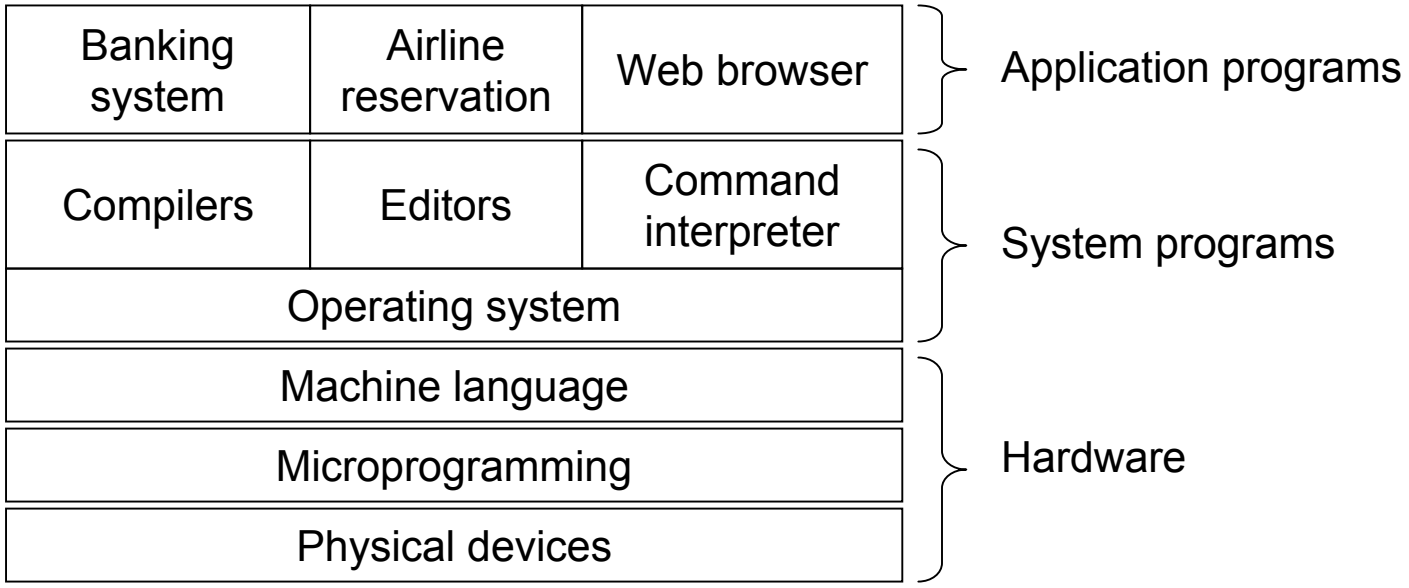
Unit 1: Introduction and Overview

1.2. The Evolution of Operating Systems

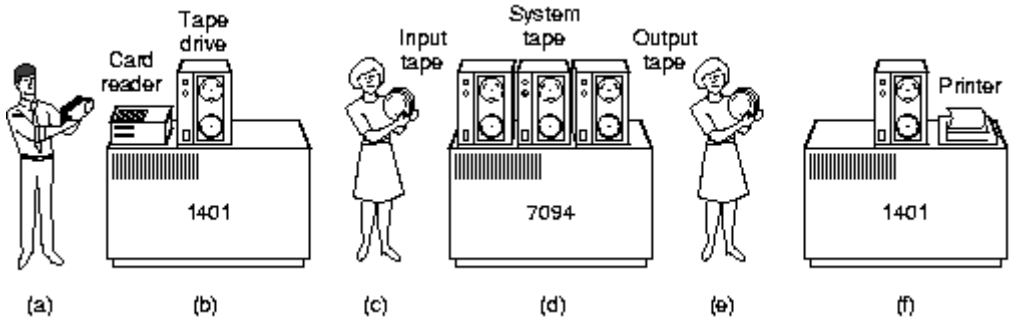
Operating Systems Concepts

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

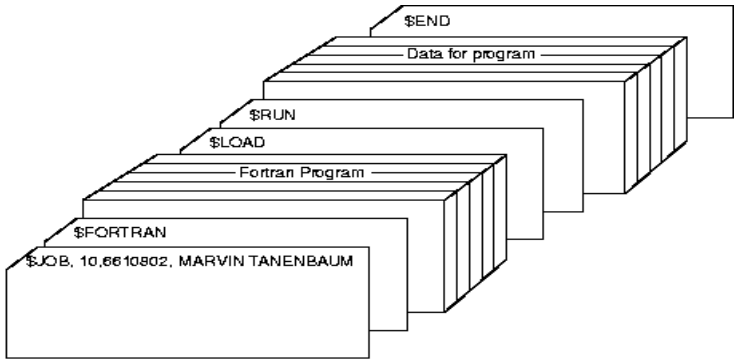


History of operating systems

