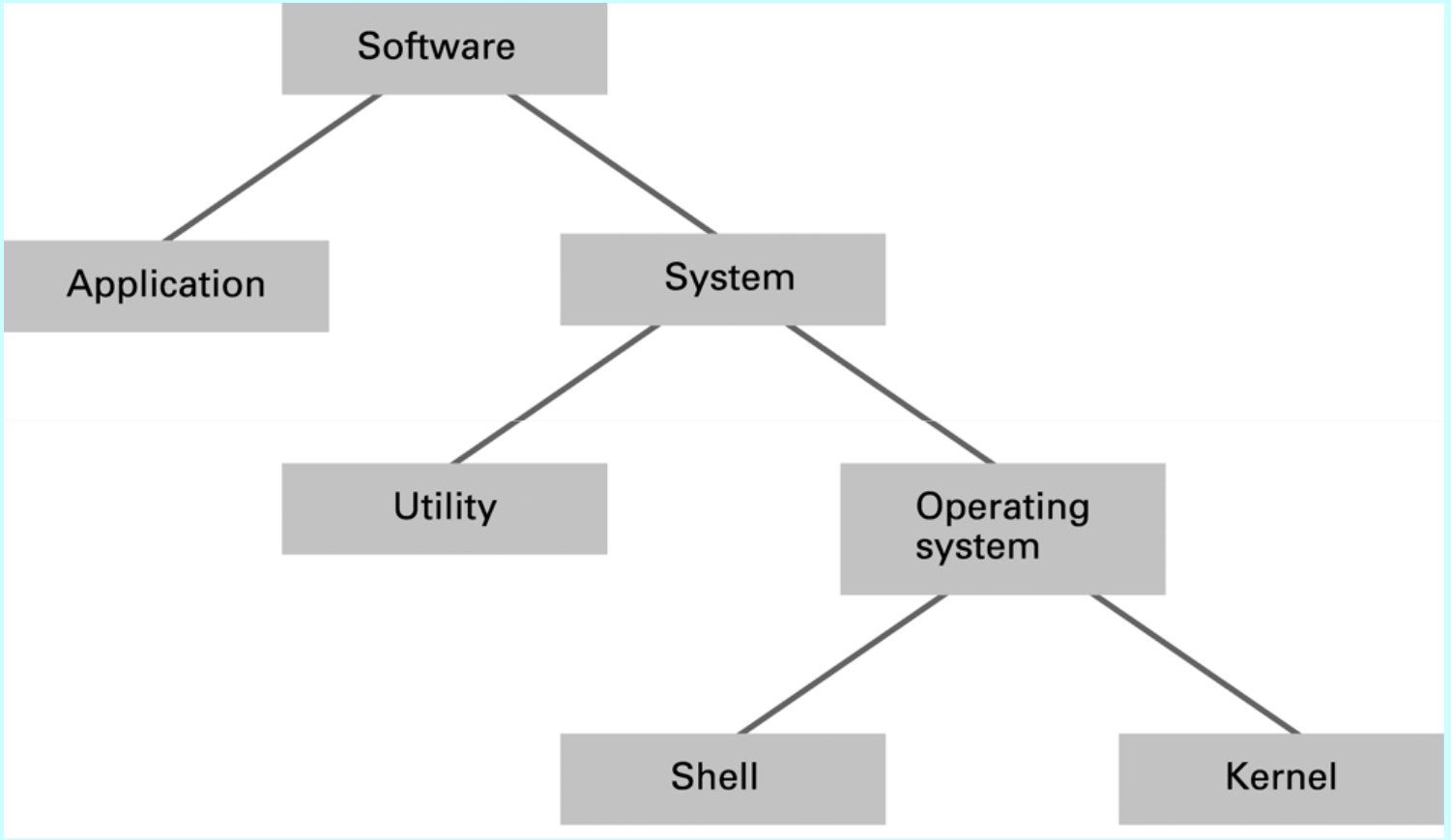
Operating Systems - Introduction

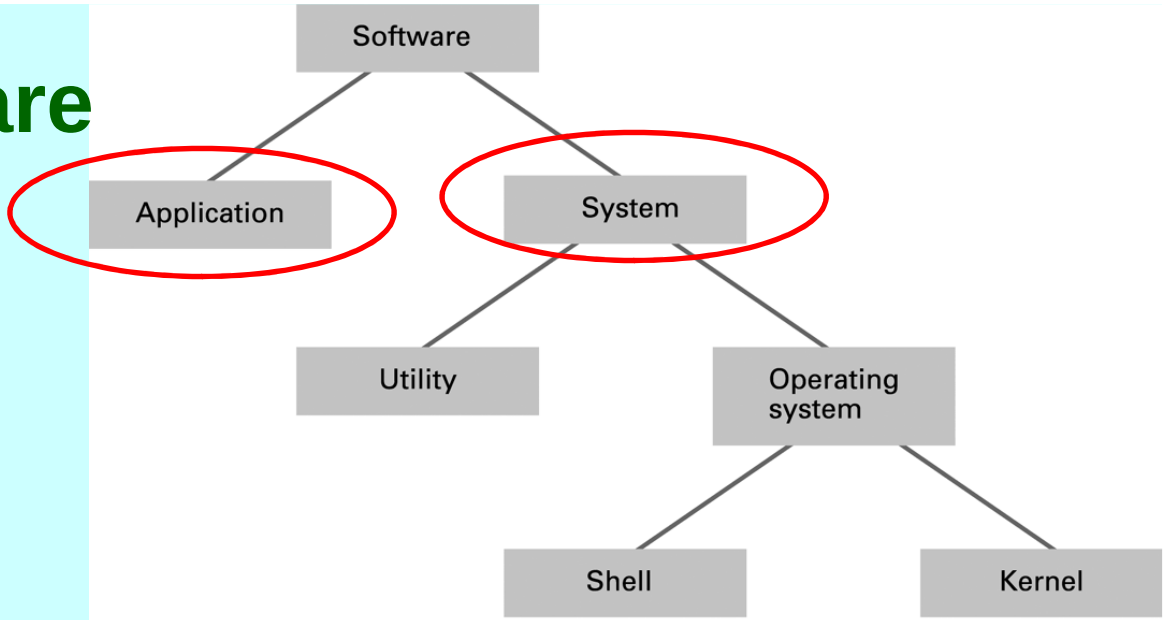
Chapter 1: Introduction

- What is an Operating System?
- Mainframe Systems
- Desktop Systems
- Multiprocessor Systems
- Distributed Systems
- Clustered System
- Real -Time Systems
- Handheld Systems
- Computing Environments

Software


Software
Application
System
Utility
Operating system
Shell
Kernel

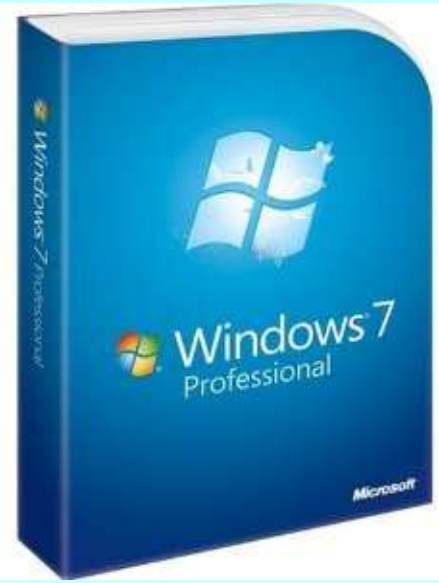
Software



- **Application software**: Programs for performing a specific task
  - Word processing, spreadsheets, gaming, web page design, graphic design
- **System software**:
  - **Operating software**: Software that controls the overall operation of the computer
    - (more next)
  - **Utility software**: Software that extends or customizes the capabilities of the operating system
    - Formatting
    - compress/decompress data
    - network communications
- Distinction between Application and System software can be vague.
  - Anti-trust and unfair business practice lawsuits against Microsoft have been filed over the years.

Operating System Software

- **Operating System (OS)** – Software that controls the overall operation of a computer

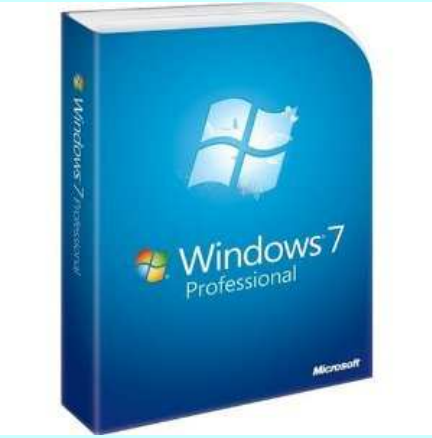


What is an operating system (OS)?

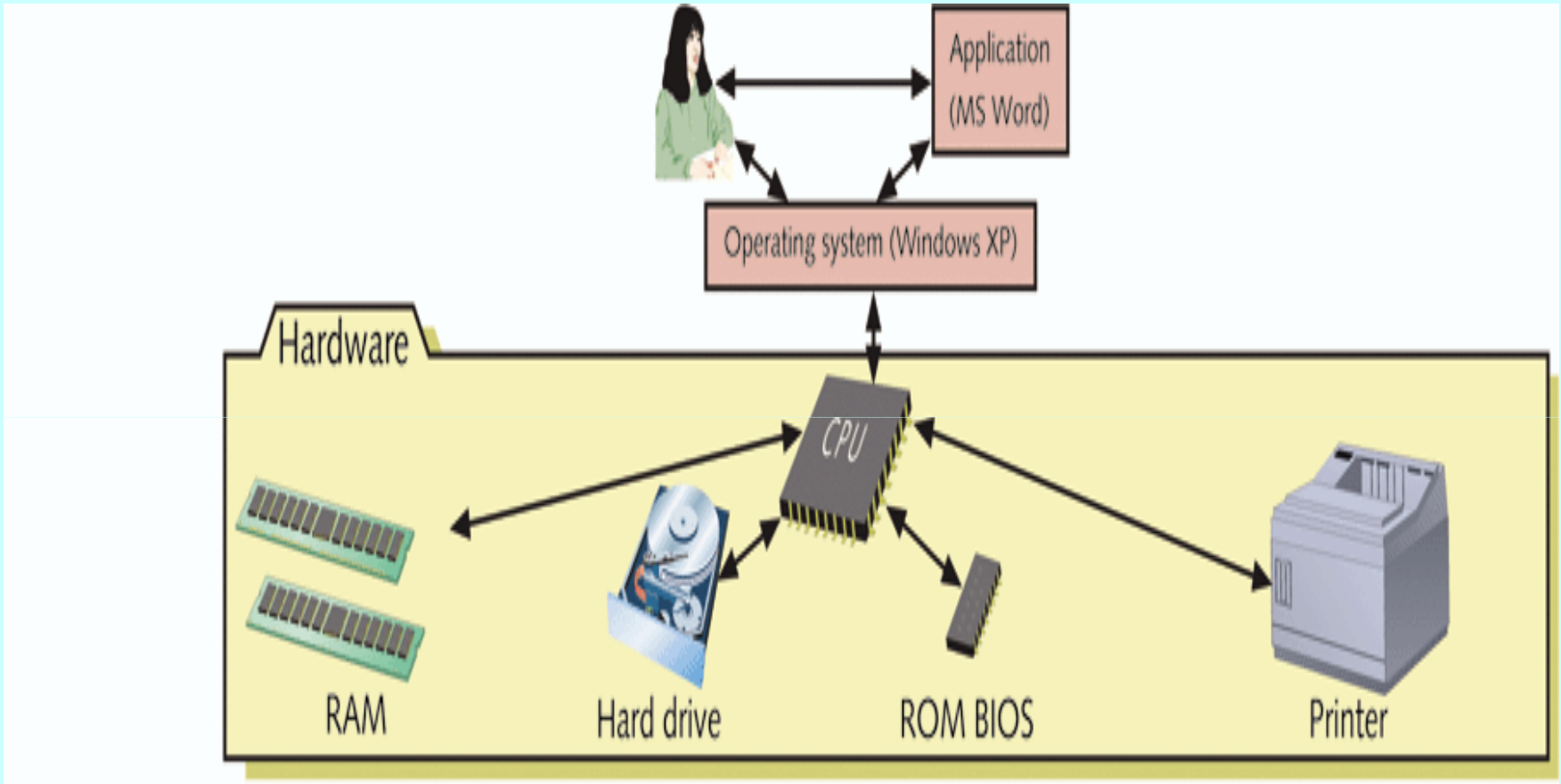
Operating System Software

Software which manages the overall operation of the computer system including:

- hardware (CPU, RAM, I/O)
- security
- system interface
- application interface



What is an operating system (OS)?



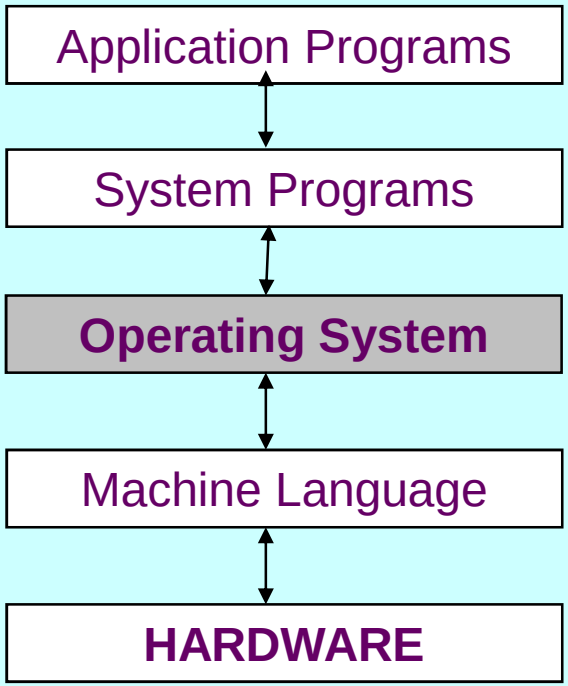
Users and applications depend on the OS to relate to all hardware components

The User's View


PHILIPS
3. 2. 2002

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use.
- Use the computer hardware in an efficient manner.



General Definition

- Since we have an already written library, namely the OS, to add two numbers we simply write the following line to our program:

```
c = a + b ;
```

General Definition

- in a system where there is no OS installed, we should consider some hardware work as:

  (Assuming an MC 6800 computer hardware)

  LDAA $80 → Loading the number at memory location 80
  LDAB $81 → Loading the number at memory location 81
  ADDB → Adding these two numbers
  STAA $55 → Storing the sum to memory location 55

- As seen, we considered memory locations and used our hardware knowledge of the system.

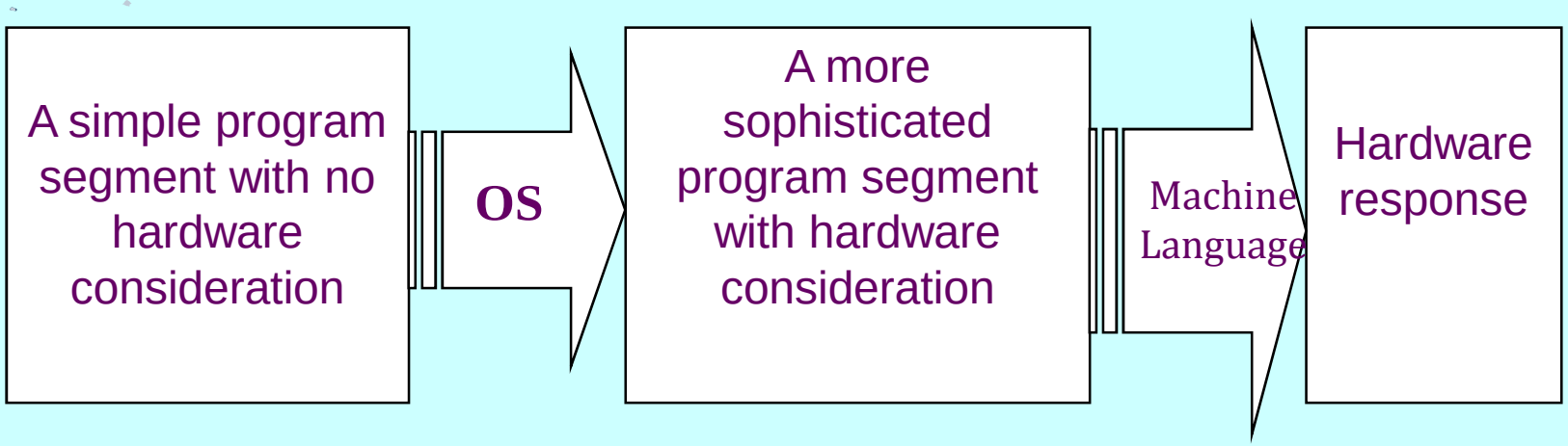
General Definition

- In an OS installed machine, since we have an intermediate layer, our programs obtain *some advantage of mobility* by not dealing with hardware.
- For example, the above program segment would not work for an 8086 machine, where as the

  “c = a + b ;”

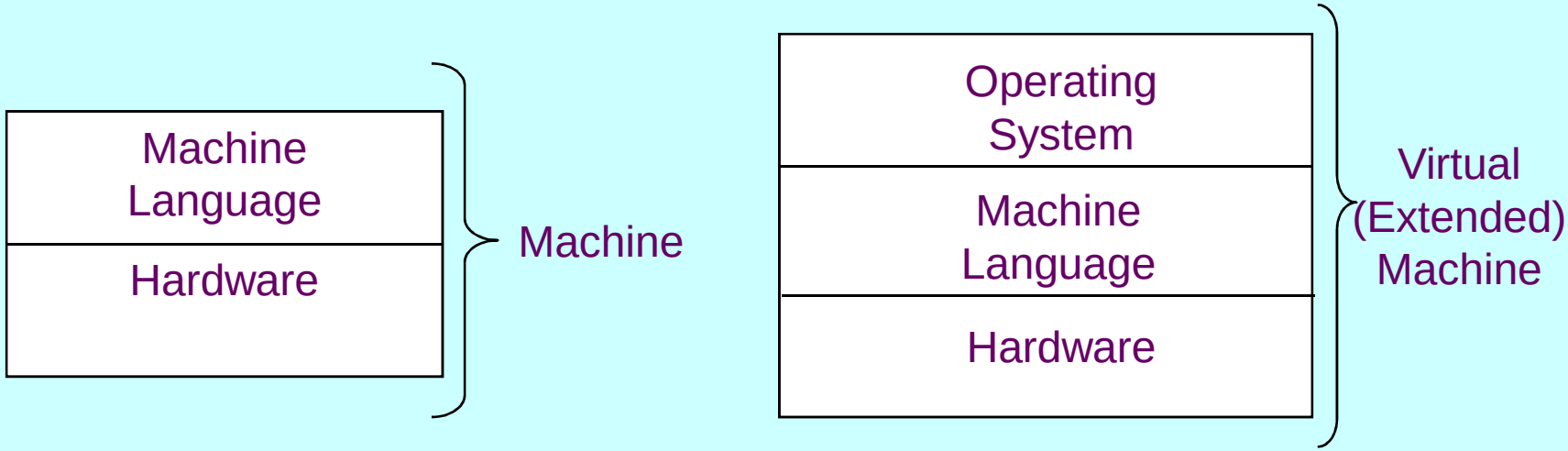
  syntax will be suitable for both.

