Evolution of OS

S.Rajalakshmi

AP/CSE

SSNCE

History of OS

- Phase 1: Hardware is expensive, humans are cheap
  - User at console: single-user systems
  - Batching systems
  - Multi-programming systems
- Phase 2: Hardware is cheap, humans are expensive
  - Time sharing: Users use cheap terminals and share servers
- Phase 3: Hardware is very cheap, humans are very expensive
  - Personal computing: One system per user
  - Distributed computing: lots of systems per user
- Phase 4: Ubiquitous computing/Cloud computing
  - Cell phone, mp3 player, DVD player, TIVO (Digital Video Recorder), PDA, iPhone, eReader
  - Software as a service, Amazon's elastic compute cloud (EC2)

History Phase I: Hardware Expensive, Humans Cheap

- Hardware: mainframes
- OS: human operators
  - – Handle one ***job*** (a unit of processing) at a time
  - – Computer time wasted while operators walk around the machine room

Hand programmed machines (1945-1955)

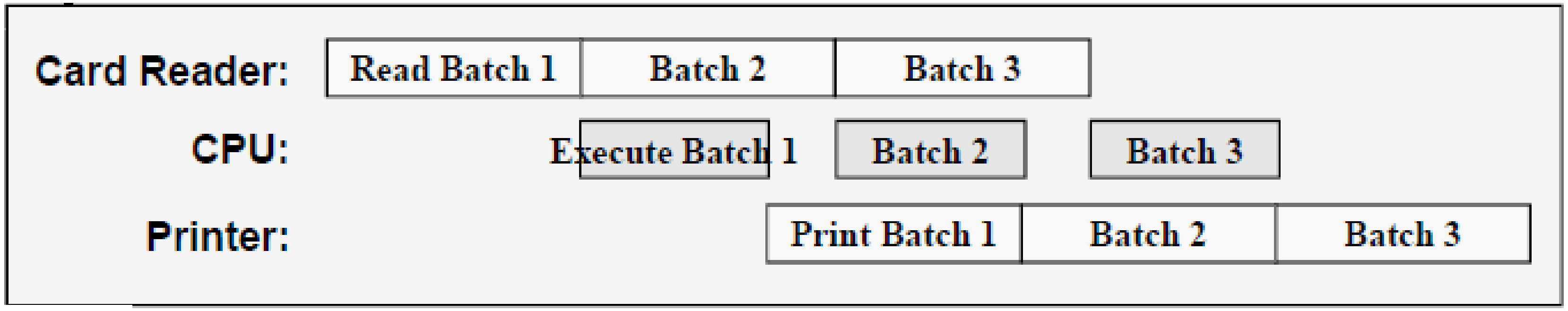
- Single user systems
- OS = *loader + libraries of common subroutines*
- Problem: low *utilization of expensive components*
- *% of Utilization = Exec time / (Exec time + card reader time)*

OS Design Goal

- Efficient use of the hardware
  - ***Batch system***: collects a batch of jobs before processing them and printing out results
    - Job collection, job processing, and printing out results can occur concurrently
  - ***Multiprogramming***: multiple programs can run concurrently
    - Example: I/O-bound jobs and CPU-bound jobs

Batch/Offline processing (1955 - 1965)