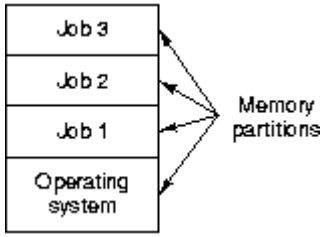
- Batch processing



- Punching cards



Multi programming



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 Space 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

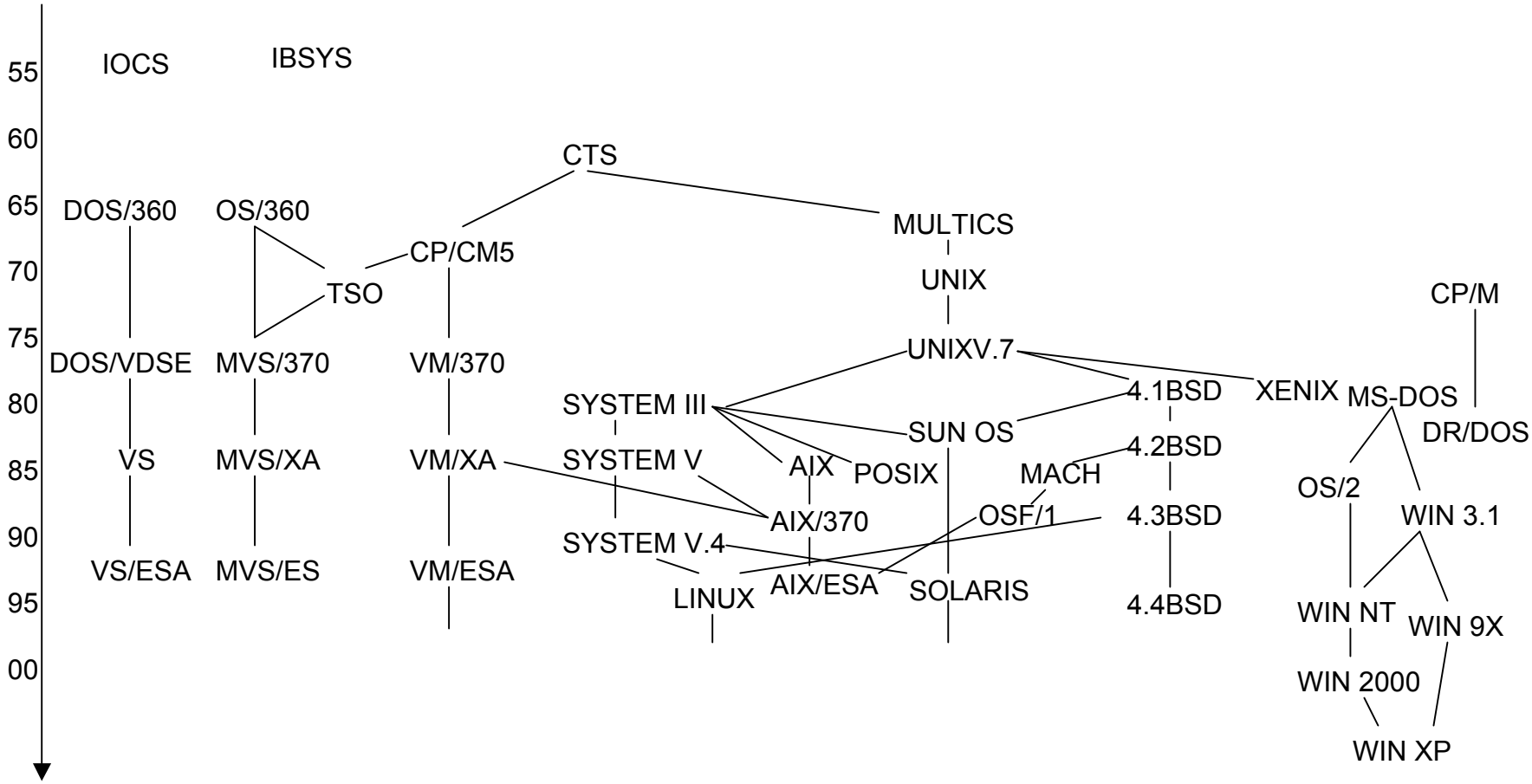
Layered Model of Operating System Concepts