General Definition



General Definition

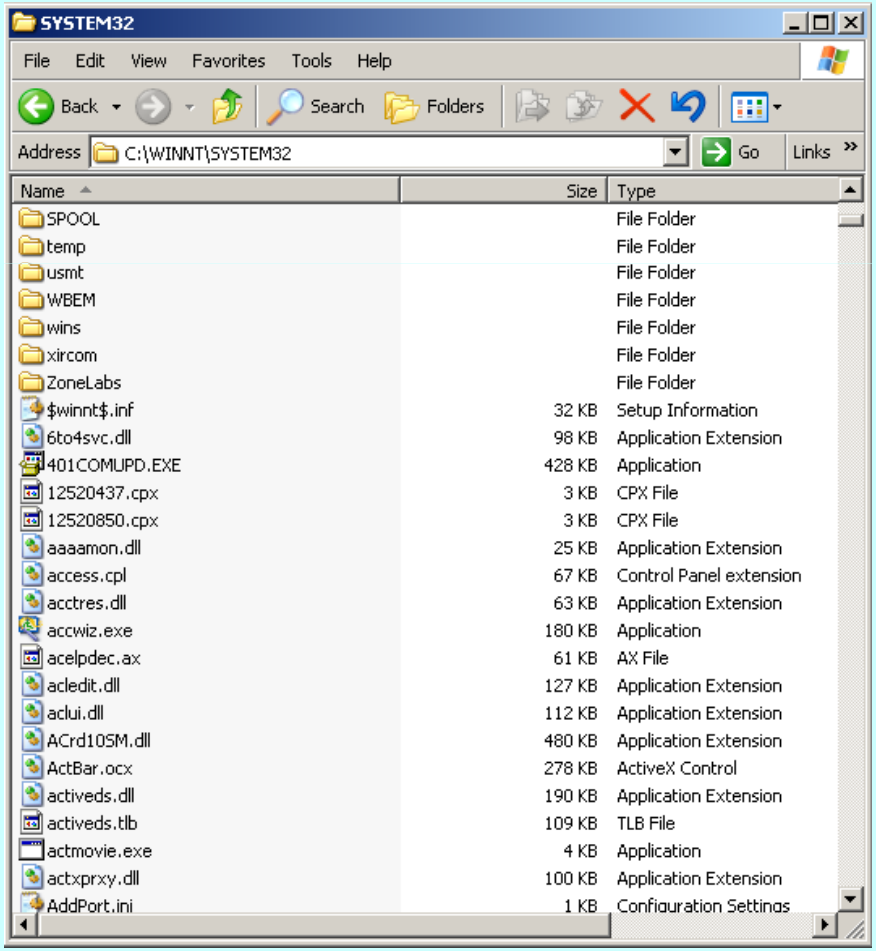
- With the advantage of easier programming provided by the OS, the hardware, its machine language and the OS constitutes a new combination called as a **virtual (extended) machine**.



What “is” an operating system?

- Software files (programs) which are stored on the hard disk
  - kernel with the internal programs
  - external programs

Supporting Data Files



The kernel

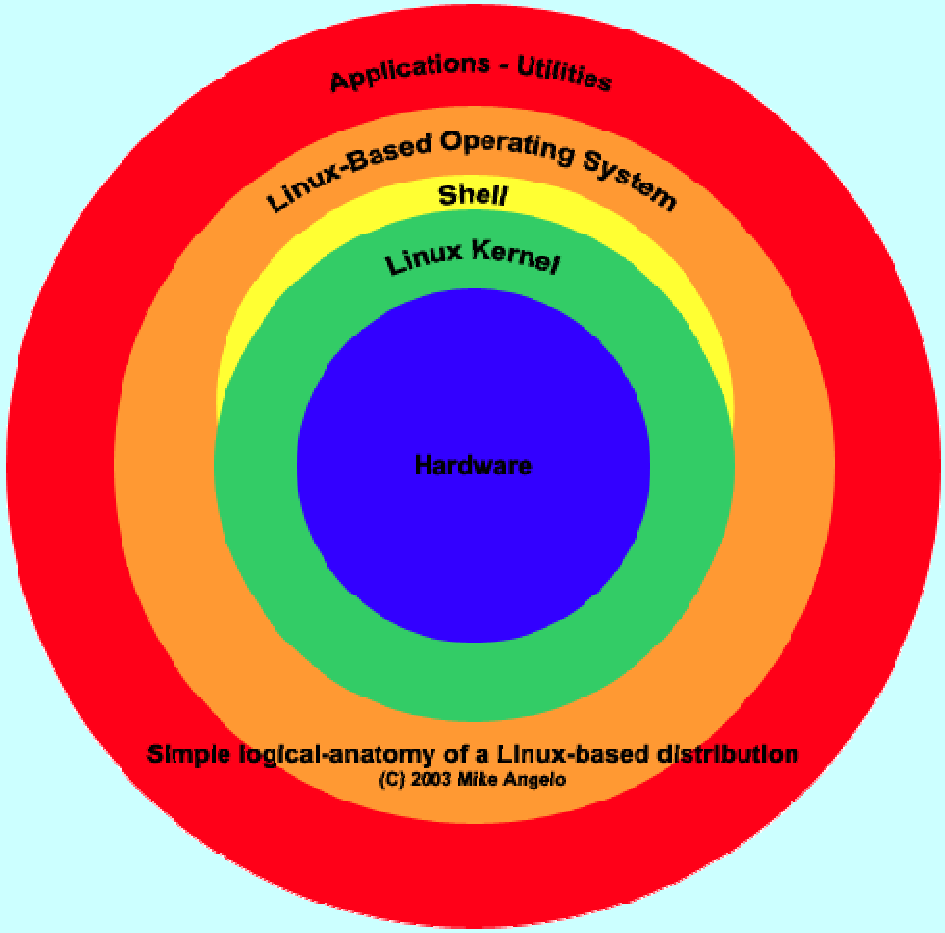
The operating system software file (program) which is **copied into RAM**, usually from the hard disk drive, during the **boot-up**.

The kernel remains in RAM while the computer is on and is **in charge of the overall operation** of the computer system.

The kernel contains the "**internal programs**" for the most often used operations like copying files.

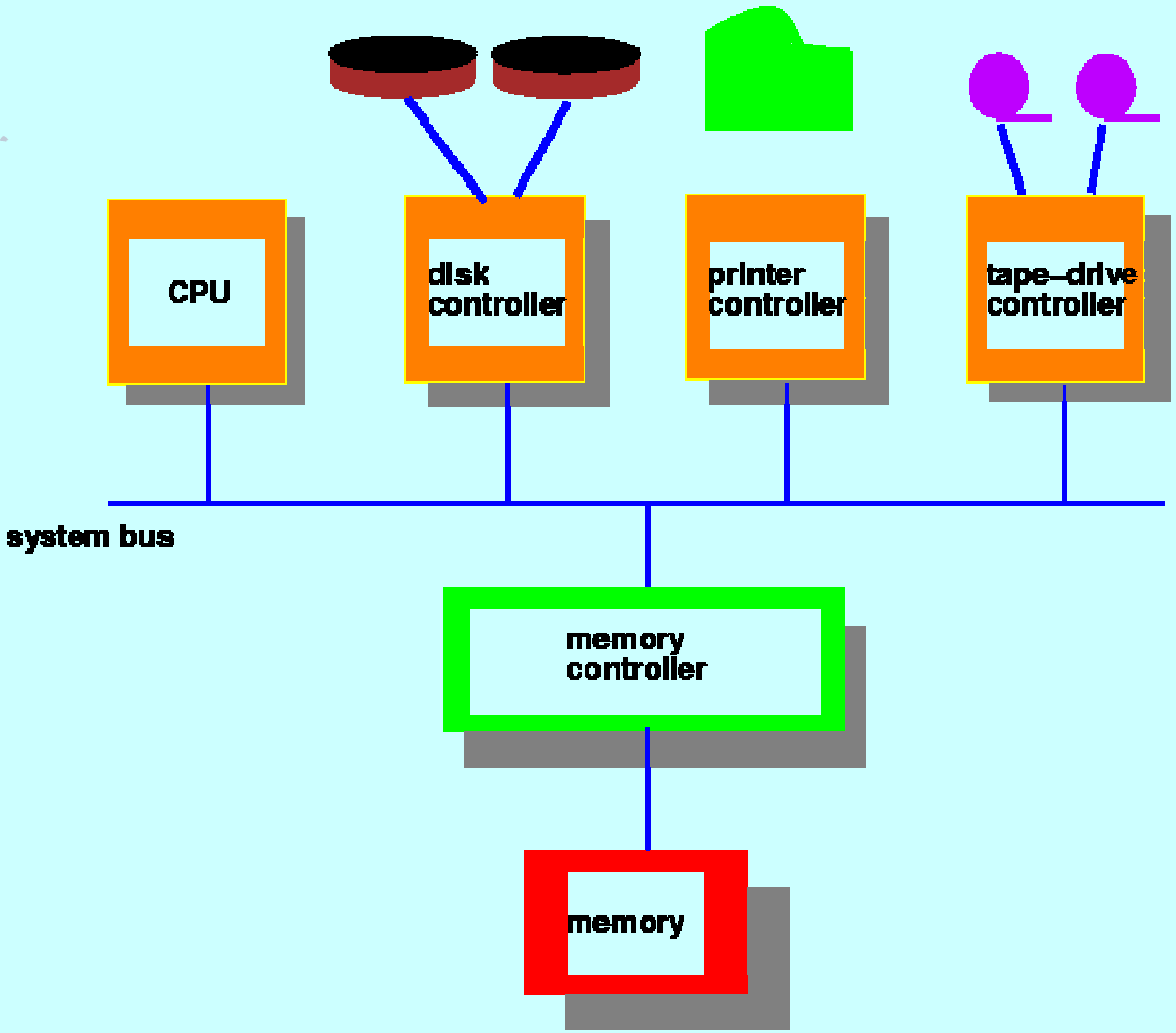
- kmem (Linux)
- command.exe (Microsoft)

Kernel



- **Kernel** – The internal part of the operating system.
  - Those software components that perform the basic functions required by the computer.
    - File management
    - Memory management (RAM)
    - Security

Computer System Architecture



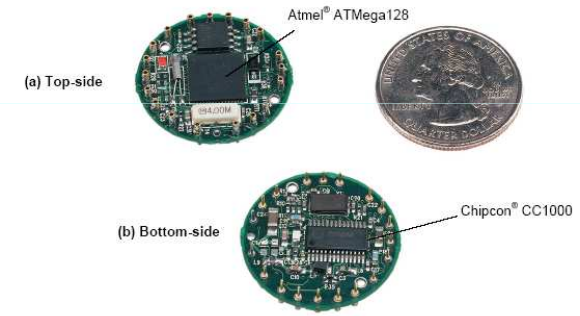
Goals of an Operating System

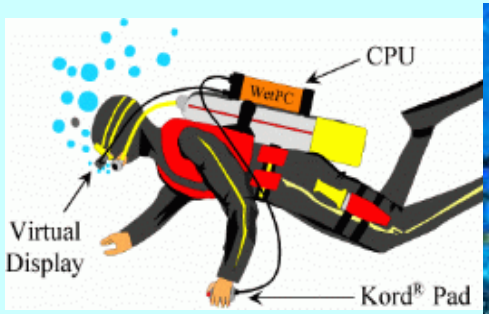
- Simplify the execution of user programs and make solving user problems easier.
- Use computer hardware efficiently.
  - Allow sharing of hardware and software resources.
- Make application software portable and versatile.
- Provide isolation, security and protection among user programs.
- Improve overall system reliability
    - error confinement, fault tolerance, reconfiguration.

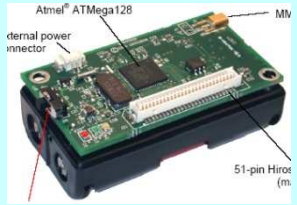
Why should I study Operating Systems?

- Need to understand interaction between the hardware and applications
  - New applications, new hardware..
  - Inherent aspect of society today
- Need to understand basic principles in the design of computer systems
  - efficient resource management, security, flexibility
- Increasing need for specialized operating systems
  - e.g. embedded operating systems for devices - cell phones, sensors and controllers
  - real-time operating systems - aircraft control, multimedia services

Systems Today



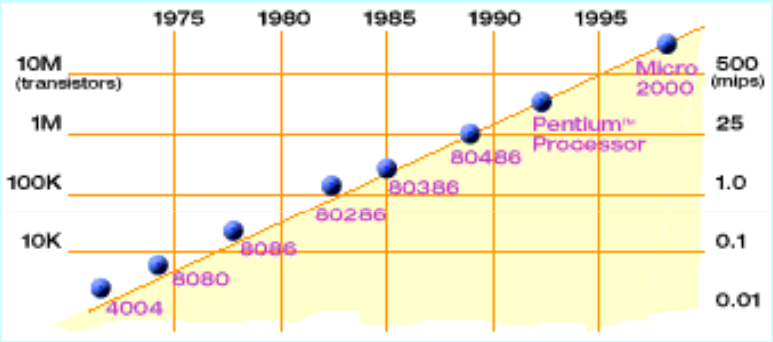


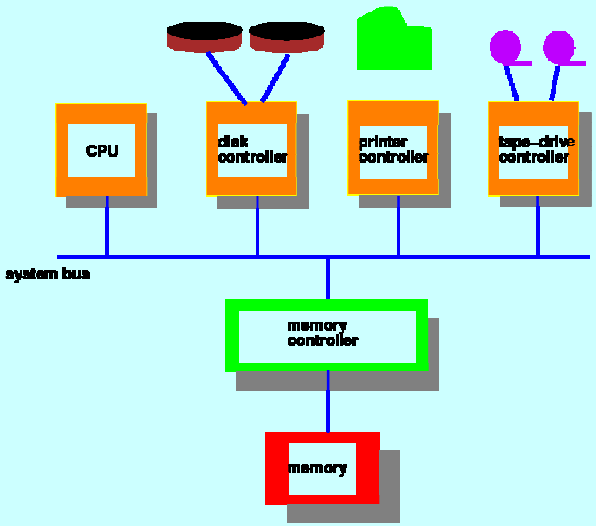


Irvine Sensorium

Hardware Complexity Increases

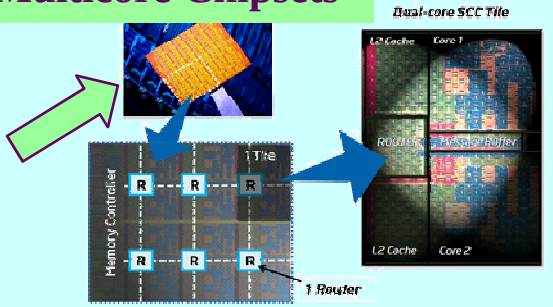
Moore's Law: 2X transistors/Chip Every 1.5 years

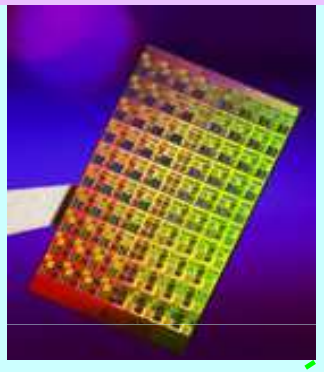


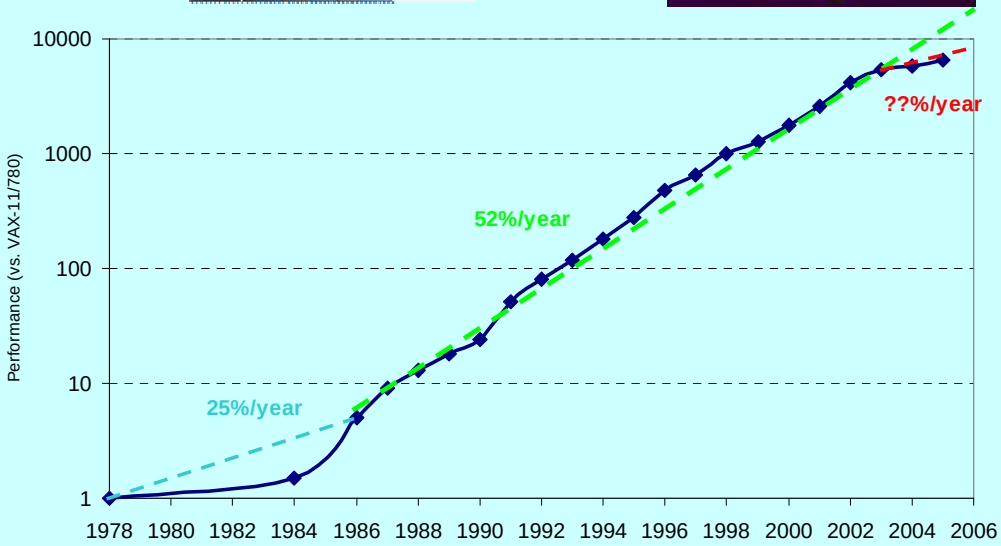


From Berkeley OS course

Intel Multicore Chipsets

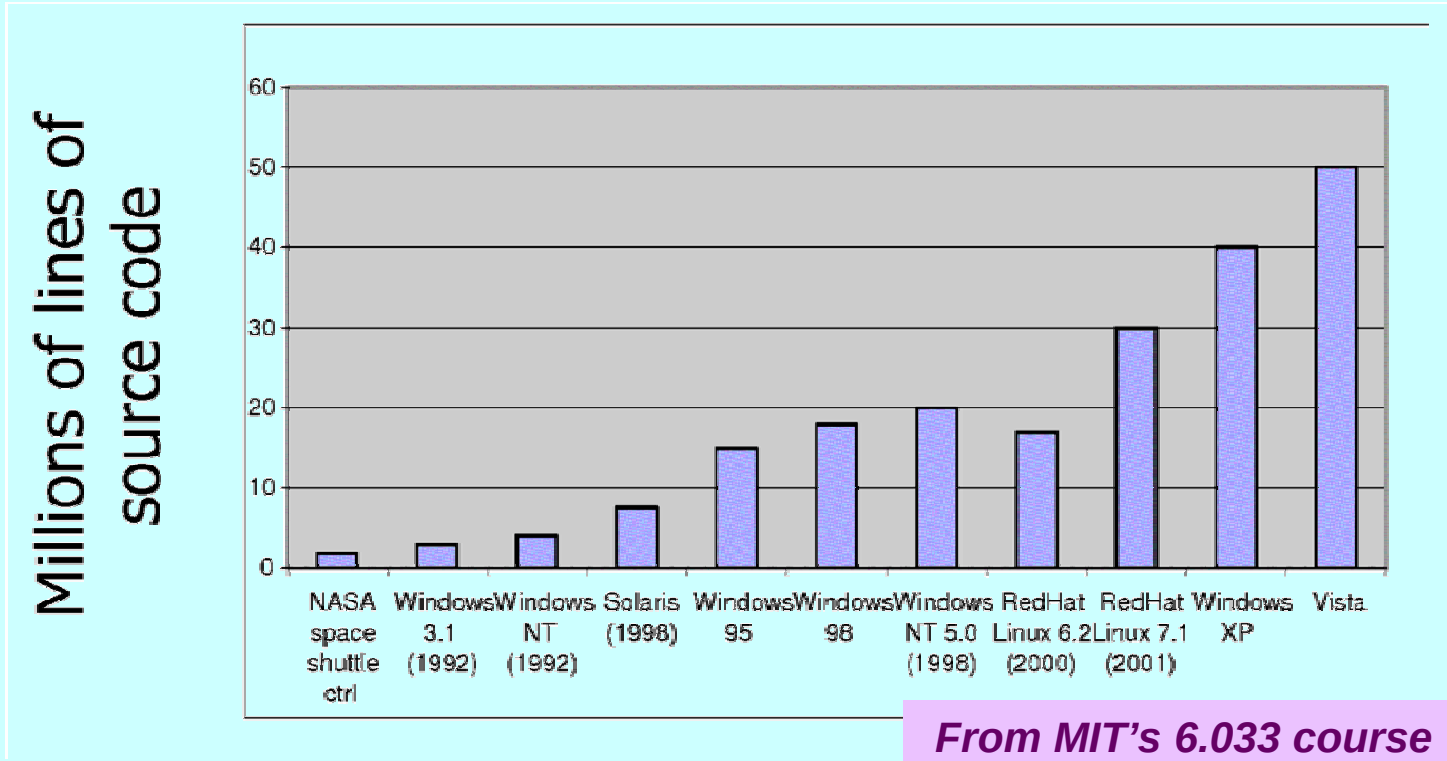






From Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*, 4th edition, Sept. 15, 2006