- Batching *v. sequential execution of jobs*



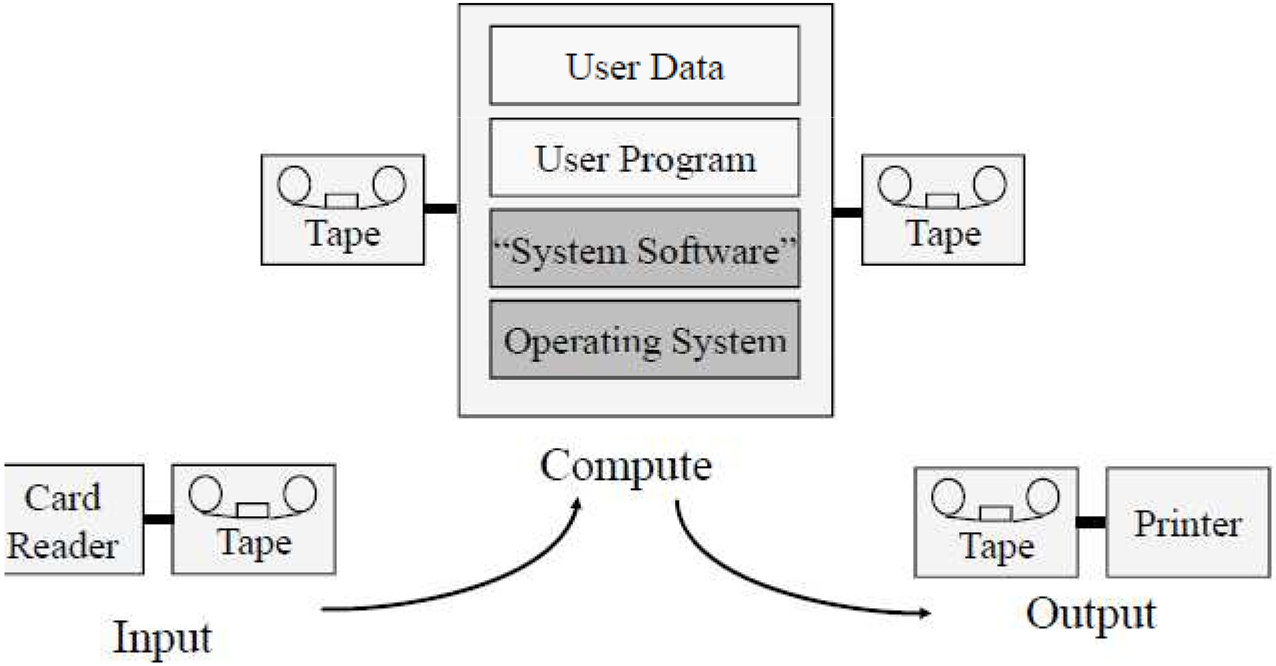
Card Reader: Read Batch 1 | Batch 2 | Batch 3

CPU: Execute Batch 1 | Batch 2 | Batch 3

Printer: Print Batch 1 | Batch 2 | Batch 3

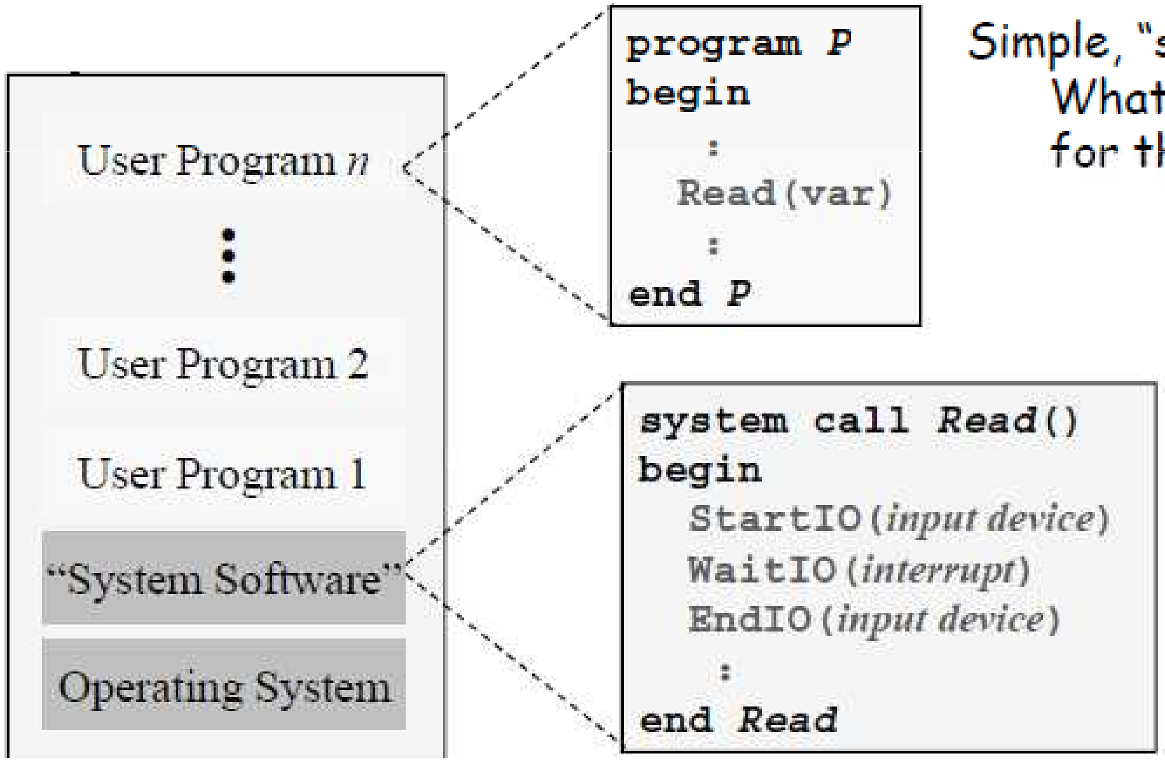
Batch/Offline processing (1955 - 1965)