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

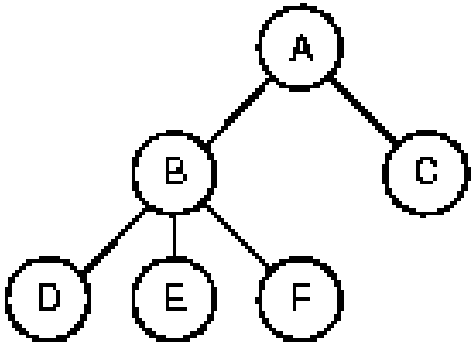
- System view: layered model of OS
  - Implementation details on one layer are hidden from higher layers
- Same machine, different operating systems:
  - IBM PC: DOS, Linux, NeXTSTEP, Windows NT, SCO Unix
  - DEC VAX: VMS, Ultrix-32, 4.3 BSD UNIX
- Same OS, different machines: UNIX
  - PC (XENIX 286,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

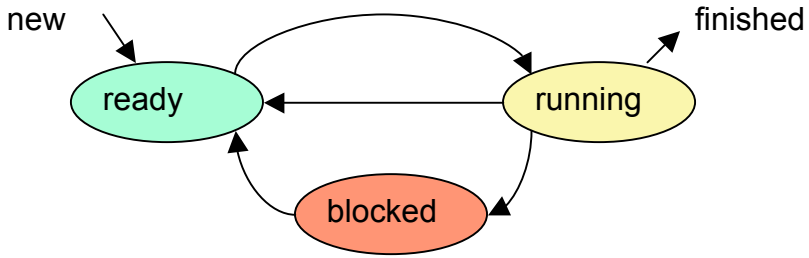
Operating Systems Evolution