Software Complexity Increases



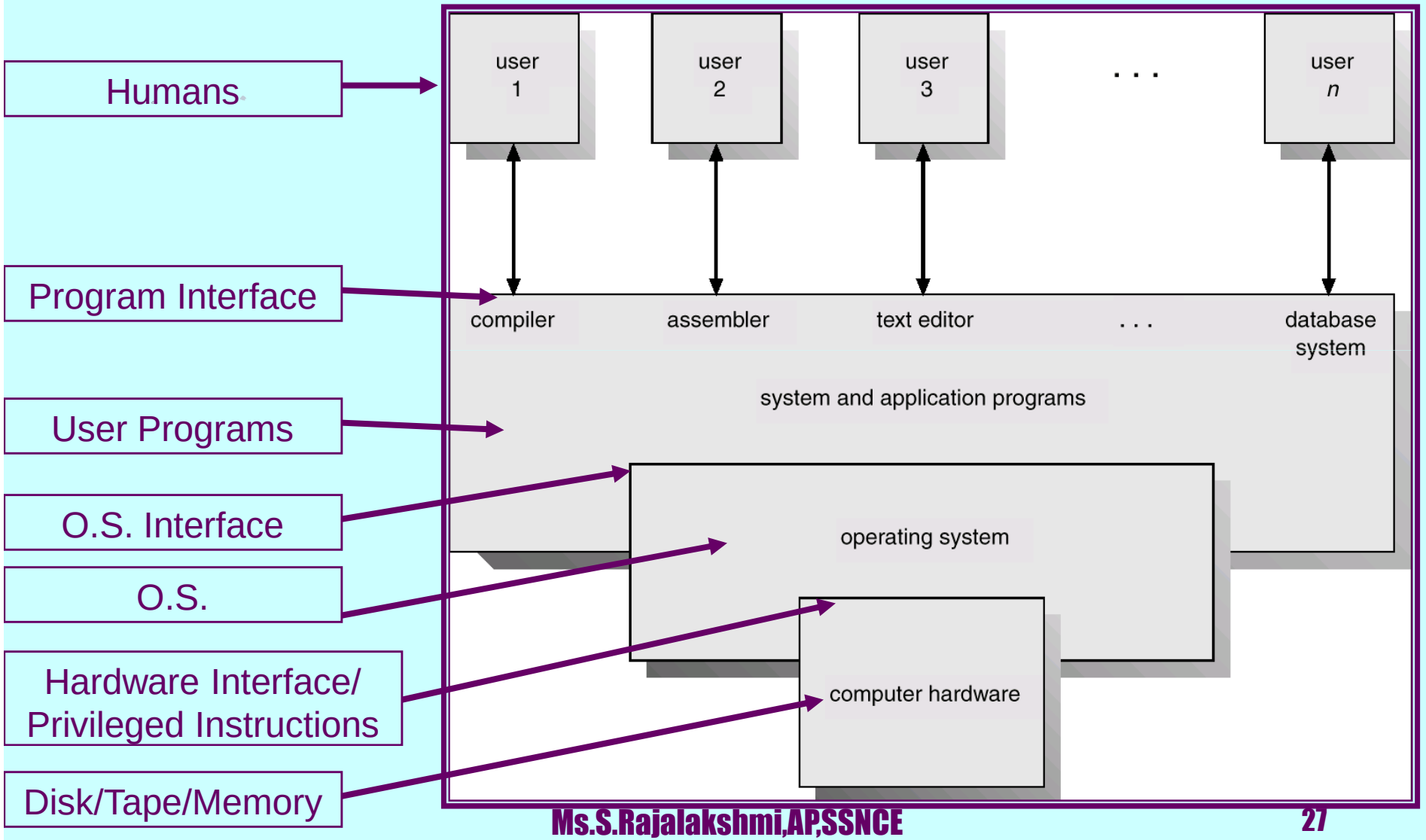
From MIT's 6.033 course

Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices).
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, machines, other computers).

Abstract View of System Components

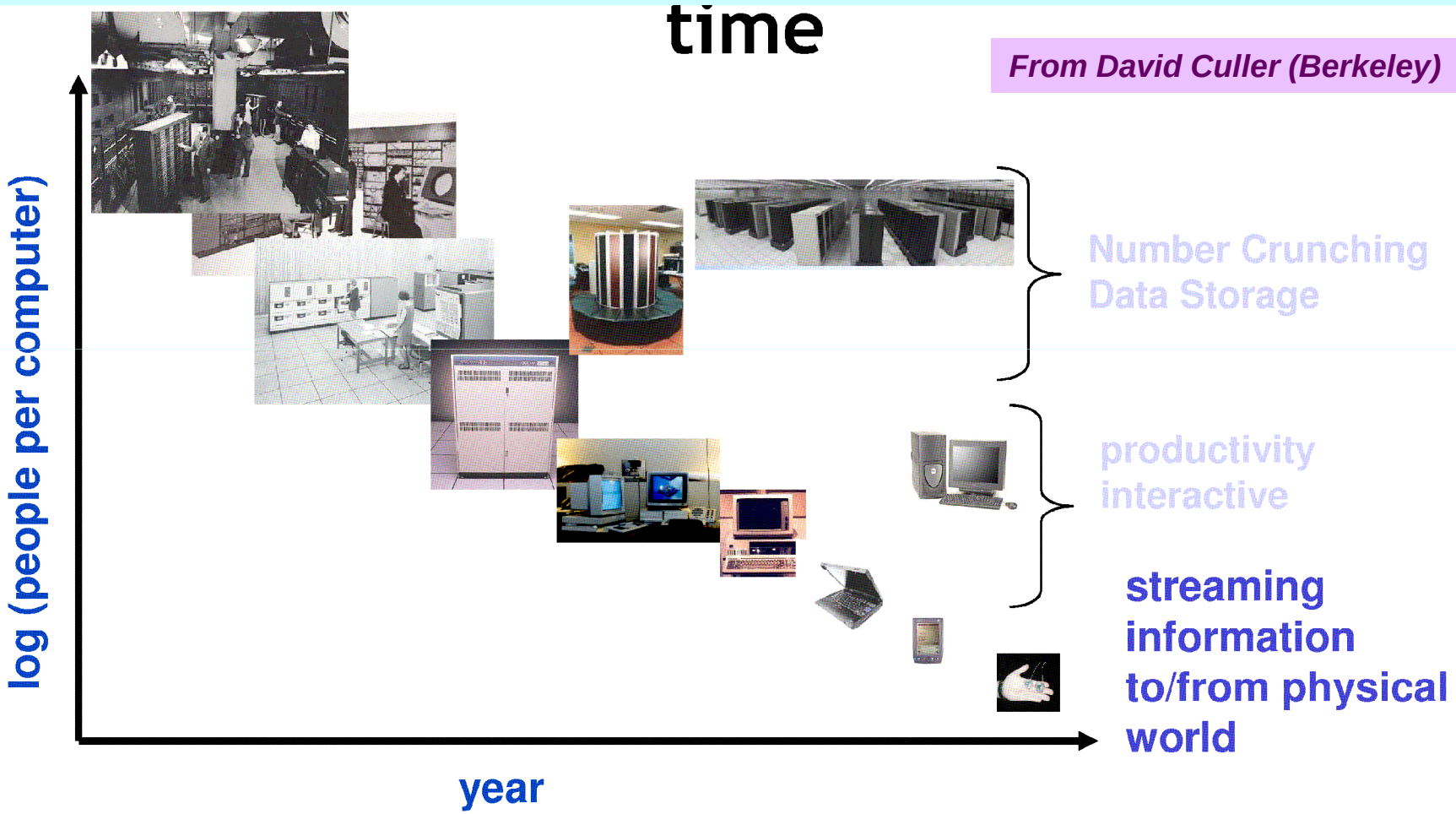
The Layers Of A System



Operating System Definitions

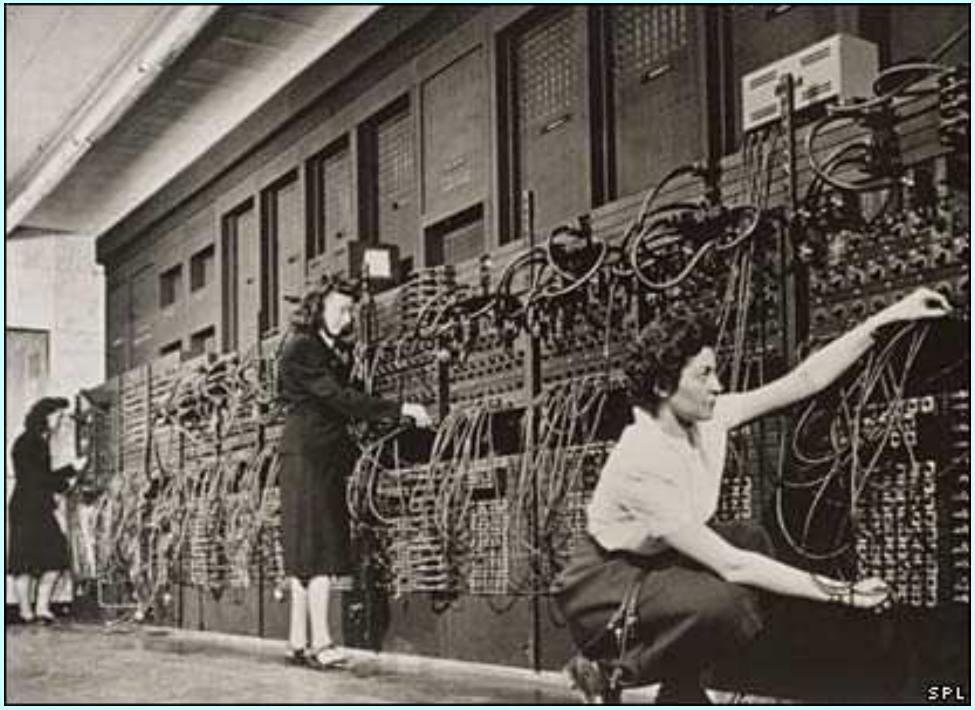
- Resource allocator – manages and allocates resources.
- Control program – controls the execution of user programs and operations of I/O devices .
- Kernel – the one program running at all times (all else being application programs).

People-to-Computer Ratio Over Time



History of Operating Systems

- It all started with computer hardware in about 1940s.



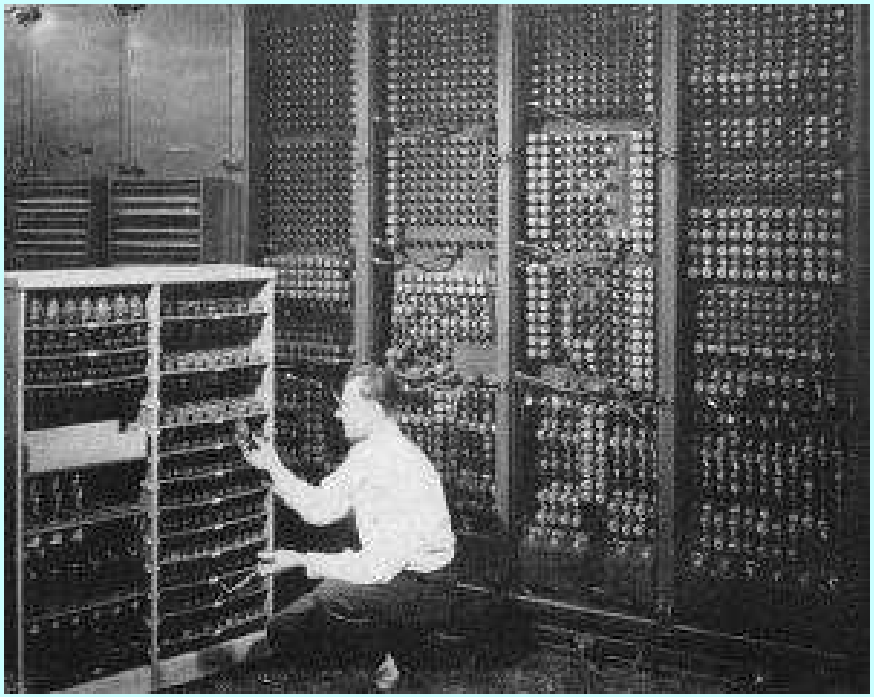
ENIAC 1943

History of Operating Systems

- Computers were using vacuum tube technology.

ENIAC’s vacuum tubes

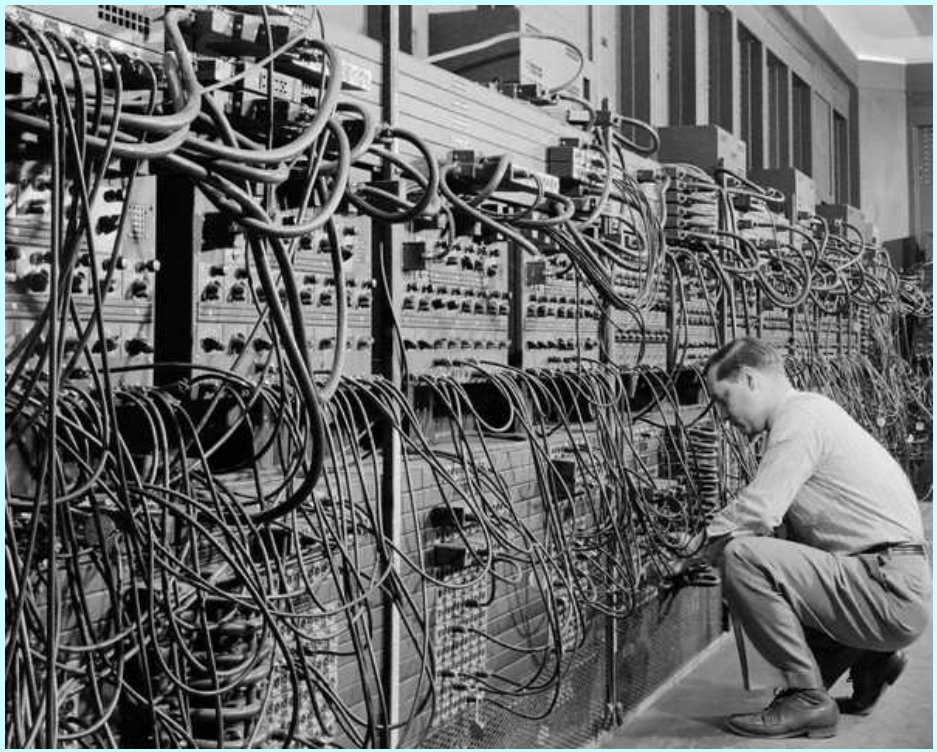
History of Operating Systems



ENIAC’s backside

History of Operating Systems

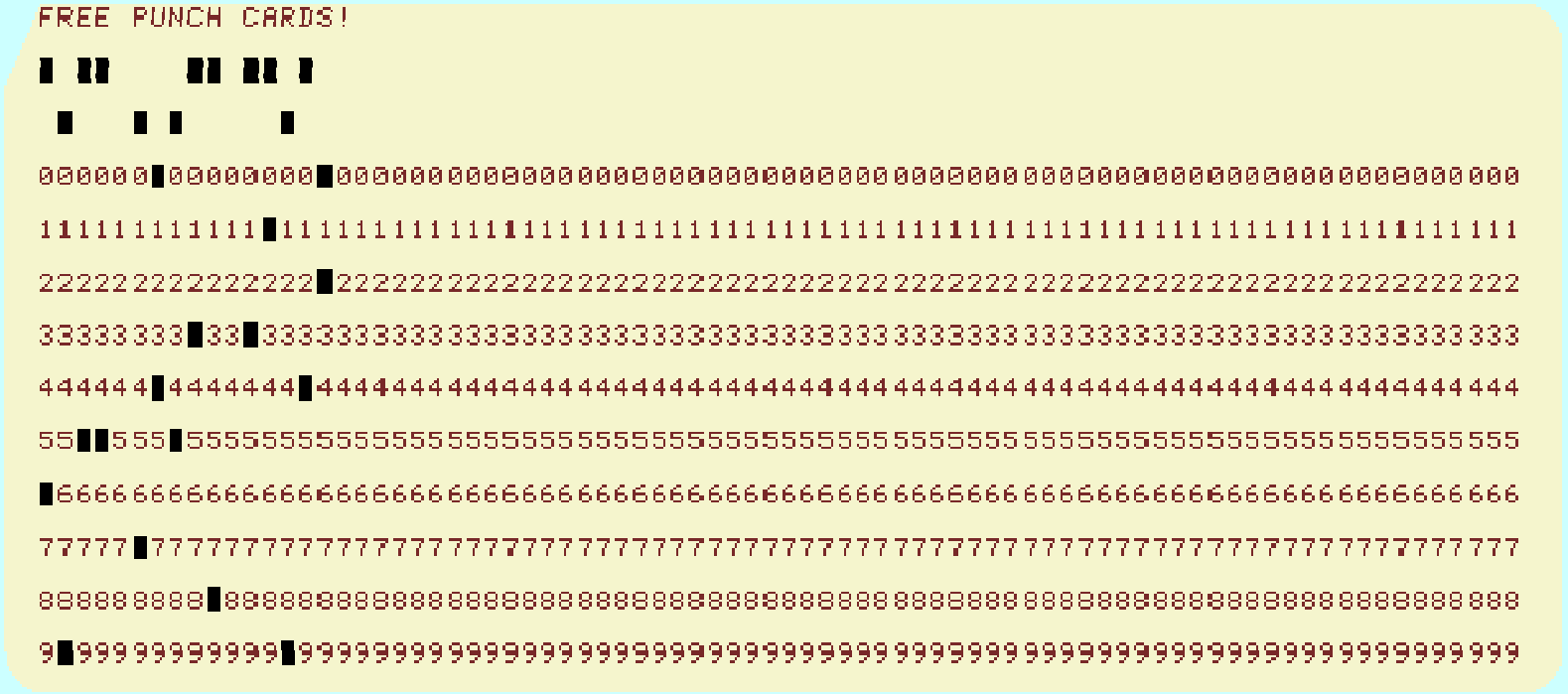
Programs were loaded into memory manually using switches, punched cards, or paper tapes.