- Operating system = *loader + sequencer + output processor*



Multiprogramming
(1965 -1980)

- Keep several jobs in memory and multiplex CPU between jobs

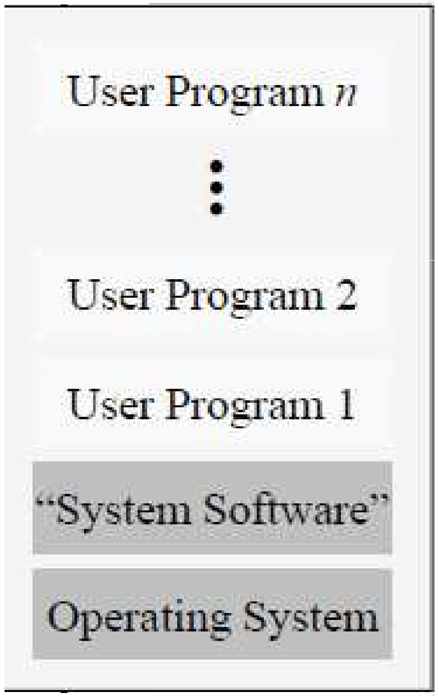


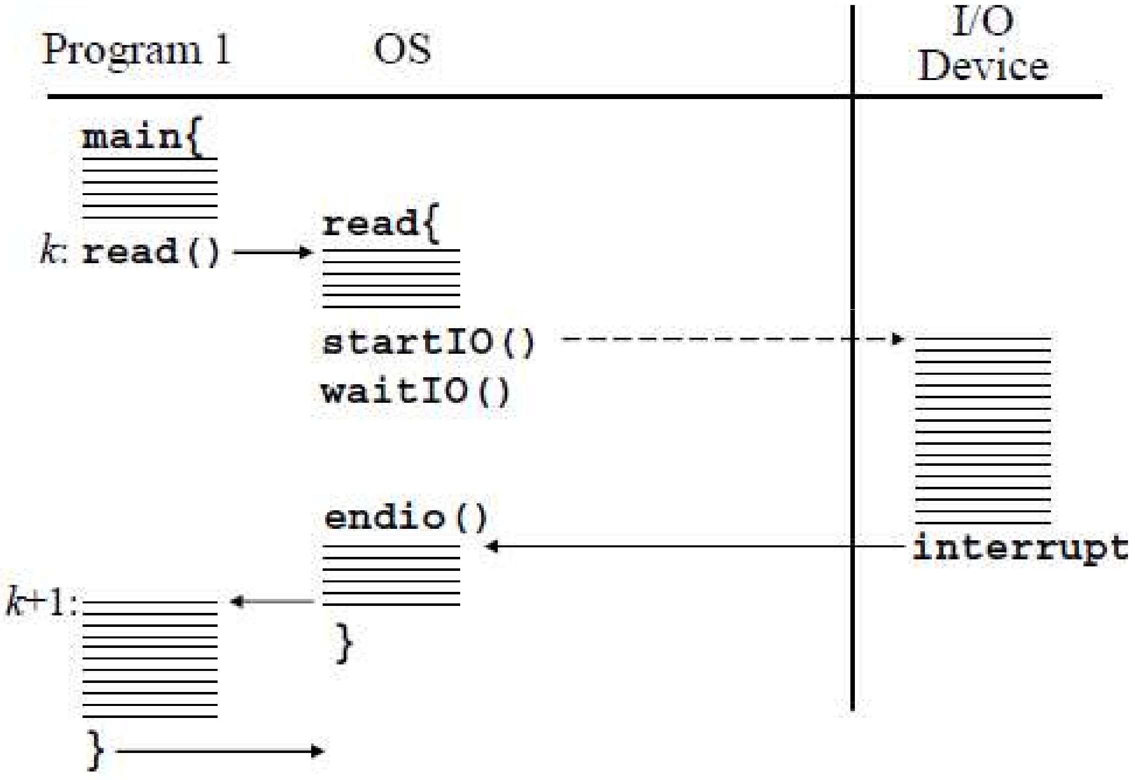
```
program P
begin
   :
  Read(var)
   :
end P
```

```
system call Read()
begin
  StartIO(input device)
  WaitIO(interrupt)
  EndIO(input device)
   :
end Read
```