Main Concepts: processes

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

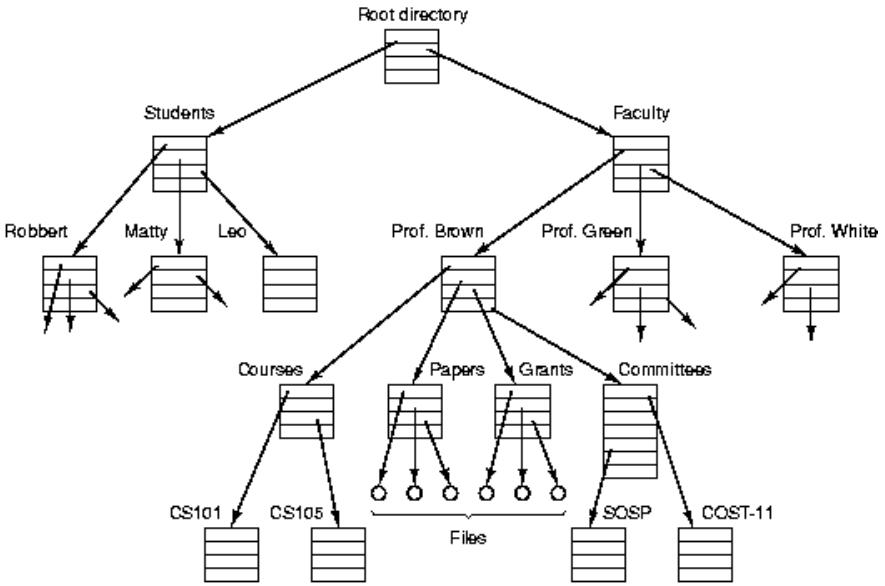




- Scheduling, signals
- User identification, group identification

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

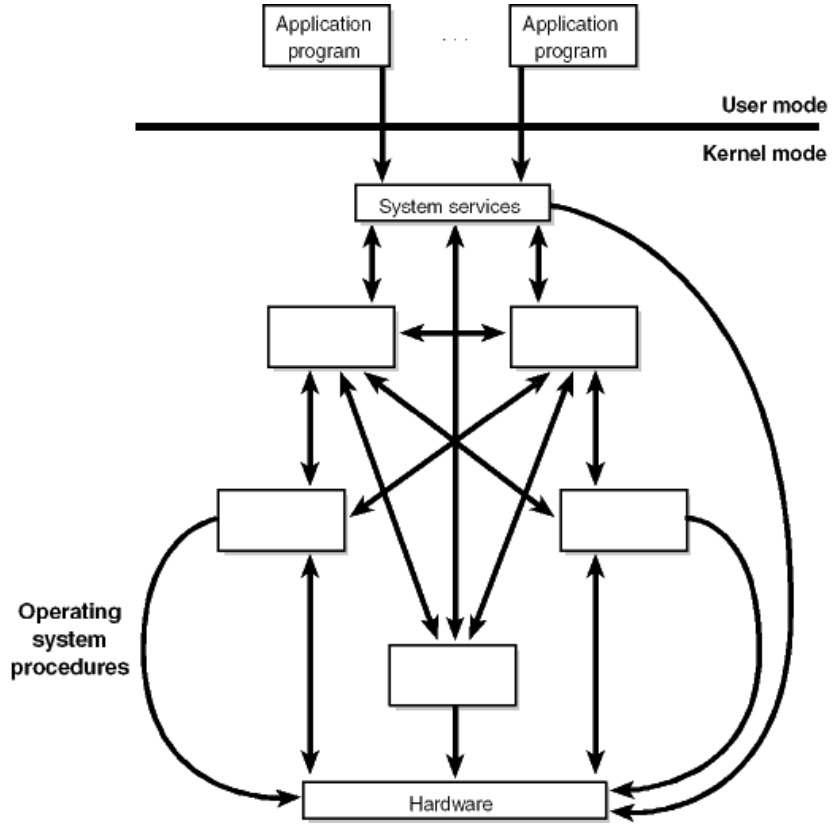
$ cat file1 file2 file3 > /dev/lp1

$ make all >log 2>&1 &

Structuring of Operating Systems

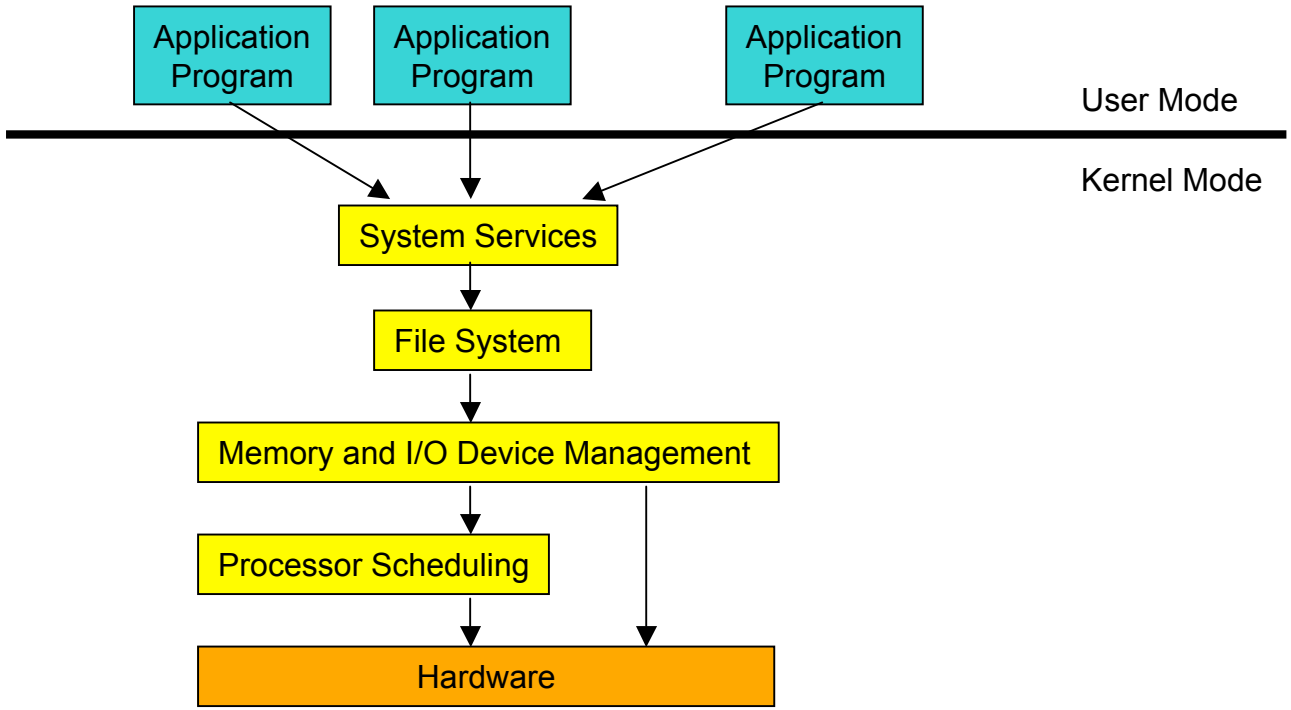
- **Monolithically systems**
- Unstructured
- Supervisor call changes from user mode into kernel mode