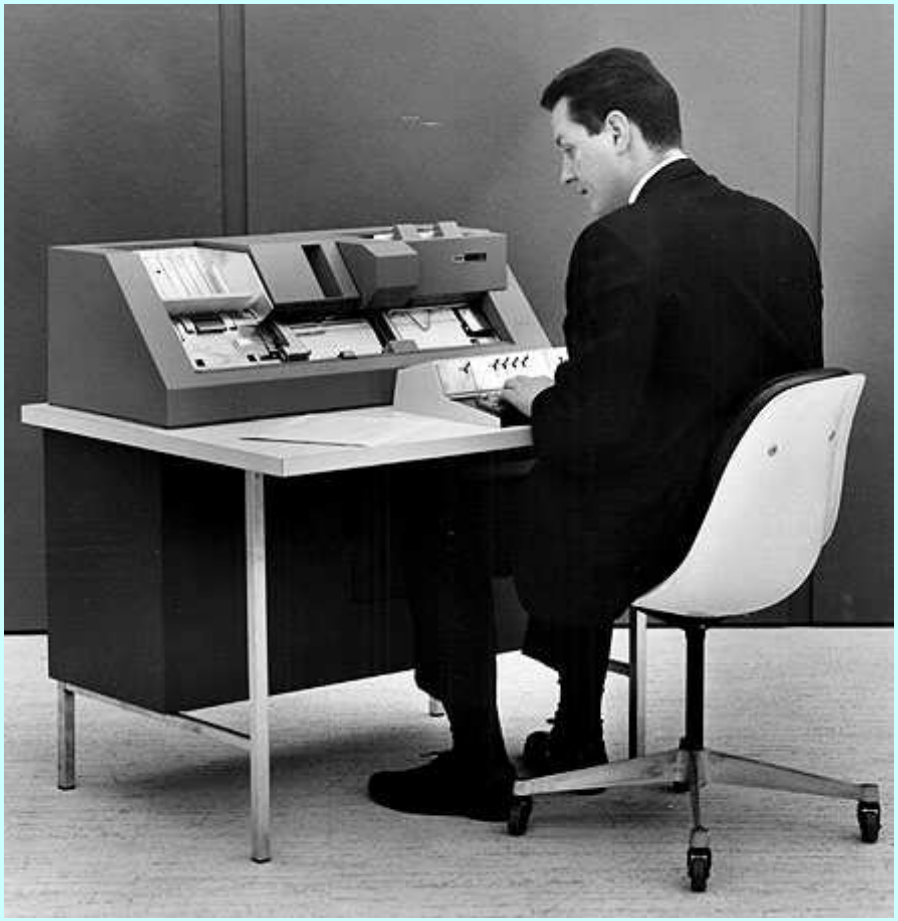
ENIAC : coding by cable connections

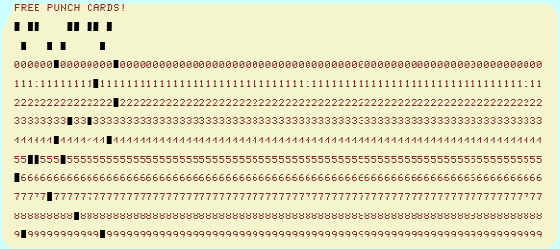
History of Operating Systems

punch card

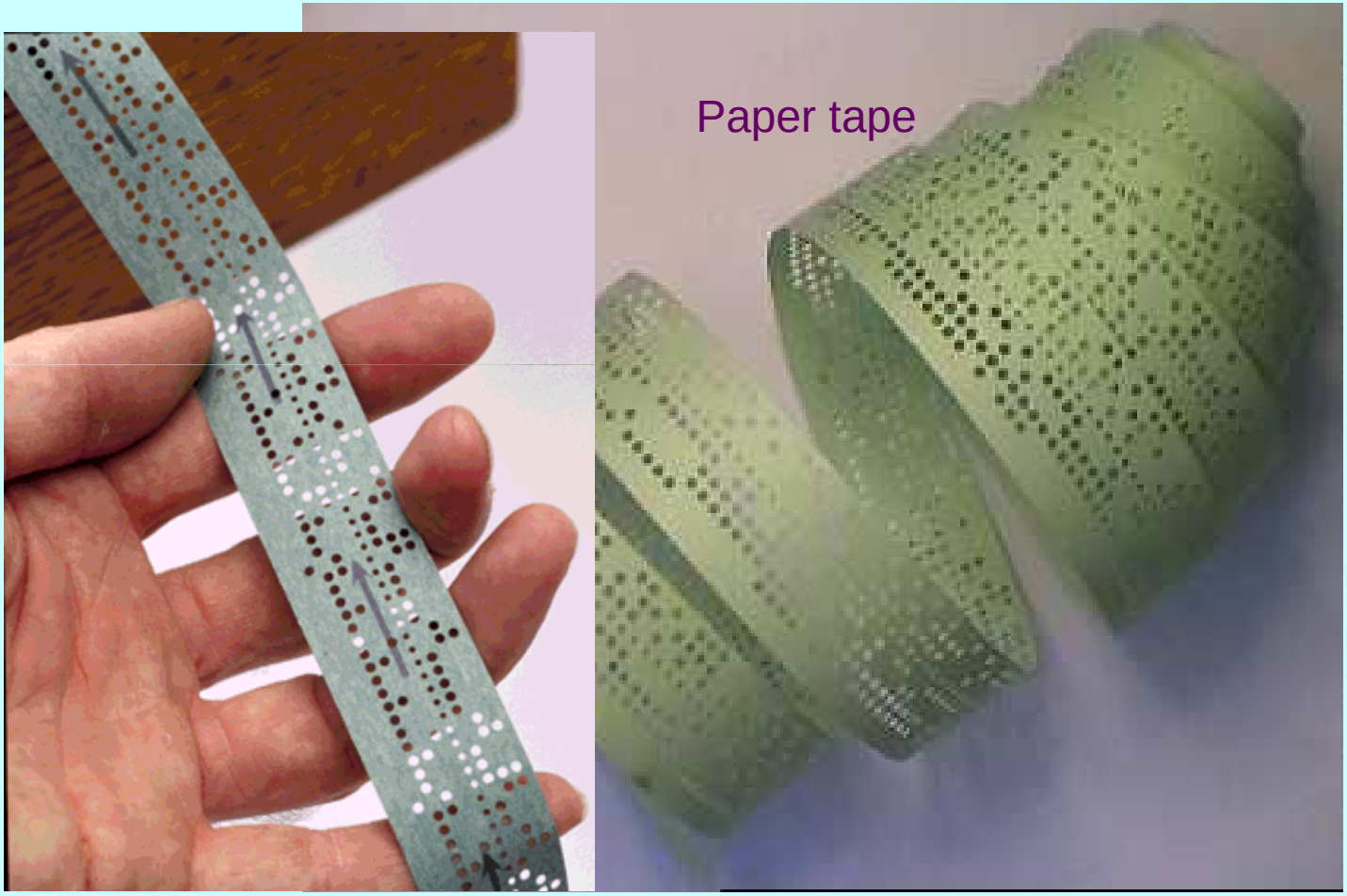


History of Operating Systems

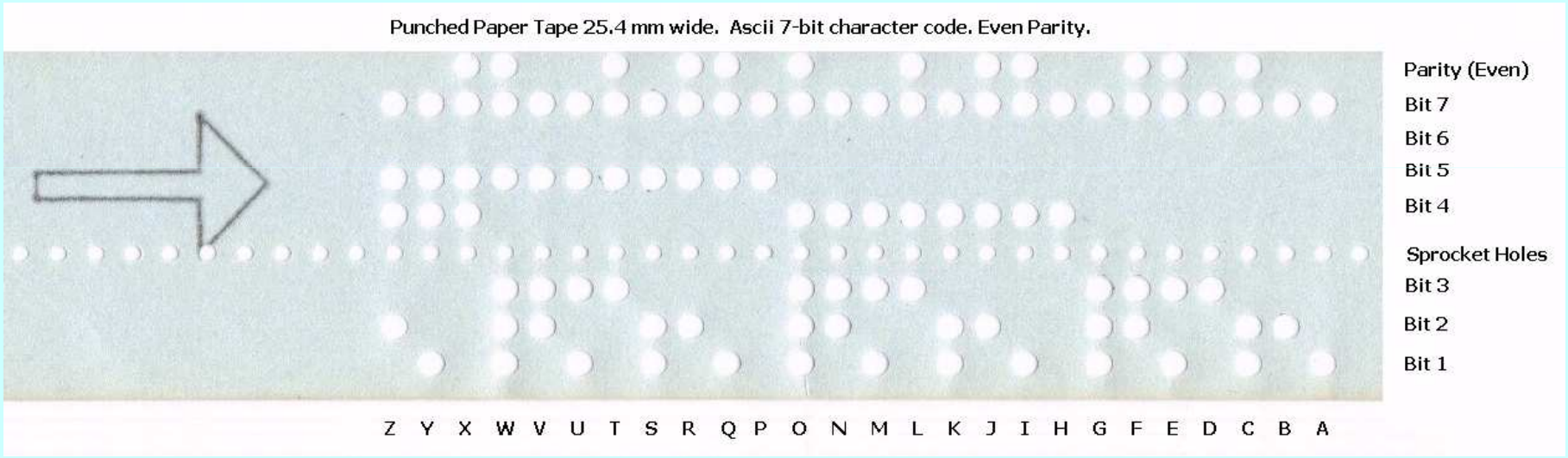




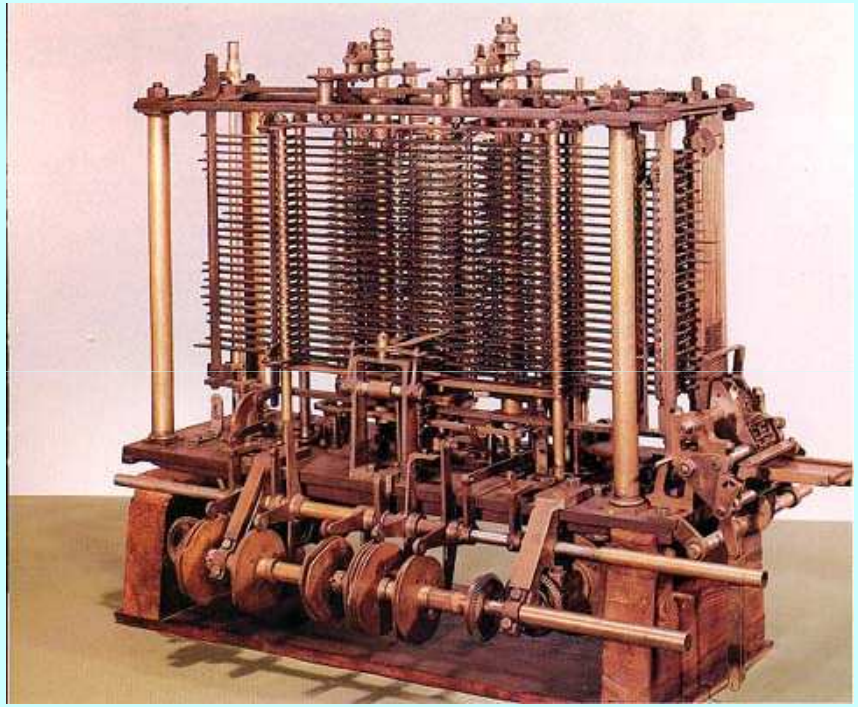
History of Operating Systems



History of Operating Systems



History of Operating Systems

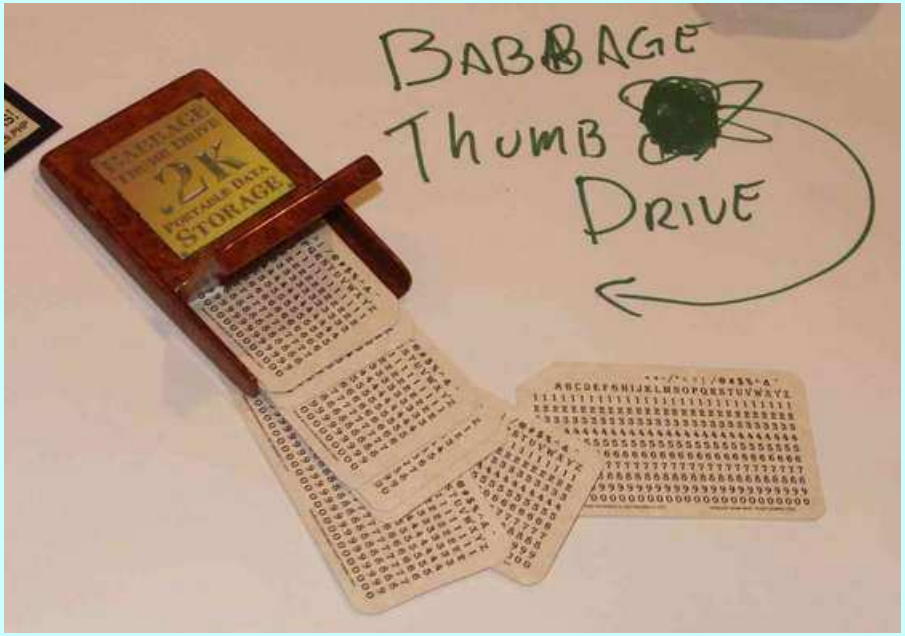


Babbage's analytical engine
(designed in 1840's by Charles Babbage, but cold not be constructed by him.
An earlier and simpler version is constructed in 2002, in London)

http://www.computerhistory.org/babbage/

History of Operating Systems

- Ada Lovalence (at time of Charles Babbage) wrote code for analytical engine to compute Bernulli Numbers



Early Systems - Bare Machine (1950s)

Hardware – ***expensive*** ; **Human** – ***cheap***

- Structure
  - Large machines run from console
  - Single user system
    - Programmer/User as operator
  - Paper tape or punched cards

From John Ousterhout slides

- Early software
  - Assemblers, compilers, linkers, loaders, device drivers, libraries of common subroutines.
- Secure execution
- Inefficient use of expensive resources
  - Low CPU utilization, high setup time.

Simple Batch Systems (1960's)

- Reduce setup time by batching jobs with similar requirements.
- Add a card reader, Hire an operator
  - User is NOT the operator
  - Automatic job sequencing
    - Forms a rudimentary OS.
  - Resident Monitor
    - Holds initial control, control transfers to job and then back to monitor.
  - Problem
    - Need to distinguish job from job and data from program.

From John Ousterhout slides

Supervisor/Operator Control

- Secure monitor that controls job processing
  - Special cards indicate what to do.
  - User program prevented from performing I/O
- Separate user from computer
  - User submits card deck
  - cards put on tape
  - tape processed by operator
  - output written to tape
  - tape printed on printer
- Problems
  - Long turnaround time - up to 2 DAYS
  - Low CPU utilization
    - I/O and CPU could not overlap; slow mechanical devices.

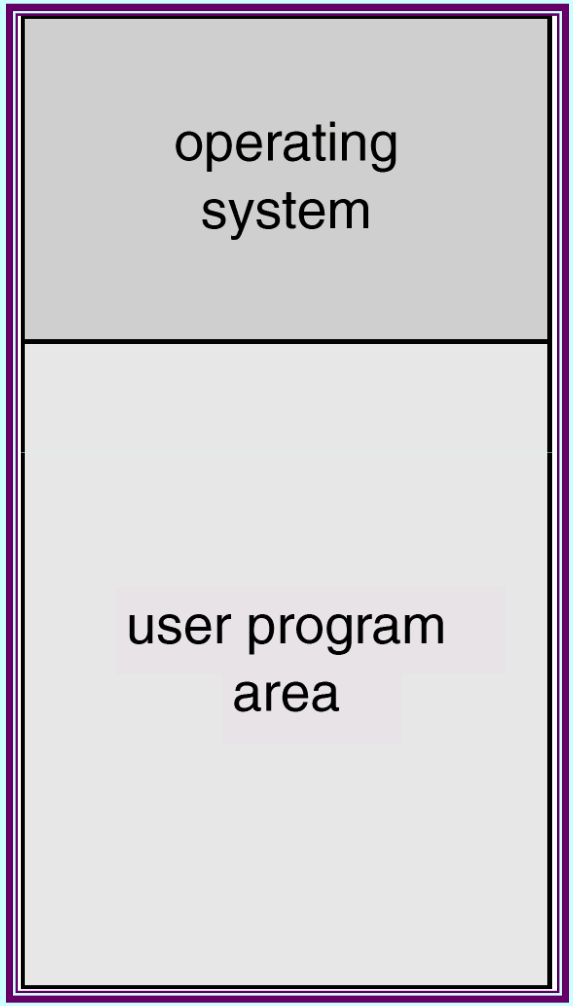
IBM 7094

From John Ousterhout slides

Mainframe Systems

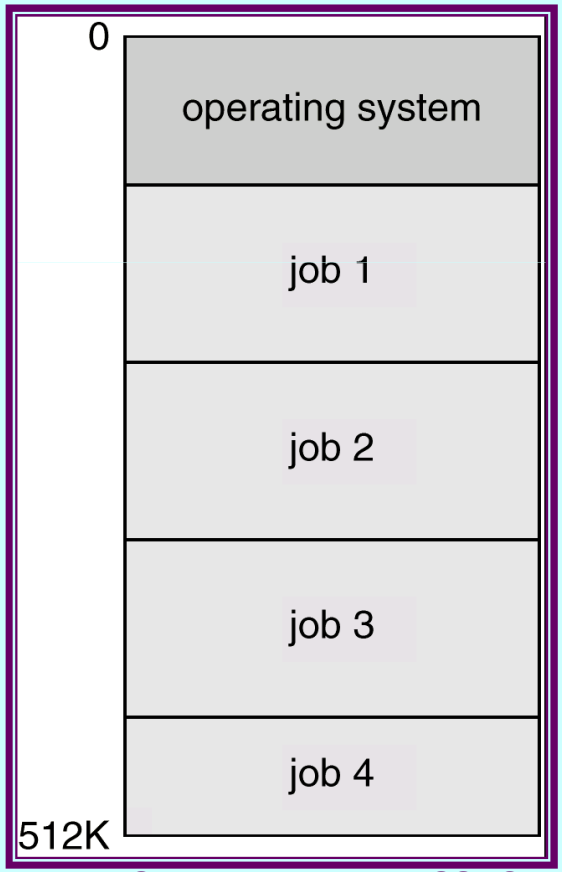
- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers pack to monitor

Memory Layout for a Simple Batch System



Multiprogrammed Batch Systems

Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.