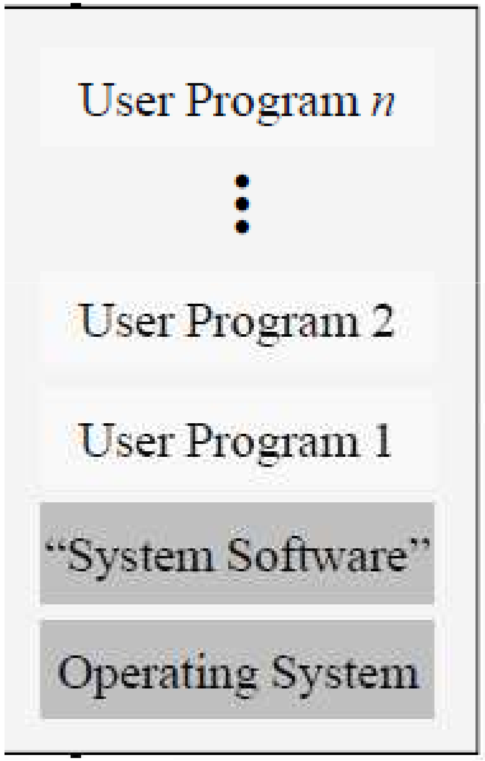
Simple, "synchronous" input:
What to do while we wait for the I/O device?

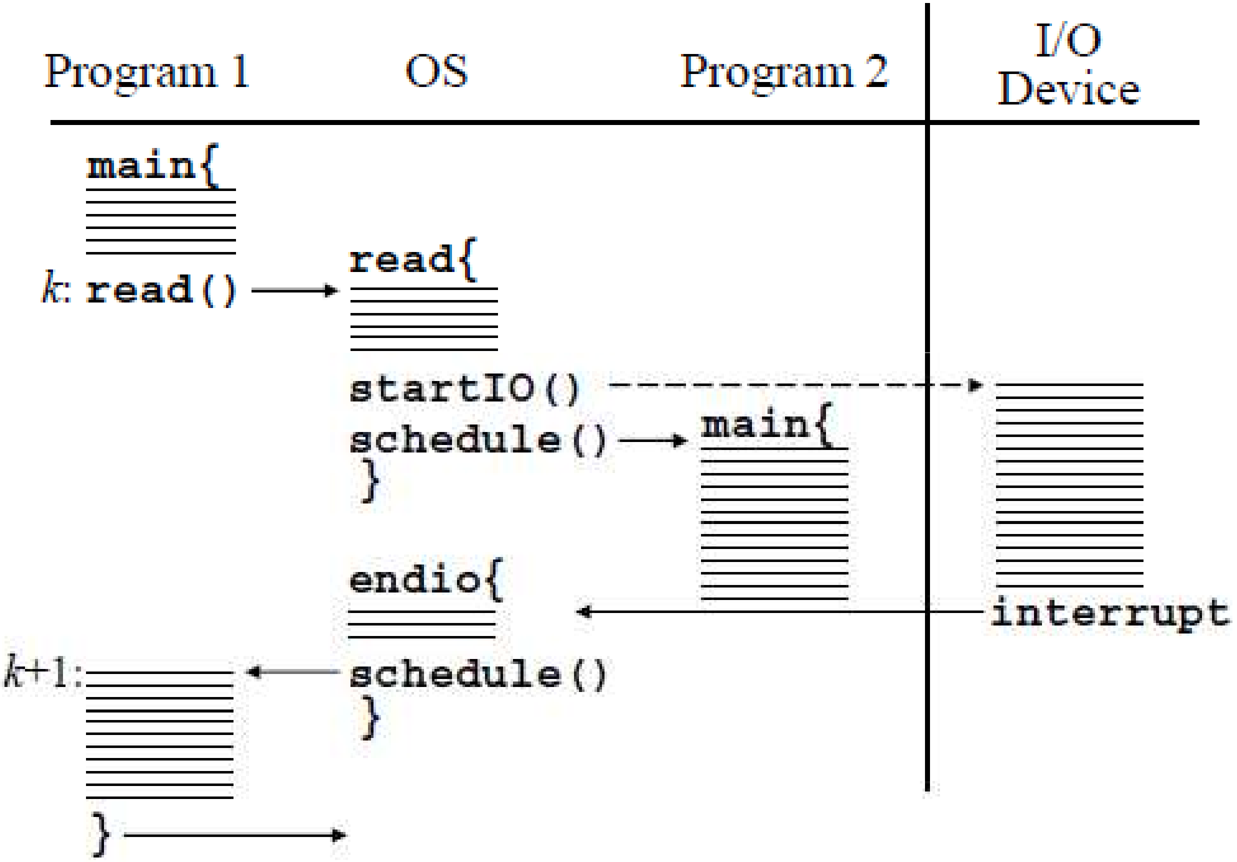
Multiprogramming (1965 -1980)