- Layered models
- Virtual machines
- Client/server systems



Layered OS

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

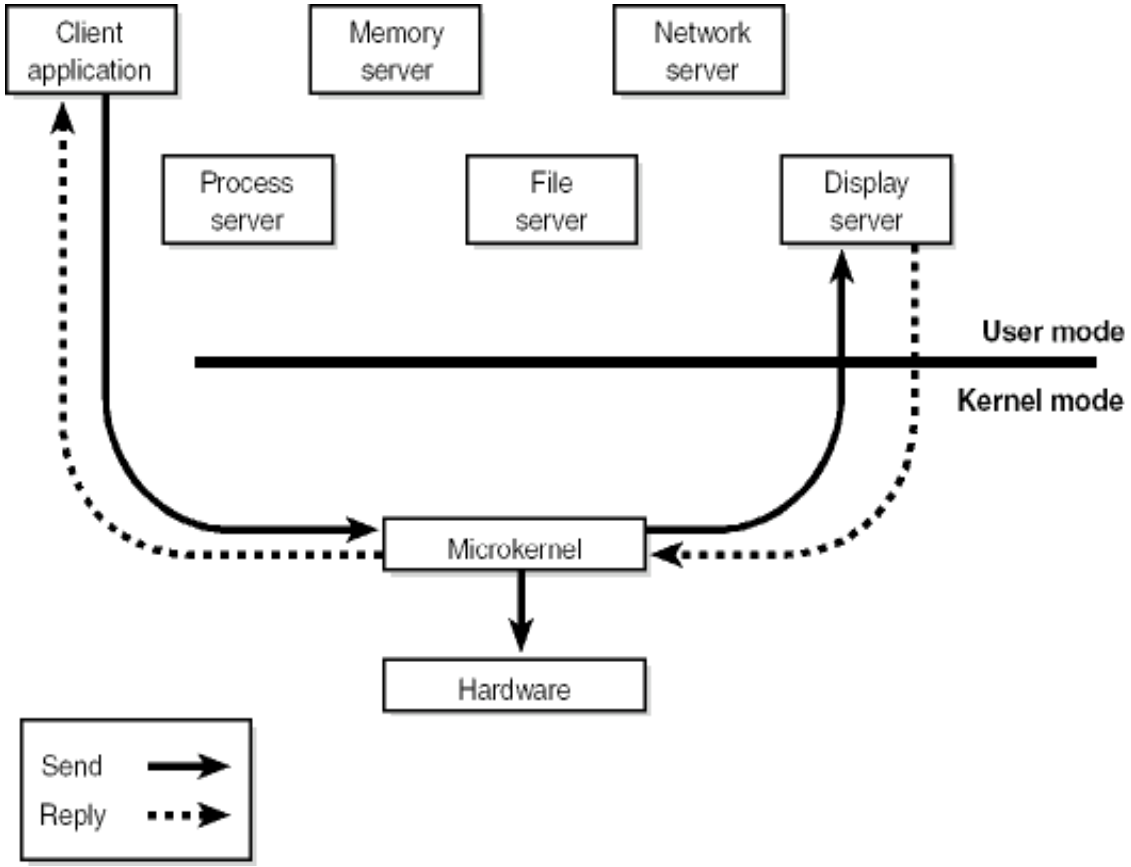


Microkernel OS (Client/server OS)

Kernel implements:

- Scheduling
- Memory Management
- Interprocess communication (IPC)