OS Features Needed for Multiprogramming

- I/O routine supplied by the system.
- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling – the system must choose among several jobs ready to run.
- Allocation of devices.

Time-Sharing Systems–Interactive Computing

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
- A job swapped in and out of memory to the disk.
- On-line communication between the user and the system is provided; when the operating system finishes the execution of one command, it seeks the next “control statement” from the user’s keyboard.
- On-line system must be available for users to access data and code.

Desktop Systems

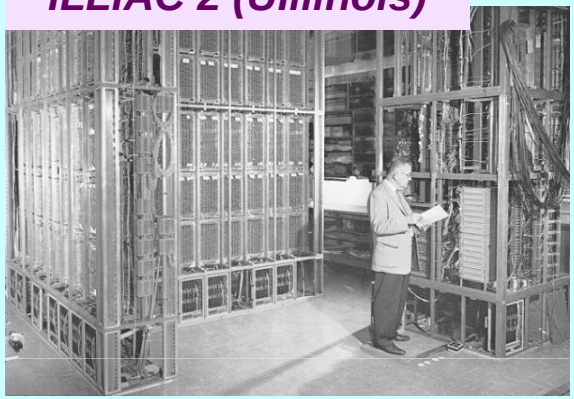
- *Personal computers* – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system' often individuals have sole use of computer and do not need advanced CPU utilization of protection features.
- May run several different types of operating systems (Windows, MacOS, UNIX, Linux)

Parallel Systems

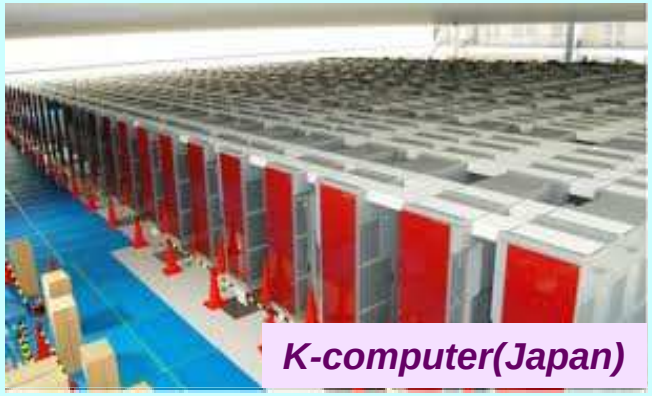
- Multiprocessor systems with more than on CPU in close communication.
- *Tightly coupled system* – processors share memory and a clock; communication usually takes place through the shared memory.
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - graceful degradation
    - fail-soft systems

Parallel Computing Systems

ILLIAC 2 (Uilllinois)



Climate modeling, earthquake simulations, genome analysis, protein folding, nuclear fusion research, …..



K-computer(Japan)

Connection Machine (MIT)

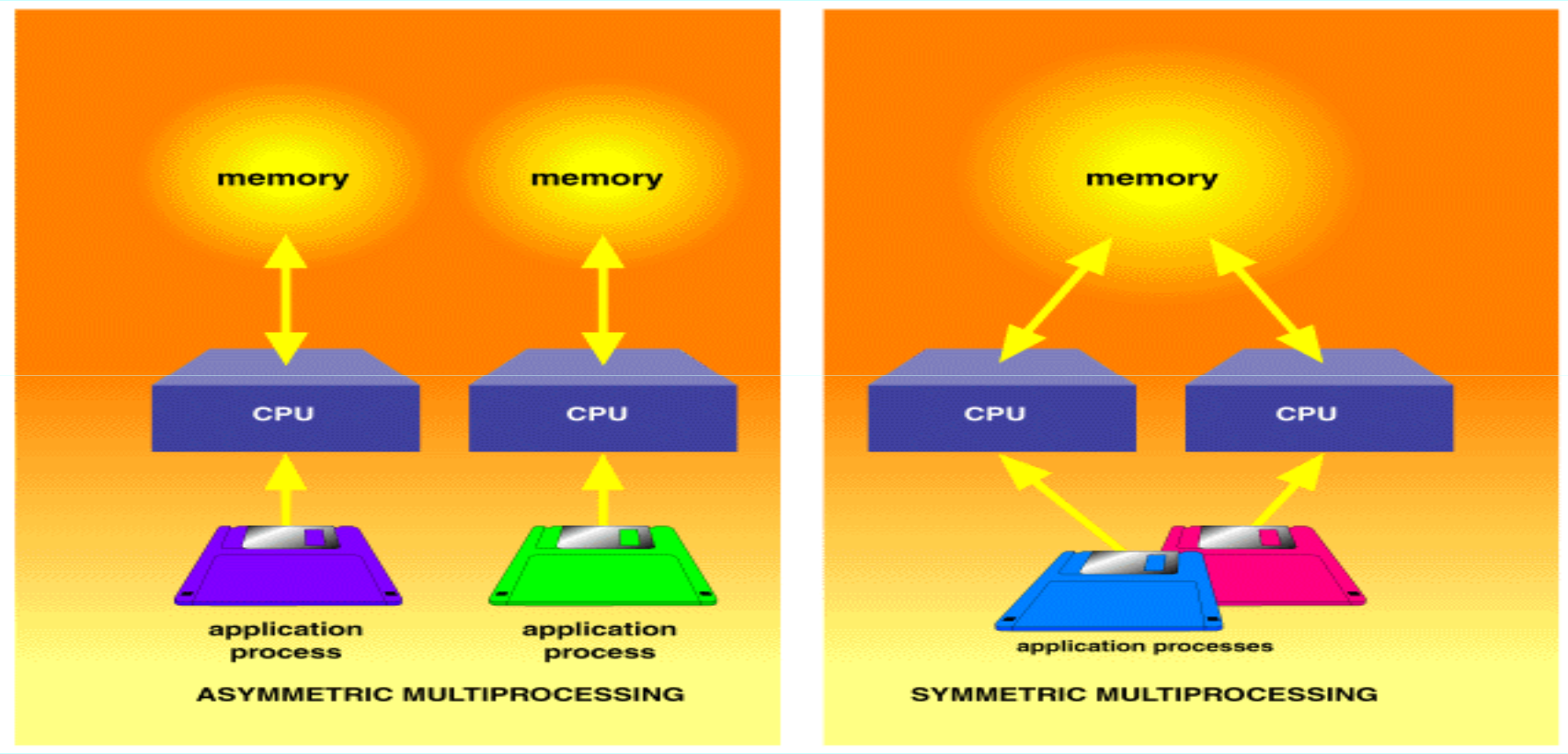
Tianhe-1(China)

IBM Blue Gene

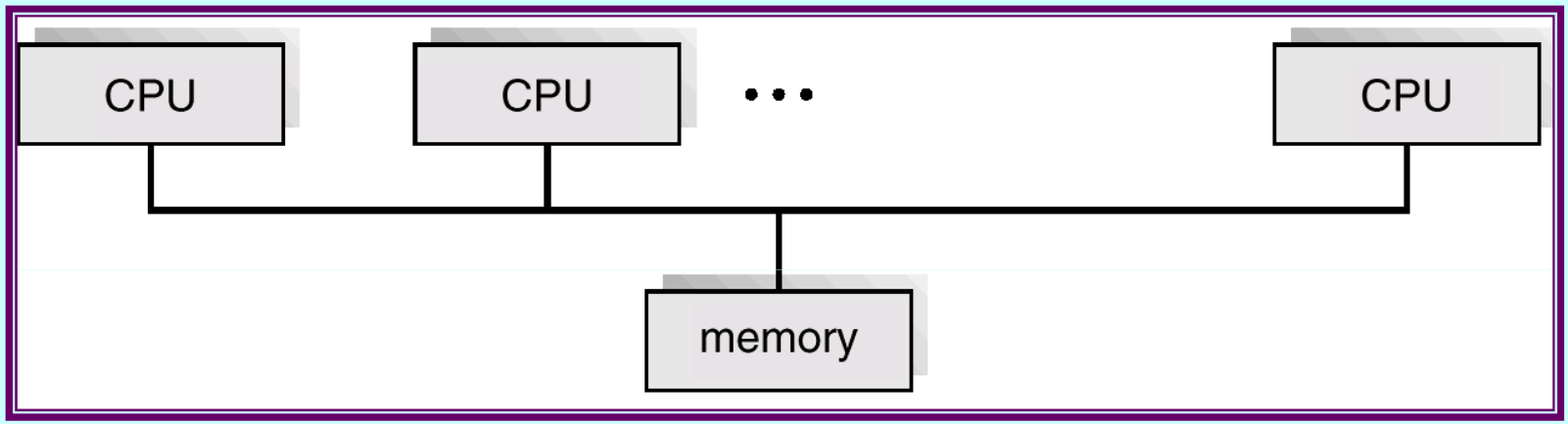
Parallel Systems (Cont.)

- *Symmetric multiprocessing (SMP)*
  - Each processor runs and identical copy of the operating system.
  - Many processes can run at once without performance deterioration.
  - Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
  - More common in extremely large systems

Multiprocessing with Multiple CPU's



Symmetric Multiprocessing Architecture



Distributed Systems

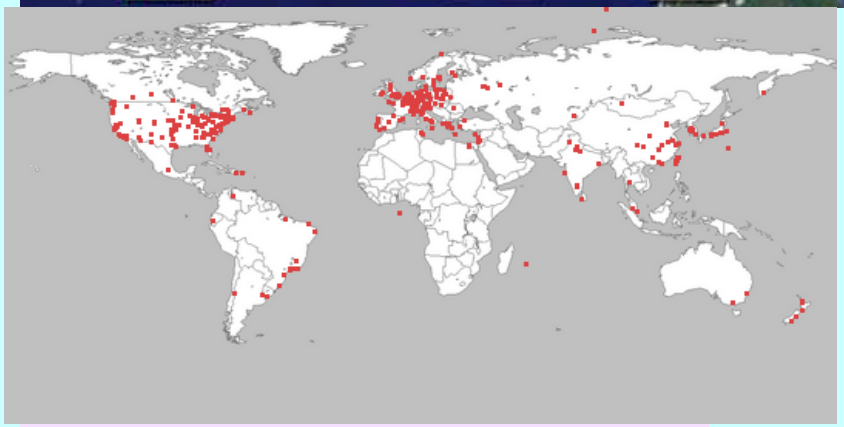
- Distribute the computation among several physical processors.
- *Loosely coupled system* – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.
- Advantages of distributed systems.
  - Resources Sharing
  - Computation speed up – load sharing
  - Reliability
  - Communications

Distributed Systems (cont)

- Requires networking infrastructure.
- Local area networks (LAN) or Wide area networks (WAN)
- May be either client-server or peer-to-peer systems.

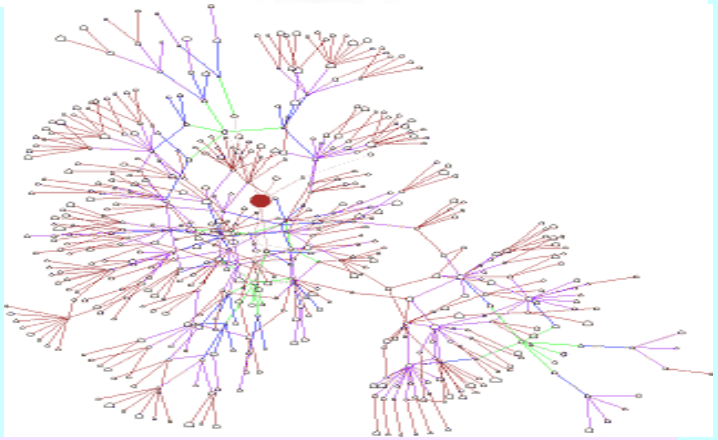
Distributed Computing Systems

Globus Grid Computing Toolkit



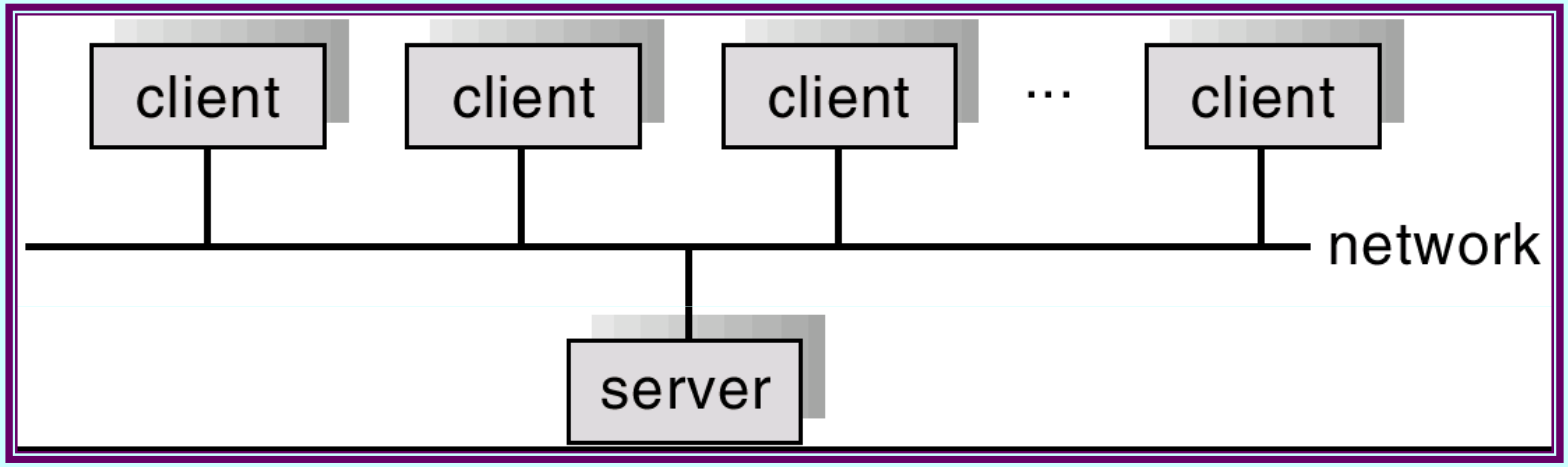
PlanetLab

Cloud Computing Offerings



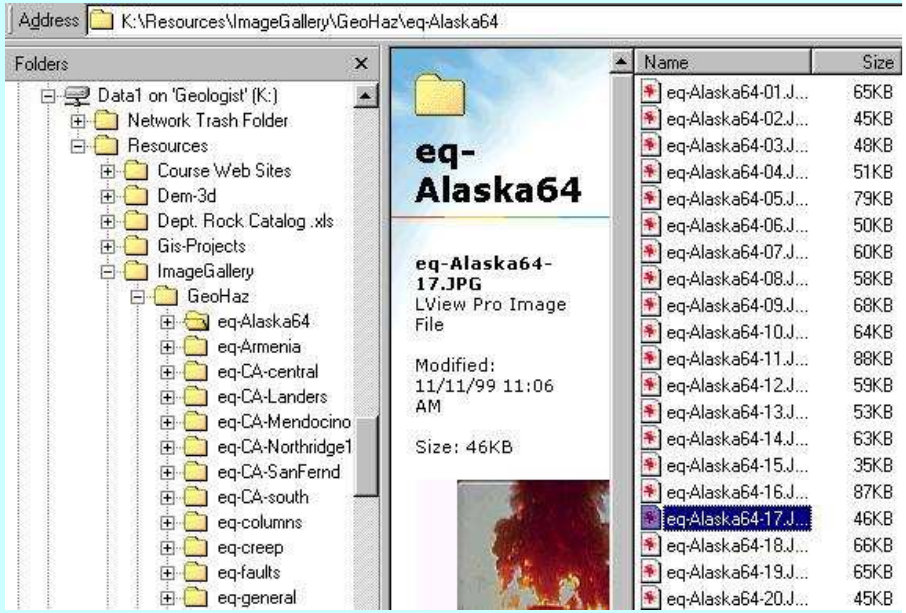
Gnutella P2P Network

General Structure of Client-Server



Server (Multiuser) Operating Systems

- Found on mainframes, minicomputers and PCs
- **server** = a computer which processes information (CPU and RAM) , stores information (hard disk), and/or provides access to peripheral devices (printers) for multiple users
  - Email
  - Web
  - Gaming



Examples of Server Operating Systems

- MAC OS X Server
- Windows Server

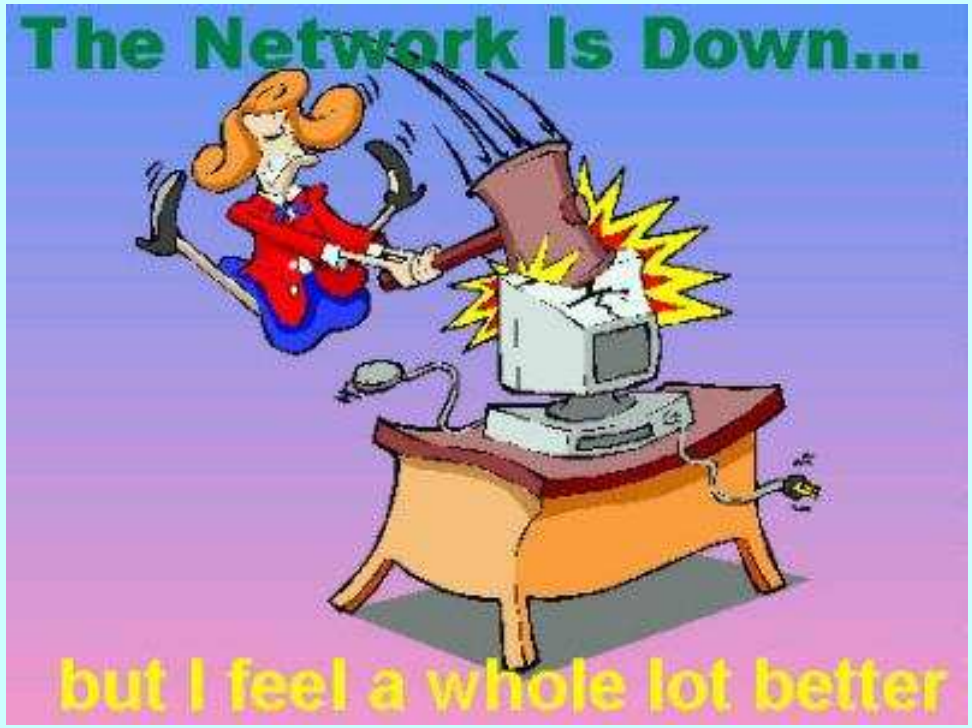
Advantages/Disadvantages of Server Operating Systems

