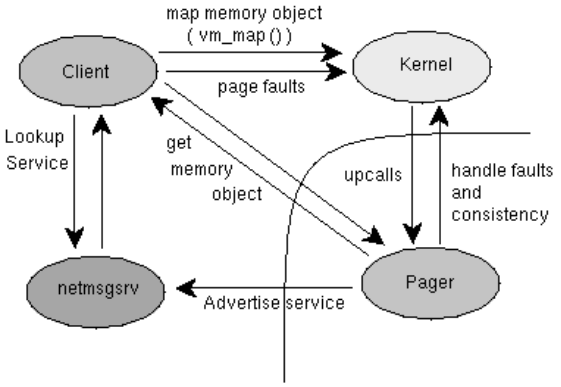
User-mode servers



Mach Microkernel OS
Extended Memory Managment

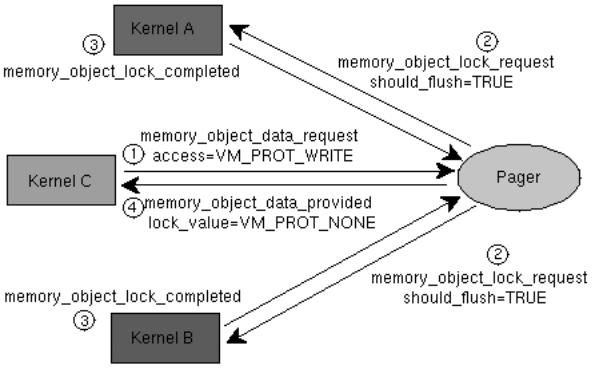
Paging handled by user-space server

Port: comm. endpoint, network-wide



Mach Microkernel OS
Distributed Shared Memory System

- Access remote memories, port access rights - ACL



Windows 2000/NT background/history

Dave Cutler:

- OS Developer at DEC 1971-1988
- RSX-11M, PDP-11 (16 bit mini) (**“Size is the Goal“)**
  - Multitasking, hierarchical file system, real-time scheduling
  - Application swapping, utilities
  - 32 kb of memory (!)
  - 16 kb Kernel, 16 kb utilities, overlay structures, assembly language
  - Time-to-market: 18 months
- VAX architecture (32 bit)
  - Most successful architecture in ´70s and ´80s

DEC Virtual Memory System (VMS) and MS Windows NT

- Cutler was leader of VMS development effort
- VAX-11 hardware had PDP-11 compatibility mode
  - RSX-11M was the compatibility environment to be supported by VMS
  - Binary and file system compatibility
- Biggest mistake: VMS written in assembly language
  - Size restrictions, no compiler available, engineering expertise
- Summer `88: call from Bill Gates
  - New OS for PC architecture
  - Portability, security, POSIX, compatibility, multiprocessor, extensibility
  - Similar goals as for PDP-11/VAX transition
- Windows NT came to market in 1993

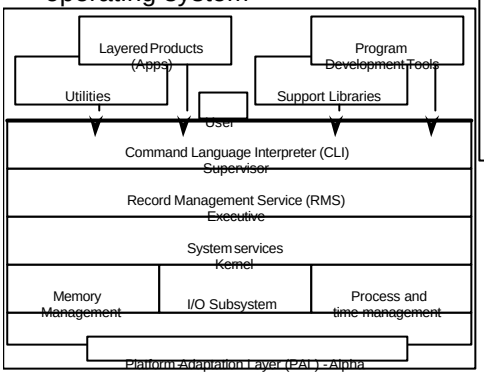
Windows NT Origins

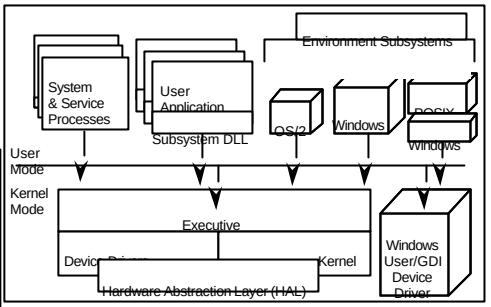
- **Design began in late 1988/early 1989 after Dave Cutler and a handful of Digital employees started at Microsoft**
  - Dave Cutler—legend in the operating system world
  - Internally, many similarities to Digital's VMS (scheduling, memory management, I/O and driver model)
  - VMS+1=WNT just a coincidence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

VMS and Windows NT

- a bird's-eye view on architectures

Layered design for VAX/VMS operating system





Windows NT high-level architecture

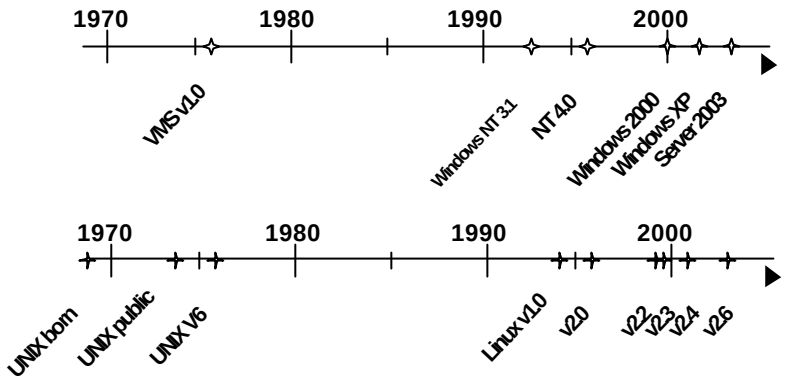
Release History

- 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
4051	Longhorn PDC Developer Preview	Oct 2003

Windows And Linux Evolution

- Windows and Linux 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 (pp. 447 ff.)
- Mark E. Russinovich and David A. Solomon, Microsoft Windows Internals,
  - 4th Edition, Microsoft Press, 2004.
  - Historical Perspective (pp. xix ff.)
- G. Pascal Zachary, Show Stopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft,
  - ISBN: 0029356717, Free Press, 1994.