Multiprogramming (1965 -1980)



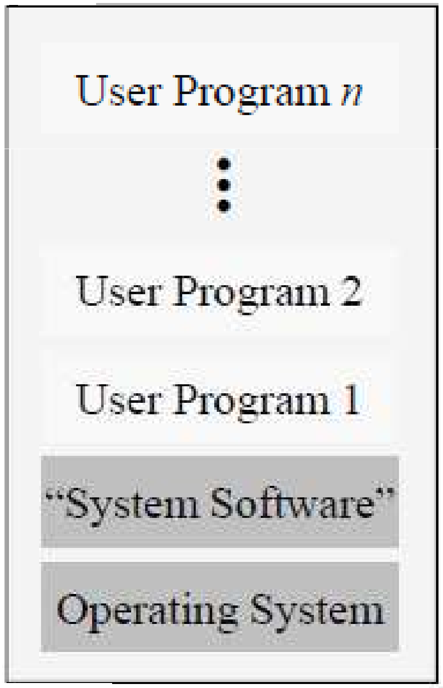


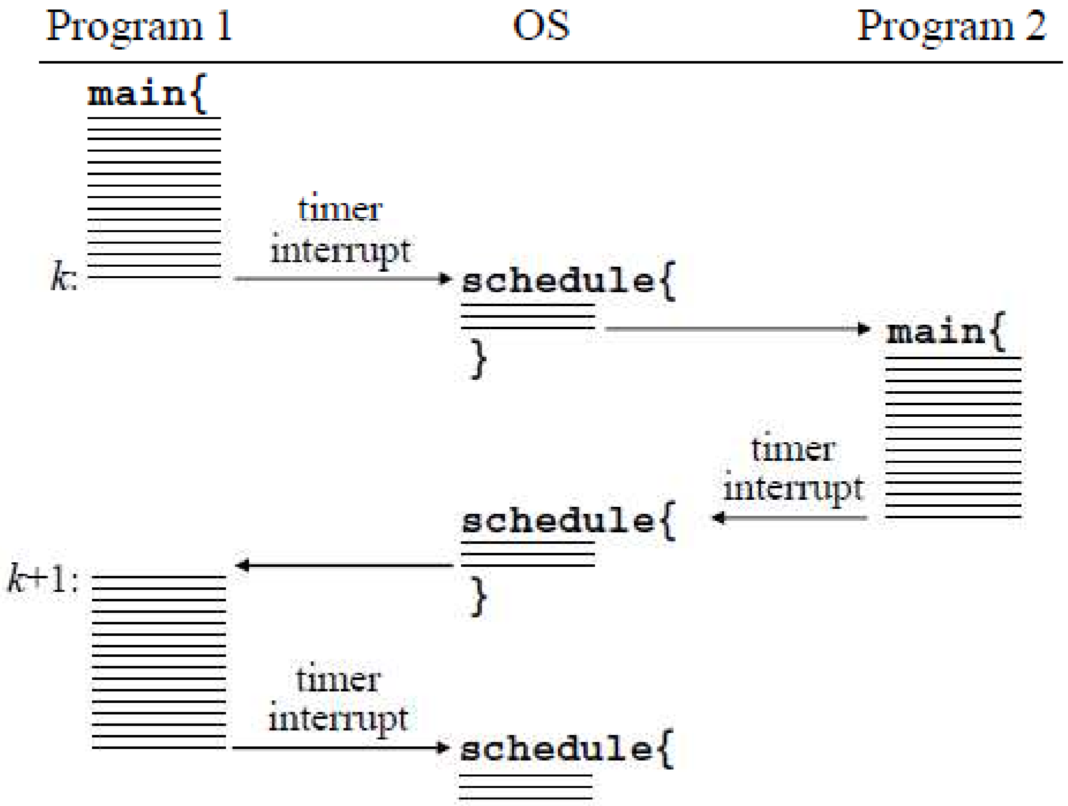
History Phase II: Hardware Cheap, Humans Expensive