Advantages

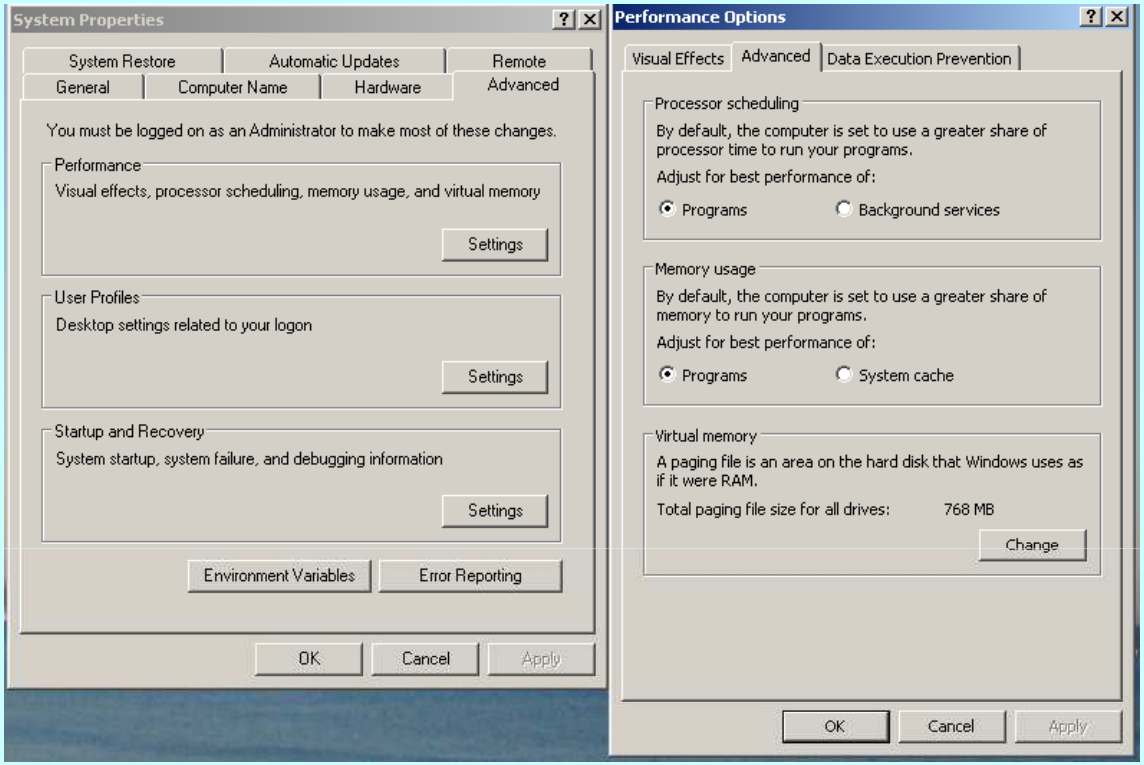
- Central location for the installation and administration of all software and data
- More cost effective - less expensive than multiple computers (PCs, Macs)

Disadvantages

- Single source for possible problems
- Loss of individual user control of their own software, data, and peripherals



Virtual Memory



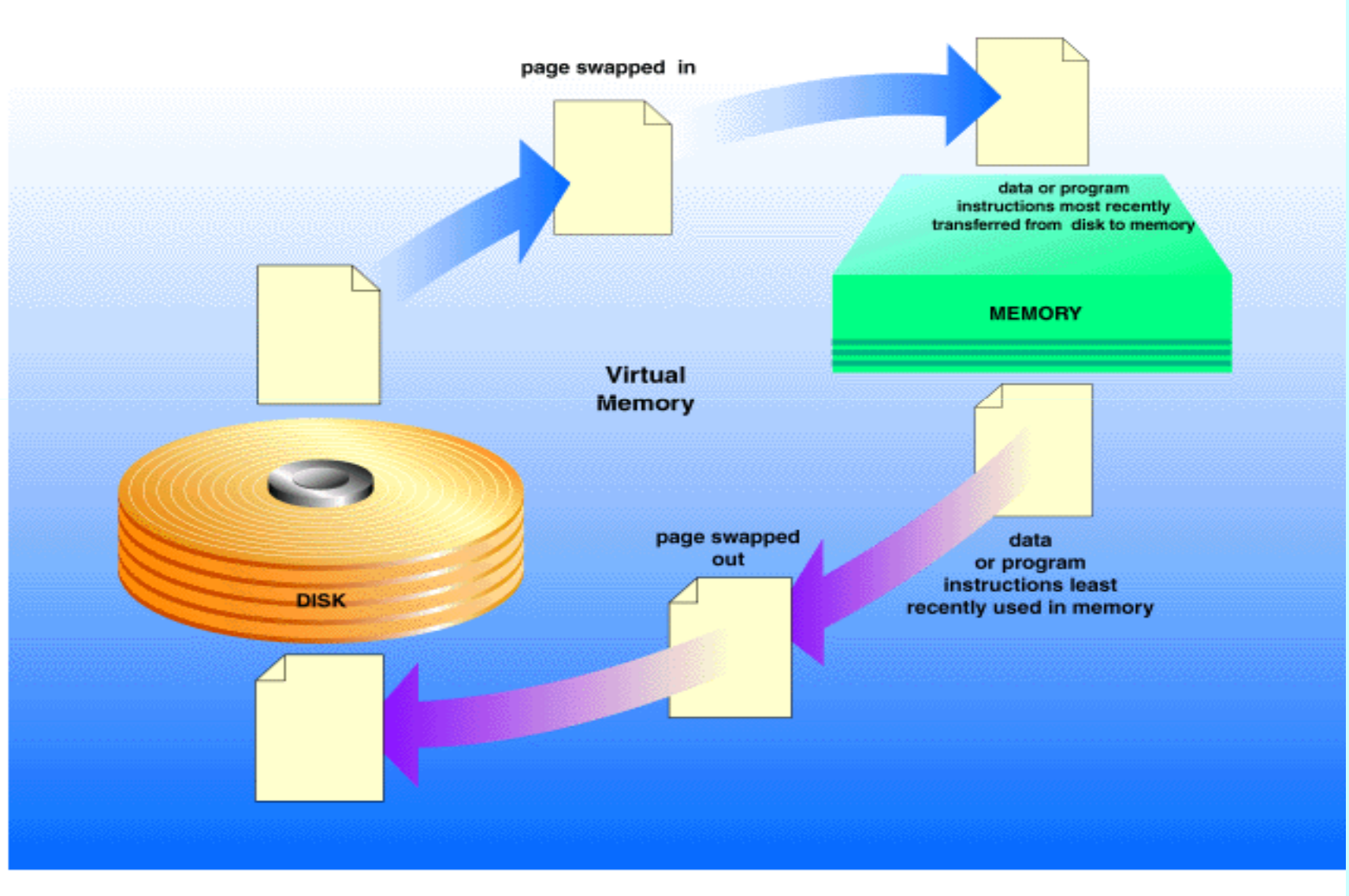
Virtual Memory (VM) = the ability of the CPU and the operating system software to use the hard disk drive as additional RAM when needed (safety net)

Good – no longer get "insufficient memory" error

Bad - performance is very slow when accessing VM

Solution = more RAM

Virtual Memory

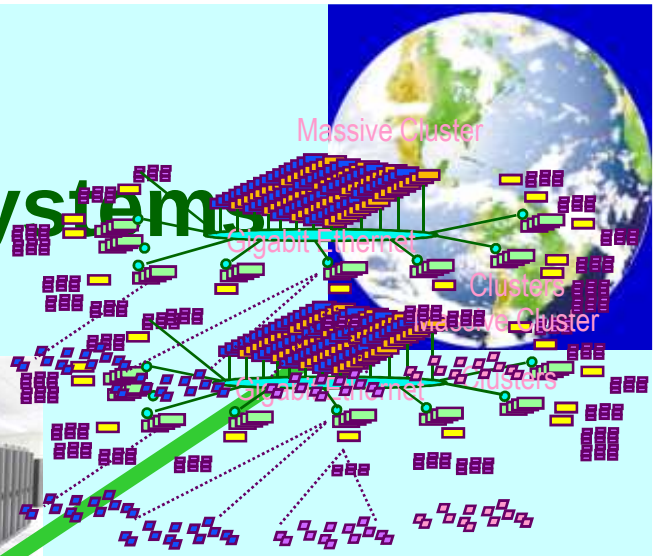


Clustered Systems

- Clustering allows two or more systems to share storage.
- Provides high reliability.
- *Asymmetric clustering*: one server runs the application while other servers standby.
- *Symmetric clustering*: all N hosts are running the application.

Societal Scale Information Systems

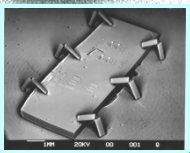
- The world is a large distributed system
  - Microprocessors in everything
  - Vast infrastructure behind them



Internet Connectivity

Scalable, Reliable, Secure Services

Databases
Information Collection
Remote Storage
Online Games
Commerce
…



MEMS for Sensor Nets

From Ion Stoica OS course slides

Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- Real-Time systems may be either *hard* or *soft* real-time.

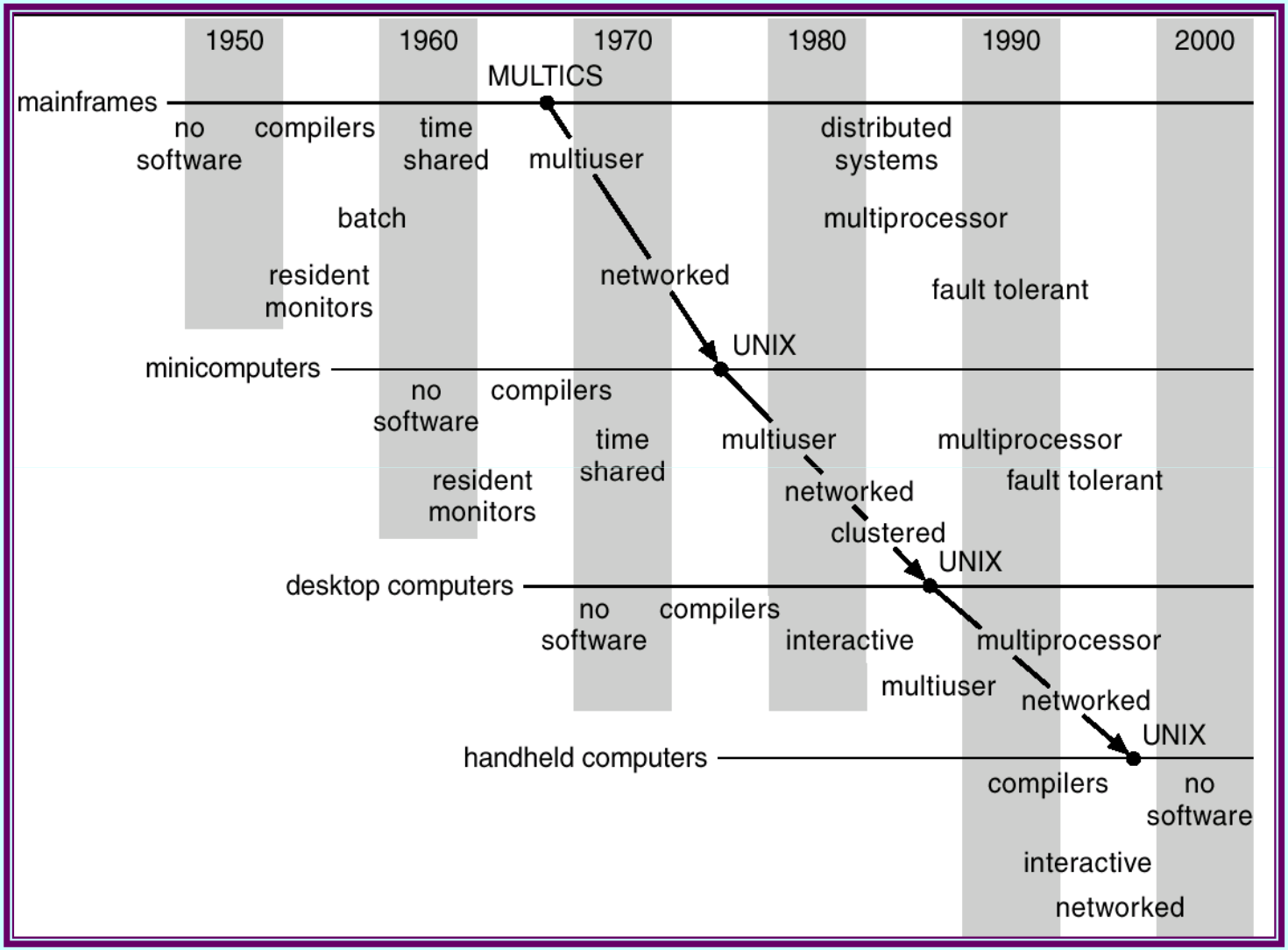
Real-Time Systems (Cont.)

- Hard real-time:
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- Soft real-time
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

Handheld Systems

- Personal Digital Assistants (PDAs)
- Cellular telephones
- Issues:
  - Limited memory
  - Slow processors
  - Small display screens.

Migration of Operating-System Concepts and Features



Computing Environments

- Traditional computing
- Web-Based Computing
- Embedded Computing

Provides and Manages System Security

Single-user Operating Systems

- minimal security
- user has full authority

Server Operating Systems

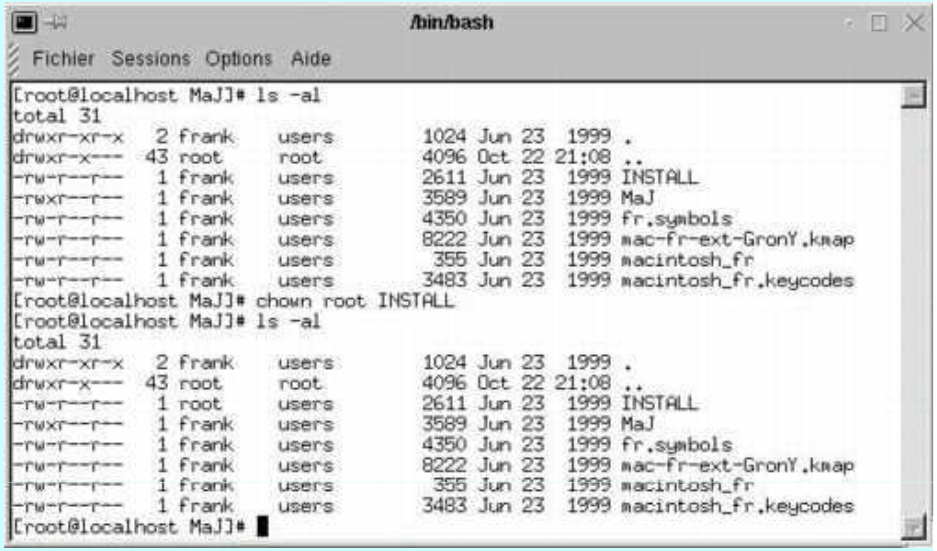
- login and password capability
- protection of user's data stored on the server's central hard disk drives
- protection and security for software programs

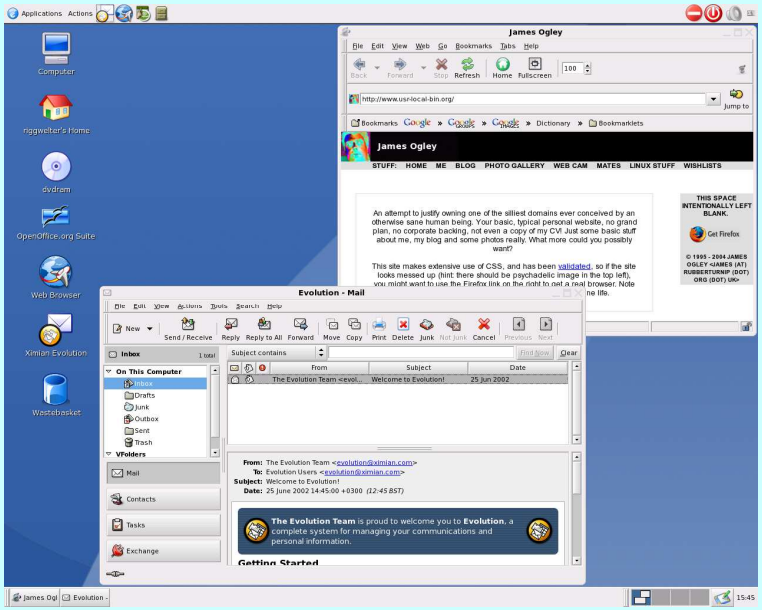
Provides the System Interface

System Interface or **shell** = the interface between the user and the computer

Command Line Interface (CLI)

