

Hardware Leads the OS

- Because of the better performance of modern computers, the OS can provide more services.
- Older computers didn't have the speed or storage to run the OS of today.

1

The Evolution of Operating Systems

- No Operating Systems
- Monitors
- Simple Batch Systems
- Multiprogrammed Batch Systems
- Time Sharing Systems

2

No Operating Systems

- Each program directly interfaced with the hardware.
- One person used the computer at a time.
- "Job Scheduling" was done with a clipboard
- Libraries of commonly used procedures were the first start of operating systems.

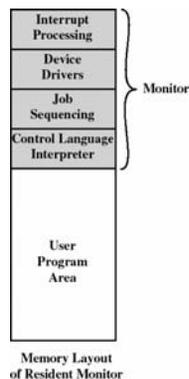
3

Monitors

- The monitor was a program that loaded application programs into RAM.
- The monitor, or a small portion of it, remained in RAM while the application program ran.
- The monitor contained device drivers to simplify access to peripherals.
- When the program terminated, it would jump back to the monitor.

4

- The monitor loads a job into the User Program Area
- A monitor instruction branches to the start of the user program
- When the user program is finished, the CPU fetches instructions from the Monitor



5

Simple Batch Systems

- The user submit a job (written on punched cards) to the computer.
- The operating system would copy the input data to a disk.
- When a job completed, the OS would select one of the jobs from the disk and run it.
- Printed output from the job was written to the disk. When the program terminated, the output file was copied to the printer.

6

Job Control Language (JCL)

- JCL was used to specify commands to the operating system.
- The first characters of the input card identified it as JCL or data.
- The JCL specified what program was to be run or what data file was to be used.

7

JCL Example

```
//KENJOB  RUN ACCOUNT=COMP450
//  DD  DDNAME=*
data cards
//  EXEC  DDNAME=WILLIAMS . PROG
//
```

8

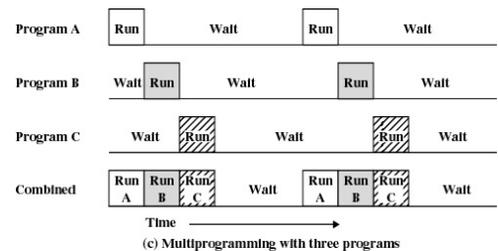
Single Program Execution

- Processor must wait for I/O instruction to complete before proceeding



9

Multiprogramming



10

Multiprogrammed Batch Systems

- Several programs had to be kept in RAM at the same time, each protected from the other.
- The OS had to be able to switch from one user environment to another.
- Relied on hardware that supports I/O interrupts and DMA

11

Desirable Hardware Features

- Memory protection
 - do not allow the memory area containing the monitor to be altered
- Timer
 - prevents a job from monopolizing the system
 - an interrupt occurs when time expires
- Privileged instructions
 - executed only by the monitor
- Interrupts
 - provides flexibility for controlling user programs

12

Time Sharing Systems

- Using multiprogramming to handle multiple interactive jobs
- Processor's time is shared among multiple users
- Multiple users simultaneously access the system through terminals
- A new control language was required for interactive work.

13

Compare Batch Multiprogramming & Time Sharing

	Batch Multiprogramming	Time Sharing
Objective	Maximize processor use	Minimize response time
Source of instructions to OS	Job Control Language instructions provided with Job	Commands entered at the terminals

14

Memory Organization

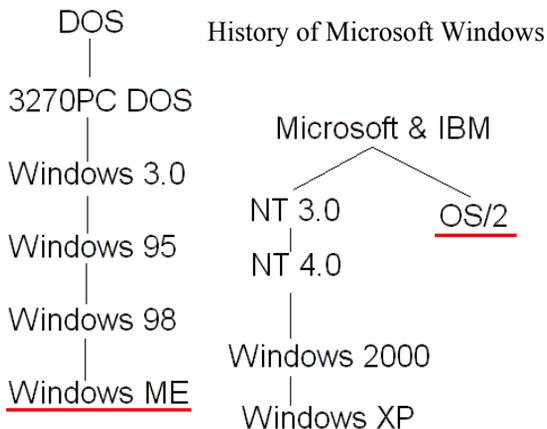
- Early computers did not have cache or Virtual Memory. Cache has little impact on the OS.
- Some early machines had two types of RAM. The OS moved jobs between the fast and slow RAM.

15

Virtual Memory

- The IBM/370 introduced virtual memory.
- The Intel 386 provided virtual memory support.
- The operating system has to move pages between RAM and disk.
- The OS has to maintain the page tables and addressing environment.

16



History of Unix

- Originally developed for a PDP-7 in 1970 by Brian Kernighan and Dennis Richie.
- Written in C in 1973
- OS with source code was available free.
- Many variants available
- Andrew Tanenbaum created Minix
- Linus Torvalds extended Minix to Linux

18