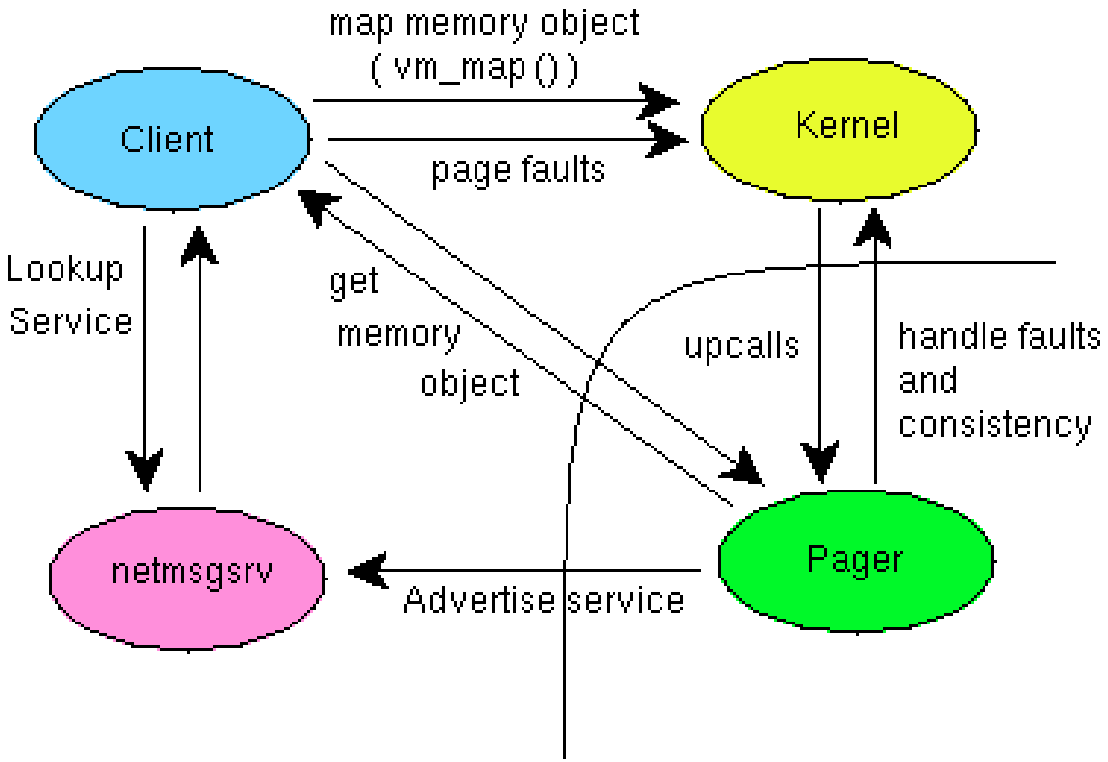
User-mode servers



Mach Microkernel OS
Extended Memory Managment

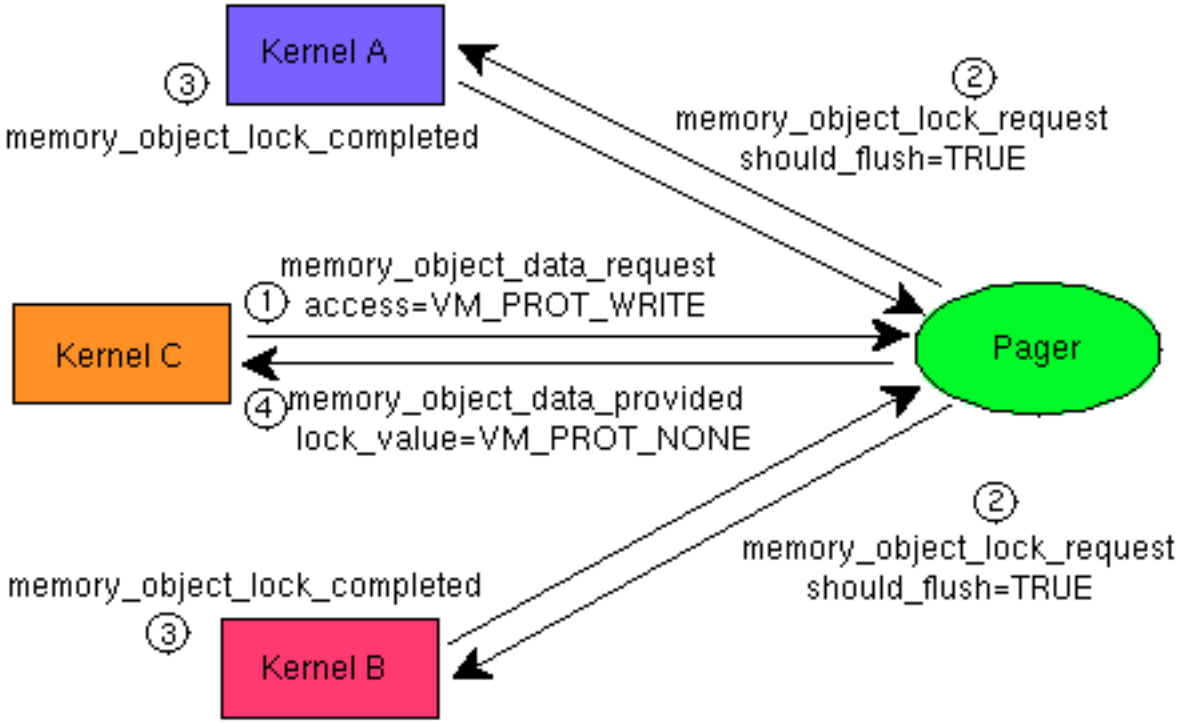
Paging handled by user-space server

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 since 1971
- 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
- Lack of address bits: 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