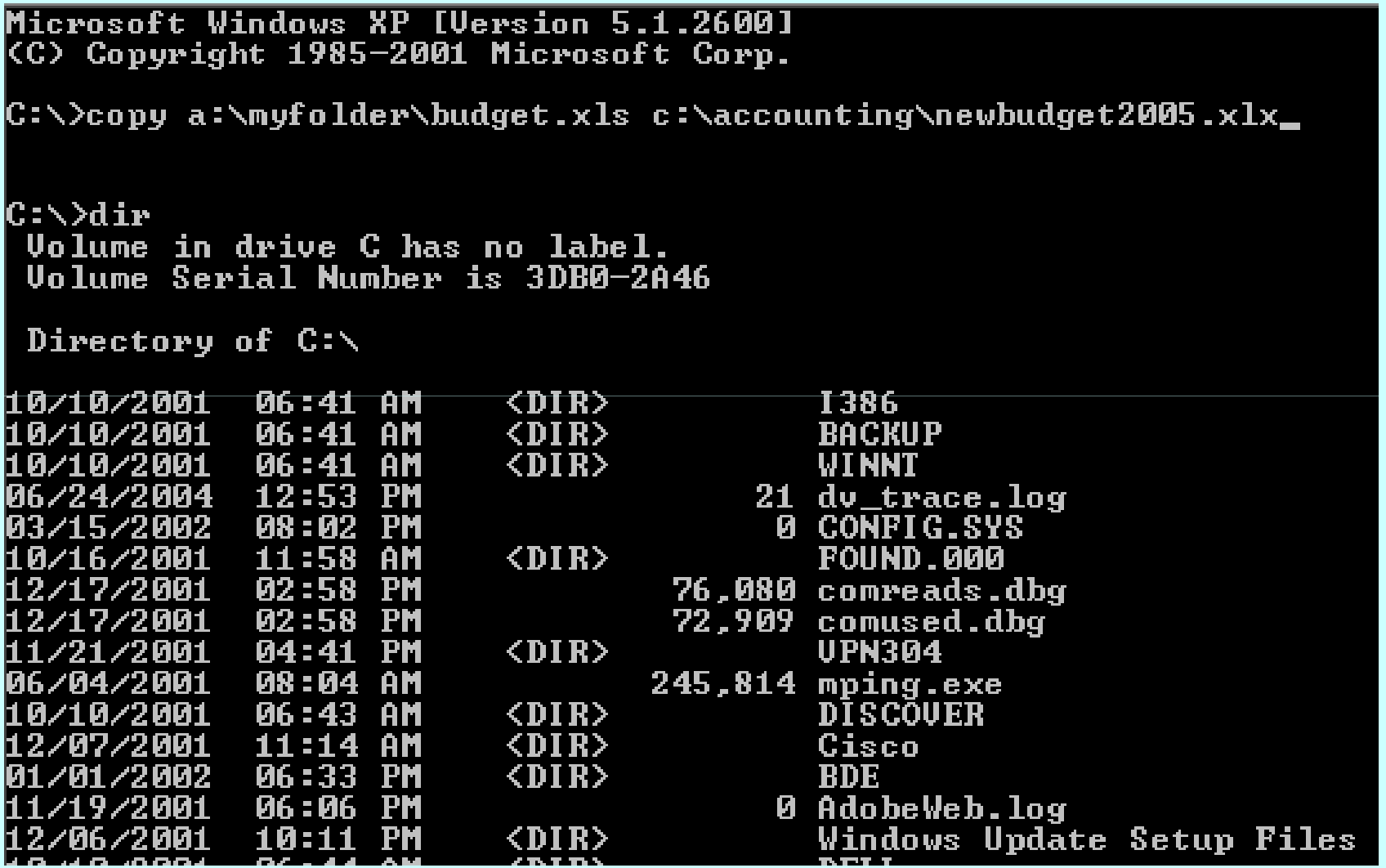
- Linux, UNIX, DOS, older OS's



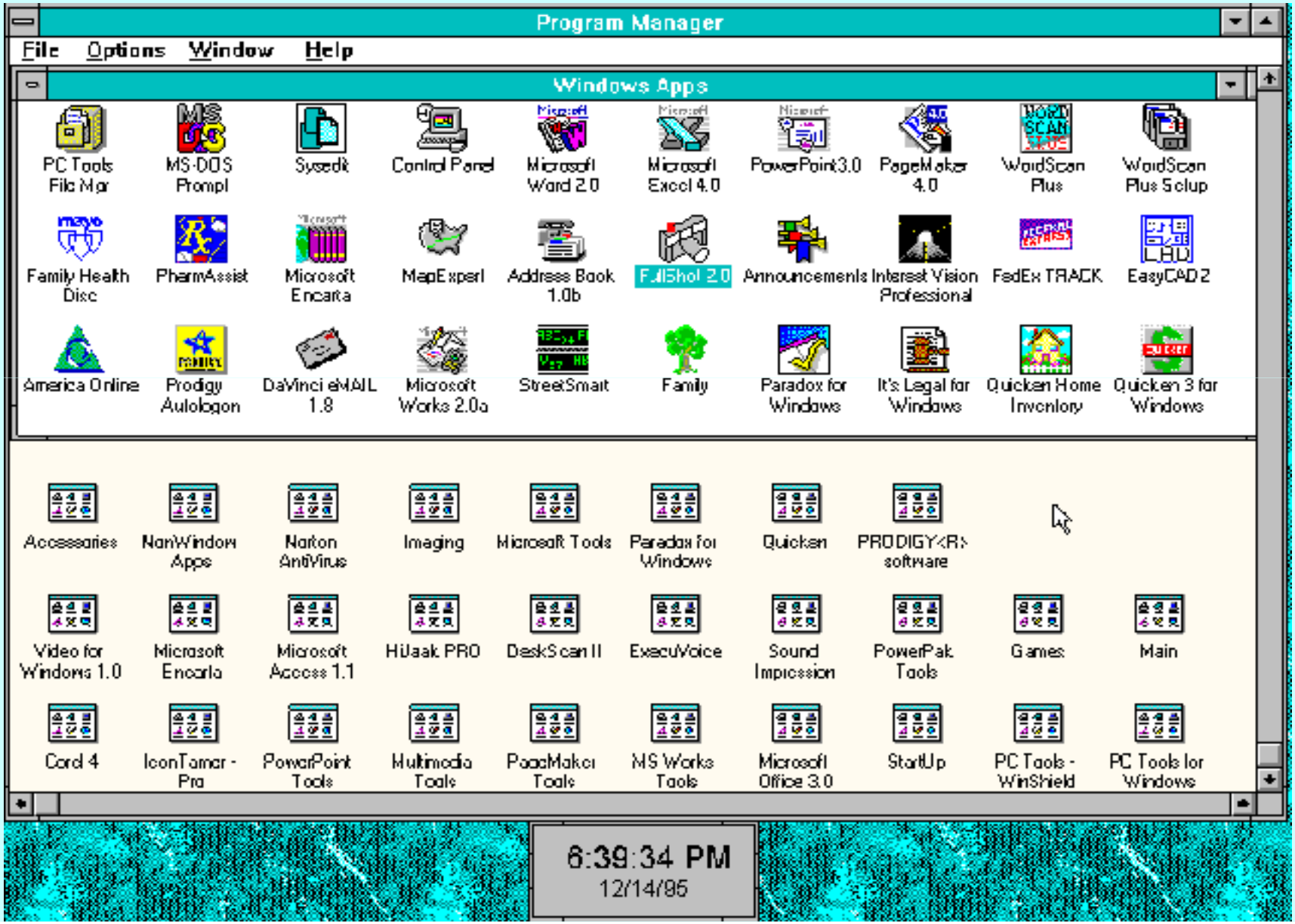


Graphical User Interface (GUI)

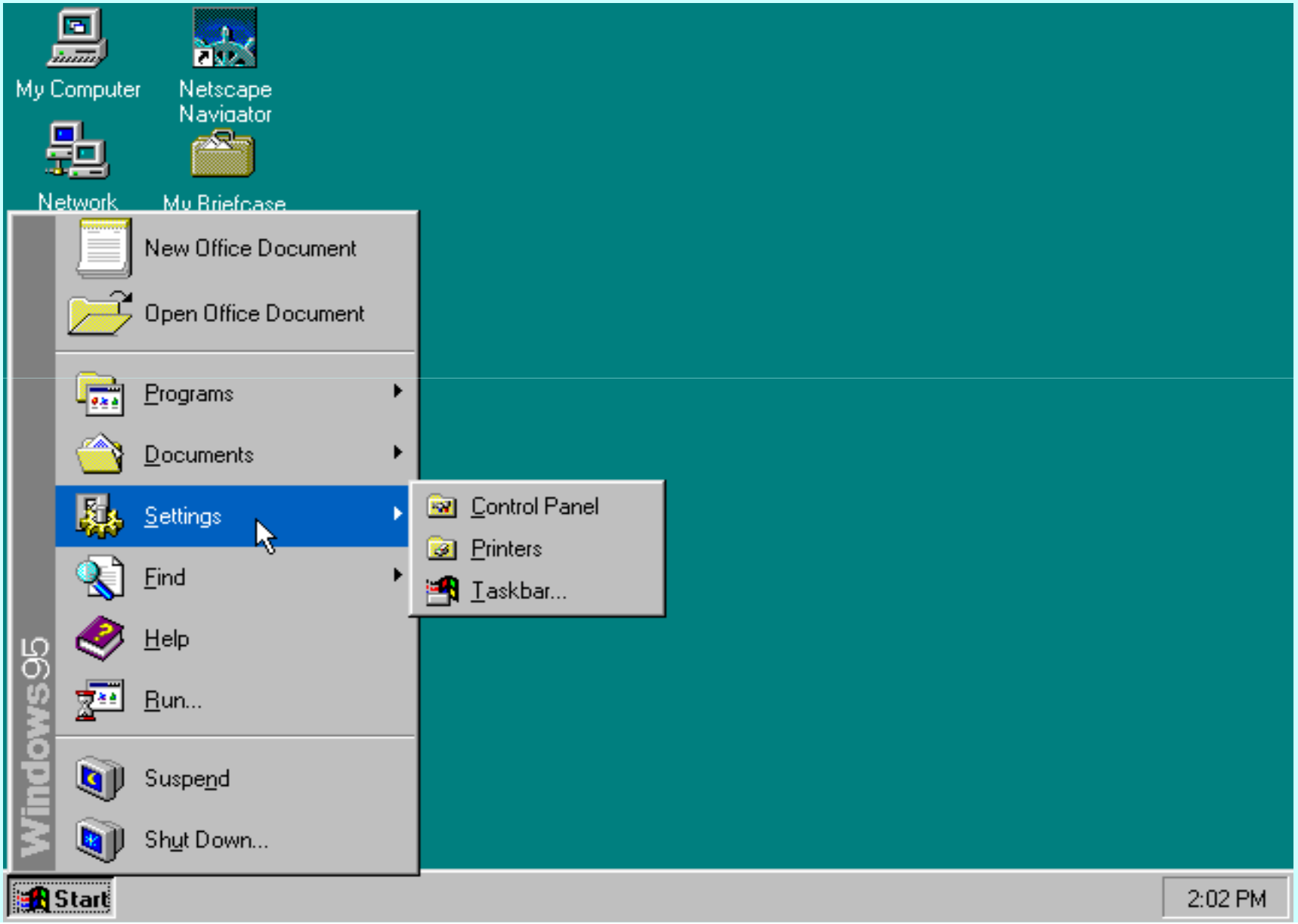
Command Line Interface
DOS, UNIX, others



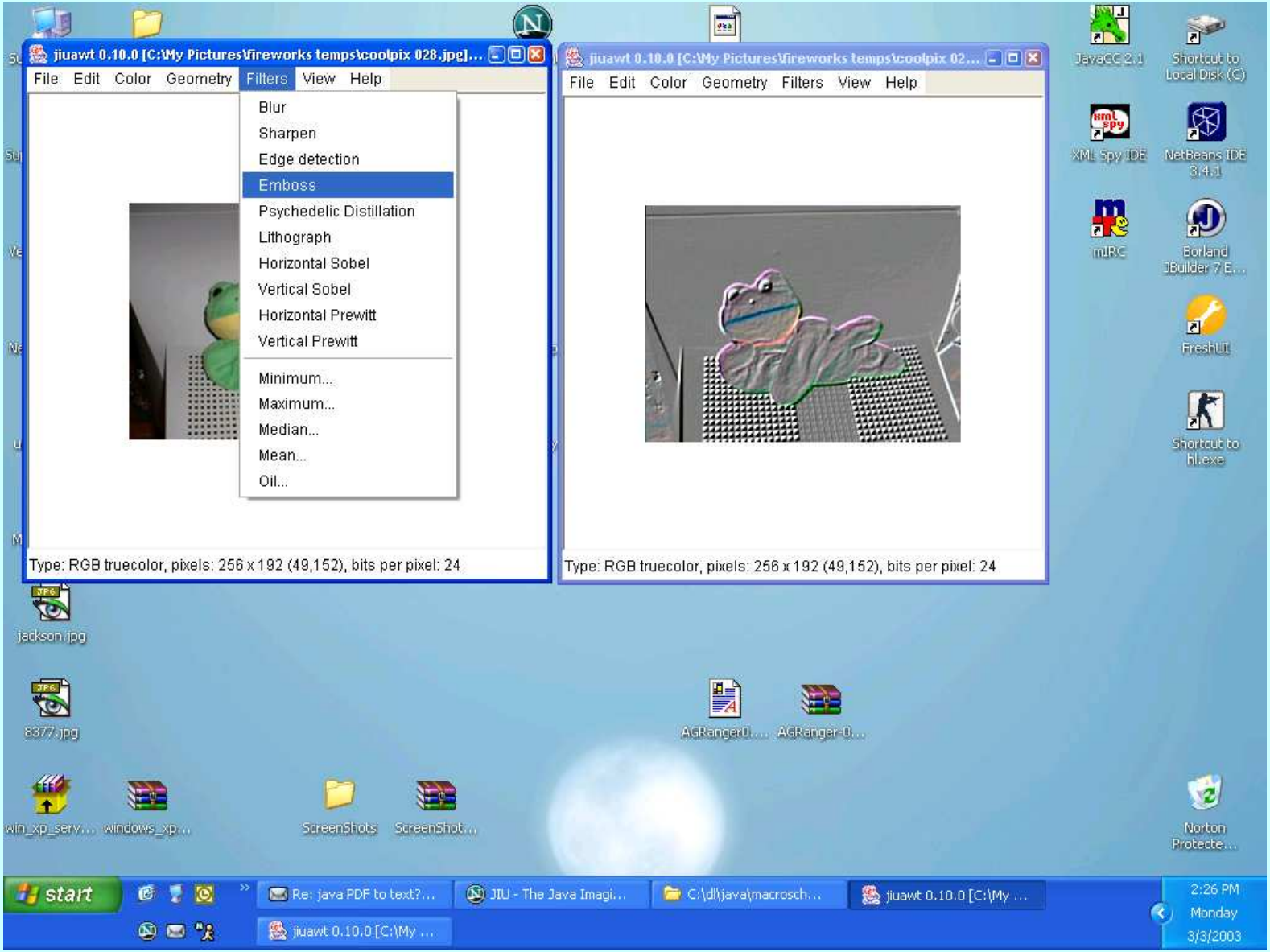
DOS plus Windows 3.1



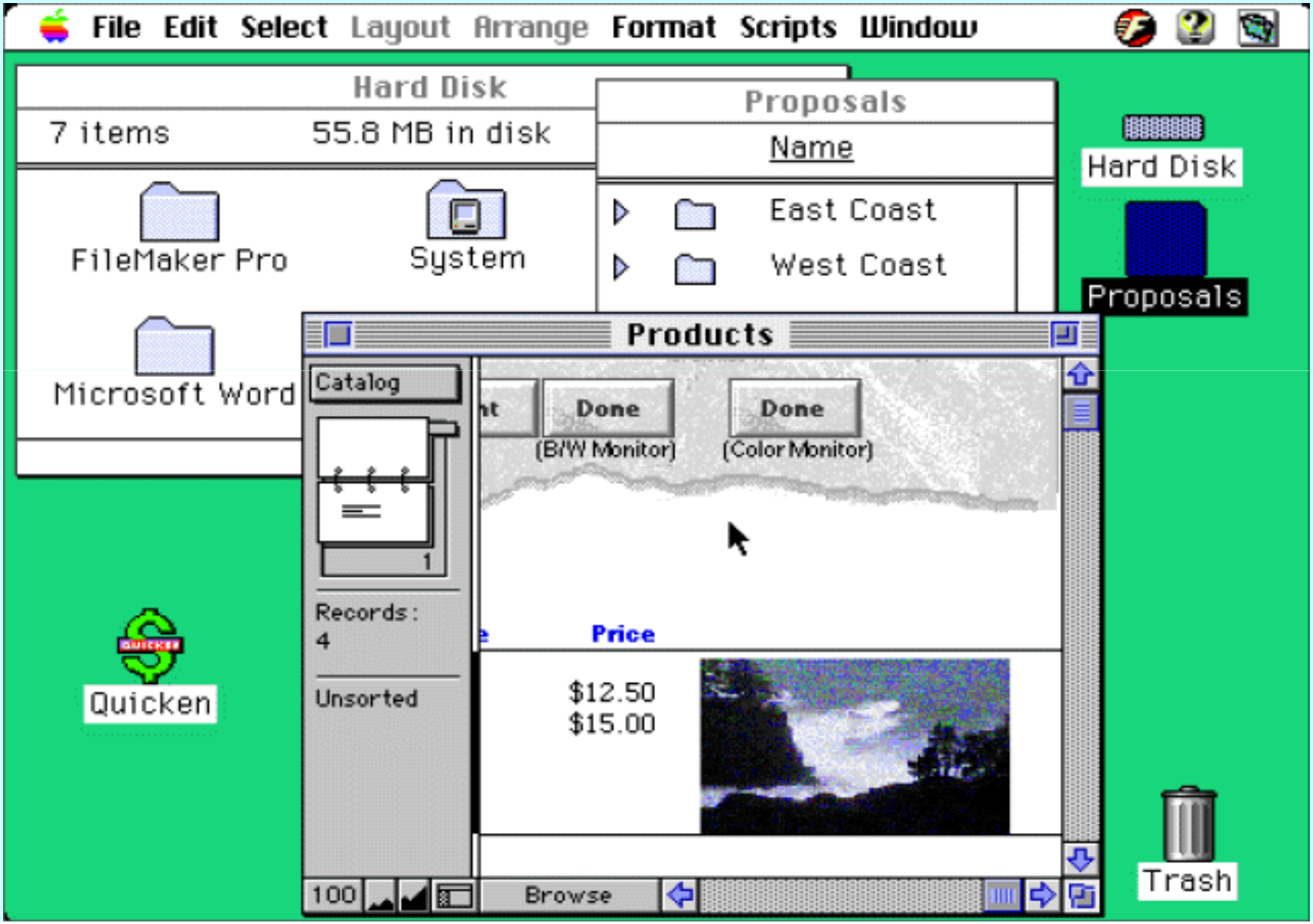
Windows 95


My Computer
Netscape Navigator
Network
My Briefcase
New Office Document
Open Office Document
Programs
Documents
Settings
Find
Help
Run...
Suspend
Shut Down...
Windows95
Control Panel
Printers
Taskbar...
Start
2:02 PM