- Hardware: terminals
- OS design goal: more efficient use of human resources
  - *Timesharing systems*: each user can afford to own terminals to interact with machines

Timesharing (1970) -

- A timer interrupt is used to multiplex CPU among jobs





History Phase III: Hardware Very Cheap, Humans Very Expensive

- Hardware: personal computers
- OS design goal: allowing a user to perform many tasks at the same time
  - *Multitasking*: the ability to run multiple programs on the same machine at the same time
  - *Multiprocessing*: the ability to use multiple processors on the same machine

History Phase III: Distributed Systems

- Hardware: computers with networks
- OS design goal: ease of resource sharing among machines

History Phase IV: Cloud Computing

- **Cloud computing is where dynamically scalable and** often virtualized resources are provided as a service over the Internet
- Infrastructure as a service (IaaS)
  - Amazon’s EC2 (Elastic Compute Cloud)
- Platform as a service (PaaS)
  - Google gears
  - Microsoft azure
- Software as a service (SaaS)
  - gmail
  - facebook
  - flickr

History Lessons

- From the late 1970s to early 1980s, UNIX evolved into two related but distinct camps.
- One camp centered about a commercial UNIX developed by American Telephone and Telegraph (AT&T) .
- The AT&T versions were named System III and later System V.

History Lessons

- The other popular UNIX variant, Berkeley Standard Distribution or BSD UNIX, was funded by the Department of Defense's Advanced Research Projects Administration.
- Networking was added to BSD UNIX making it an important force in the early development of the Internet.
- BSD UNIX formed the basis for the early SunOS, IRIX, NeXTSTEP and later versions of BSD.
- The most common current BSD-styled UNIXes are the BSD triplets, OpenBSD, FreeBSD, NetBSD.

History Lessons

- Differences between flavors of UNIX become more apparent the more sophisticated the user becomes.
- At the most basic level, all UNIX variants share a common set of user level commands that function in a nearly identical manner with the exception of some option flags.
- More sophisticated users will note the different option flags.
- Programmers will note that the BSD and System V families of UNIX have significant differences in libraries and system calls.
- The differences between UNIX variants will be most apparent to the system administrator.

History Lessons

- From the perspective of a system administrator, a BSD derived UNIX has several noteworthy differences from its System V cousins.
- The option flags and output format for several programs, are different from their System V counterparts.
- BSD UNIXes use a small collection of run control files (e.g. /etc/rc, /etc/rc.local) which each start multiple service daemons.
- The BSD UNIX device naming convention for disk and tape drives usually uses a flat directory scheme with a letter, number, letter pattern (e.g. /dev/rz0a) to specify the device driver, device number and partition or density.

History Lessons

- So where does Linux fit into all this?
- Linux is sort of a mutt as far as its orientation with regard to the BSD and System V styles of UNIXes.
- Many of the commands share the BSD style option flags and output
- Run-control files and other aspects of system administration vary between distributions.

History Lessons

- Where did Windows come from?
- At nearly the same time the two major UNIX varieties were establishing themselves and the commercial UNIX derivatives were being born, the personal computer was being created.
- Bill Gates and Microsoft delivered the first PC operating system, the command line oriented Disk Operating System (DOS).

History Lessons

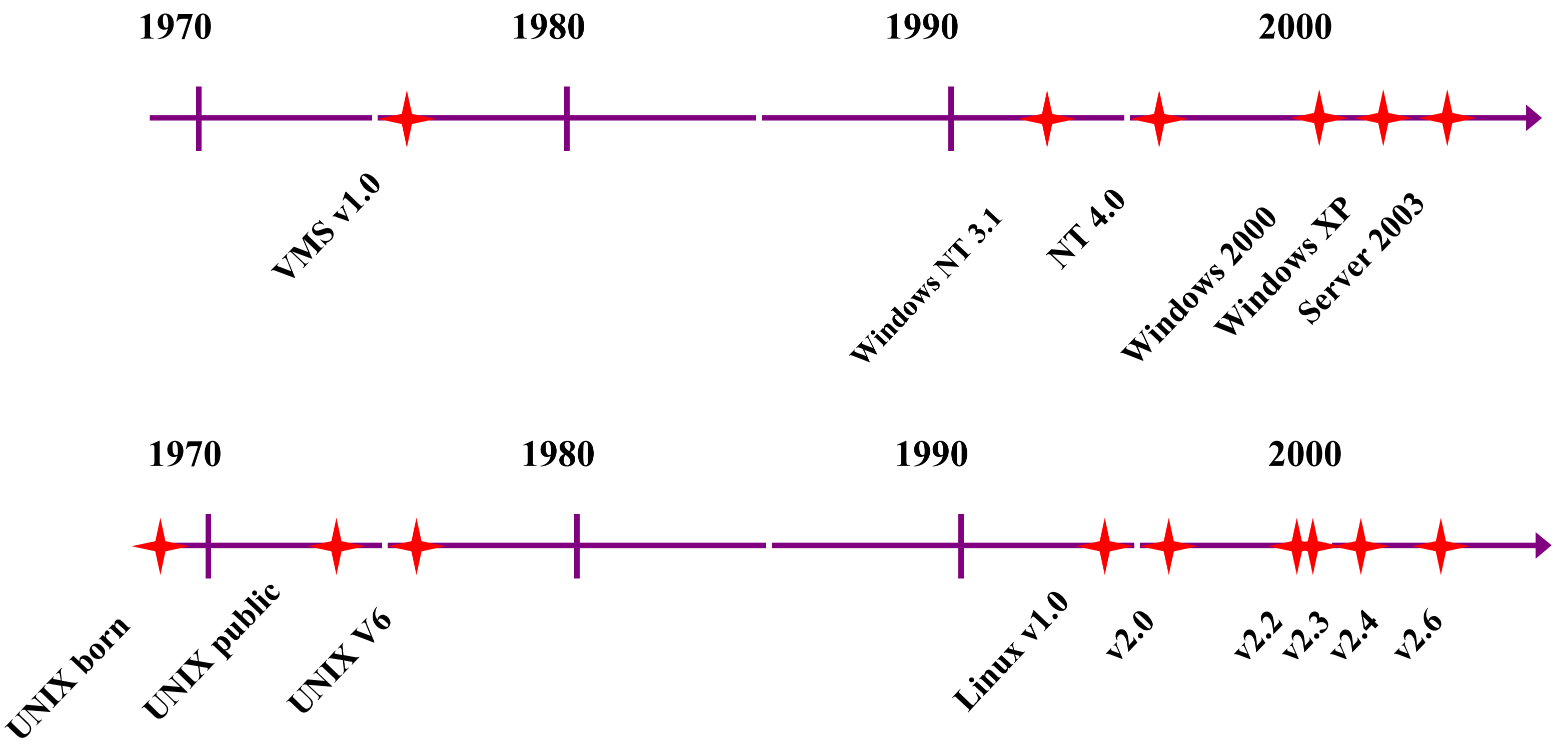
- In an effort to compete with the Apple graphical user interface (GUI), the Windows Operating System was developed.
- Early versions of Windows lacked the maturity of the Apple GUI.
- The Windows development split into two streams: a “home” or personal version of Windows, and a “business” or commercial version.
- Recently, Microsoft has been trying to merge the two versions into a single offering.

History Lessons

- Apple and Mac OS X
- Driving along a third parallel set of tracks, Apple developed Mac OS for their Macintosh computer during roughly the same time frame as Windows.
- Inspired by the work Xerox had done on graphical user interfaces, Apple developed the Macintosh, the first commercially successful personal computer with a graphical user interface.

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)