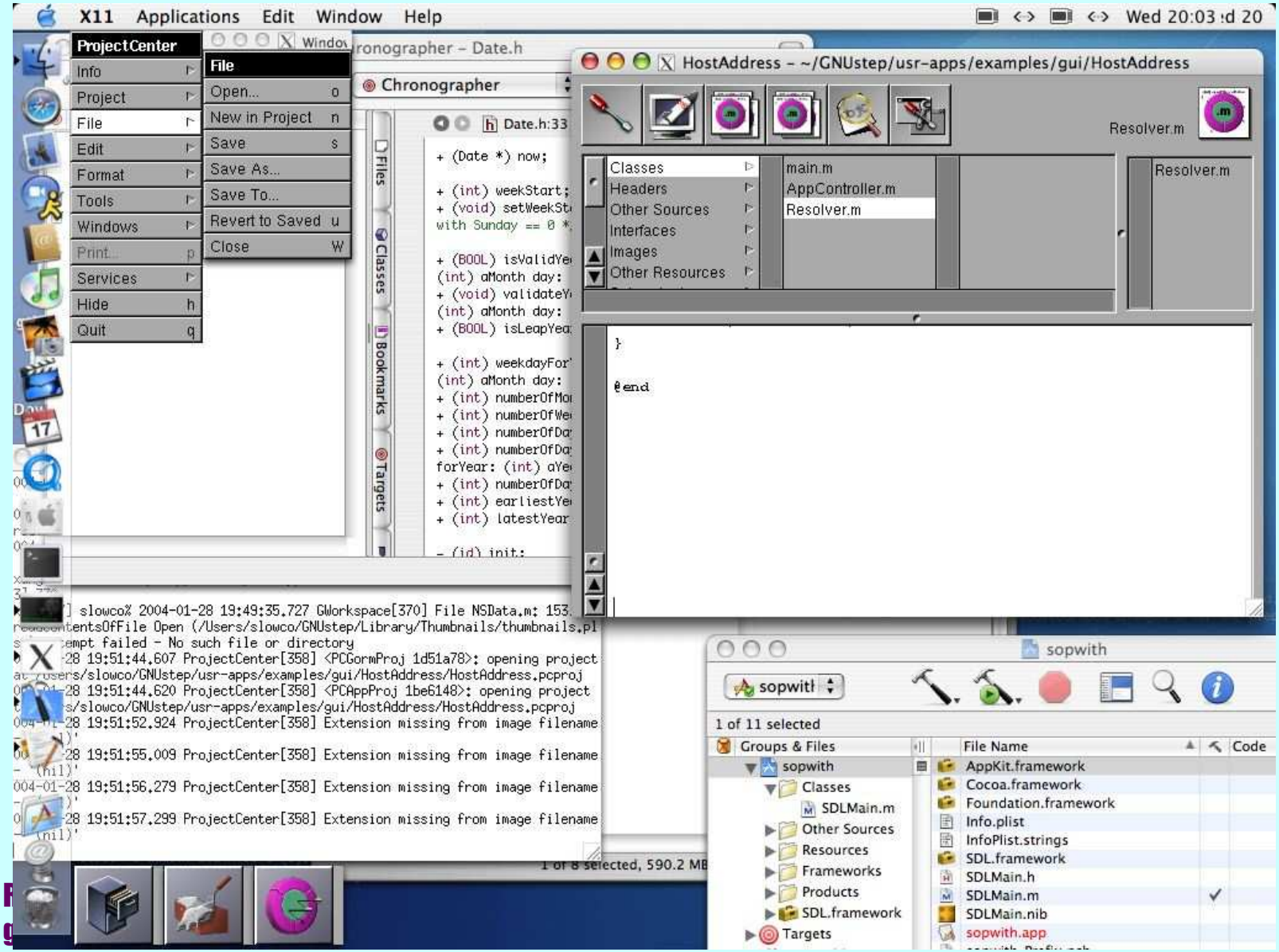
Windows XP


jiuawt 0.10.0 [C:\My Pictures\fireworks temps\coolpix 028.jpg]...
File Edit Color Geometry Filters View Help
Blur
Sharpen
Edge detection
Emboss
Psychedelic Distillation
Lithograph
Horizontal Sobel
Vertical Sobel
Horizontal Prewitt
Vertical Prewitt
Minimum...
Maximum...
Median...
Mean...
Oil...
Type: RGB truecolor, pixels: 256 x 192 (49,152), bits per pixel: 24
jiuawt 0.10.0 [C:\My Pictures\fireworks temps\coolpix 02...
File Edit Color Geometry Filters View Help
Type: RGB truecolor, pixels: 256 x 192 (49,152), bits per pixel: 24
JavaCC 2.1
Shortcut to Local Disk (C)
XML Spy IDE
NetBeans IDE 3.4.1
mIRC
Borland JBuilder 7 E...
FreshUI
Shortcut to hl.exe
jackson.jpg
8377.jpg
AGRanger0....
AGRanger-0...
win_xp_serv...
windows_xp...
ScreenShots
ScreenShot...
Norton Protecte...
start
Re: java PDF to text?...
JIU - The Java Imagi...
C:\dl\java\macrosch...
jiuawt 0.10.0 [C:\My ...
2:26 PM
Monday
3/3/2003

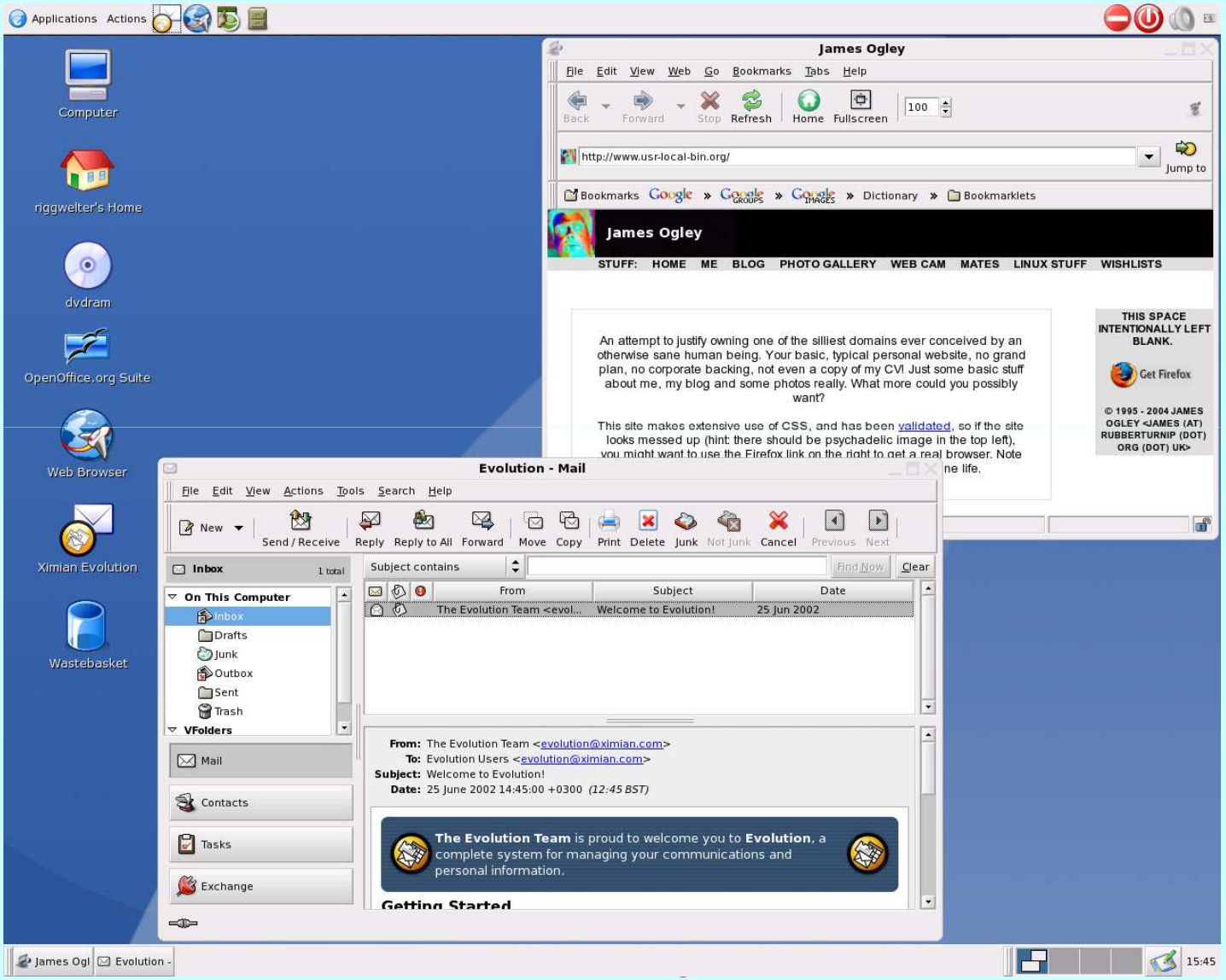
Macintosh


File Edit Select Layout Arrange Format Scripts Window
Hard Disk
7 items
55.8 MB in disk
FileMaker Pro
System
Microsoft Word
Proposals
Name
East Coast
West Coast
Hard Disk
Proposals
Products
Catalog
Done
(B/W Monitor)
Done
(Color Monitor)
Records:
4
Unsorted
Price
$12.50
$15.00
100
Browse
Quicken
Trash

Mac OS X


X11 Applications Edit Window Help
Wed 20:03 :d 20
ProjectCenter
Info
Project
File
Edit
Format
Tools
Windows
Print... p
Services
Hide h
Quit q
File
Open... o
New in Project n
Save s
Save As...
Save To...
Revert to Saved u
Close W
Chronographer
Date.h:33
+ (Date *) now;
+ (int) weekStart;
+ (void) setWeekSt
with Sunday == 0 *
+ (BOOL) isValidYe
(int) aMonth day:
+ (void) validateY
(int) aMonth day:
+ (BOOL) isLeapYea
+ (int) weekdayFor
(int) aMonth day:
+ (int) numberOfMo
+ (int) numberOfWe
+ (int) numberOfDa
+ (int) numberOfDa
forYear: (int) aYe
+ (int) numberOfDa
+ (int) earliestYe
+ (int) latestYear
- (id) init:
Files
Classes
Bookmarks
Targets
HostAddress - ~/GNUstep/usr-apps/examples/gui/HostAddress
Resolver.m
Classes
Headers
Other Sources
Interfaces
Images
Other Resources
main.m
AppController.m
Resolver.m
}
@end
slowco% 2004-01-28 19:49:35.727 GWorkspace[370] File NSData.m: 153.
entsOfFile Open (/Users/slowco/GNUstep/Library/Thumbnails/thumbnails.pl
empt failed - No such file or directory
28 19:51:44.607 ProjectCenter[358] <PCGormProj 1d51a78>: opening project
at /Users/slowco/GNUstep/usr-apps/examples/gui/HostAddress/HostAddress.pcproj
28 19:51:44.620 ProjectCenter[358] <PCAppProj 1be6148>: opening project
s/slowco/GNUstep/usr-apps/examples/gui/HostAddress/HostAddress.pcproj
28 19:51:52.924 ProjectCenter[358] Extension missing from image filename
28 19:51:55.009 ProjectCenter[358] Extension missing from image filename
004-01-28 19:51:56.279 ProjectCenter[358] Extension missing from image filename
28 19:51:57.299 ProjectCenter[358] Extension missing from image filename
1 of 8 selected, 590.2 MB
sopwith
sopwith
1 of 11 selected
Groups & Files
File Name
Code
sopwith
Classes
SDLMain.m
Other Sources
Resources
Frameworks
Products
SDL.framework
Targets
AppKit.framework
Cocoa.framework
Foundation.framework
Info.plist
InfoPlist.strings
SDL.framework
SDLMain.h
SDLMain.m
SDLMain.nib
sopwith.app

UNIX with X-Windows



Microsoft Windows 8

Provides the Interface for Application Software

- Operating systems are software
- Operating systems are designed and developed for a specific CPU or “family of CPUs”
  - **Macintosh OS**: Motorola 680xx, PowerPC Gx, Intel
  - **DOS**: Intel CPUs
  - **Windows 9x and XP**: Intel 80386, 80486, and Pentium CPUs
  - **Linux:** Intel CPUs
  - **MS NT** & **2000**: Intel CPUs

Provides the Interface for Application Software (continued)

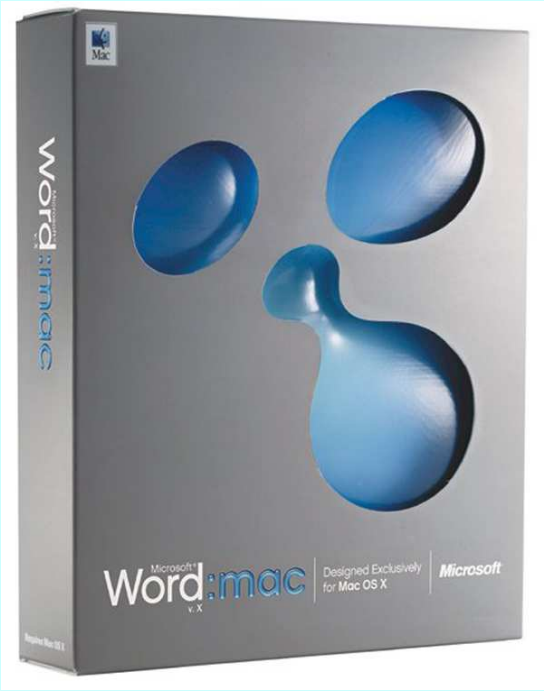
- Application software is developed for an operating system

MS Word for Windows XP

- Windows XP
- Intel CPU

MS Word for the Macintosh

- Macintosh OS X
- Gx CPU or Intel CPU



Compatibility

Question

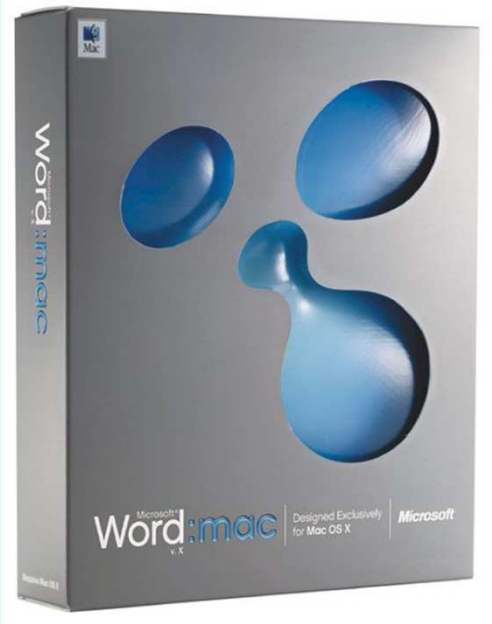
Will software developed for one operating system work on another?

- Will MS Word for Macintosh run on a PC with Windows XP?

Answer

No (unless there is special emulation software or hardware). The software must be developed separately for each operating system.

Much of this is beginning to change with MAC using the Intel CPU.



Order of Development

1. **The CPU**

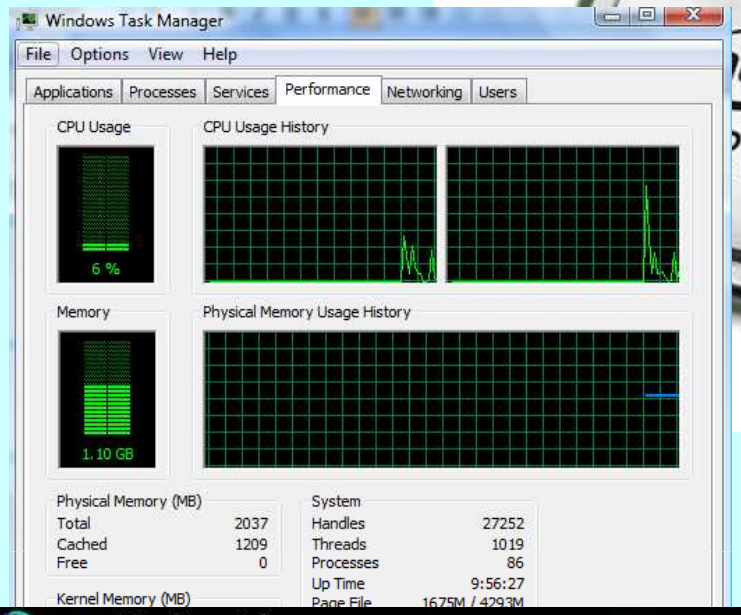
2. **Other Hardware Components**

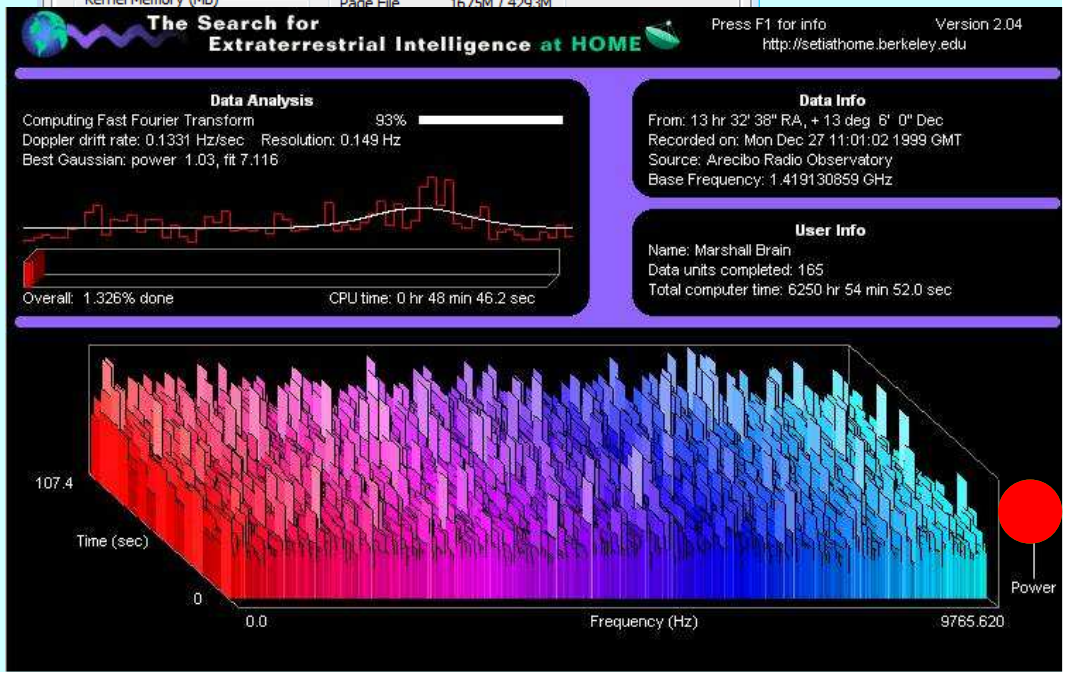
3. **Operating System Software**

4. **Application Software**

Today's Operating Systems

- **Today's CPUs** (multiprocessors)
  - **Multiple processors**
  - **Load balancing**
    - Dynamically allocating tasks to the various processors so that all processors are used efficiently.
  - **Scaling**
    - Breaking tasks into a number of subtasks equal to the number of processors available.
  - **The Network (Internet)**
    - Becoming a single network-wide operating system rather than a network of individual operating systems.





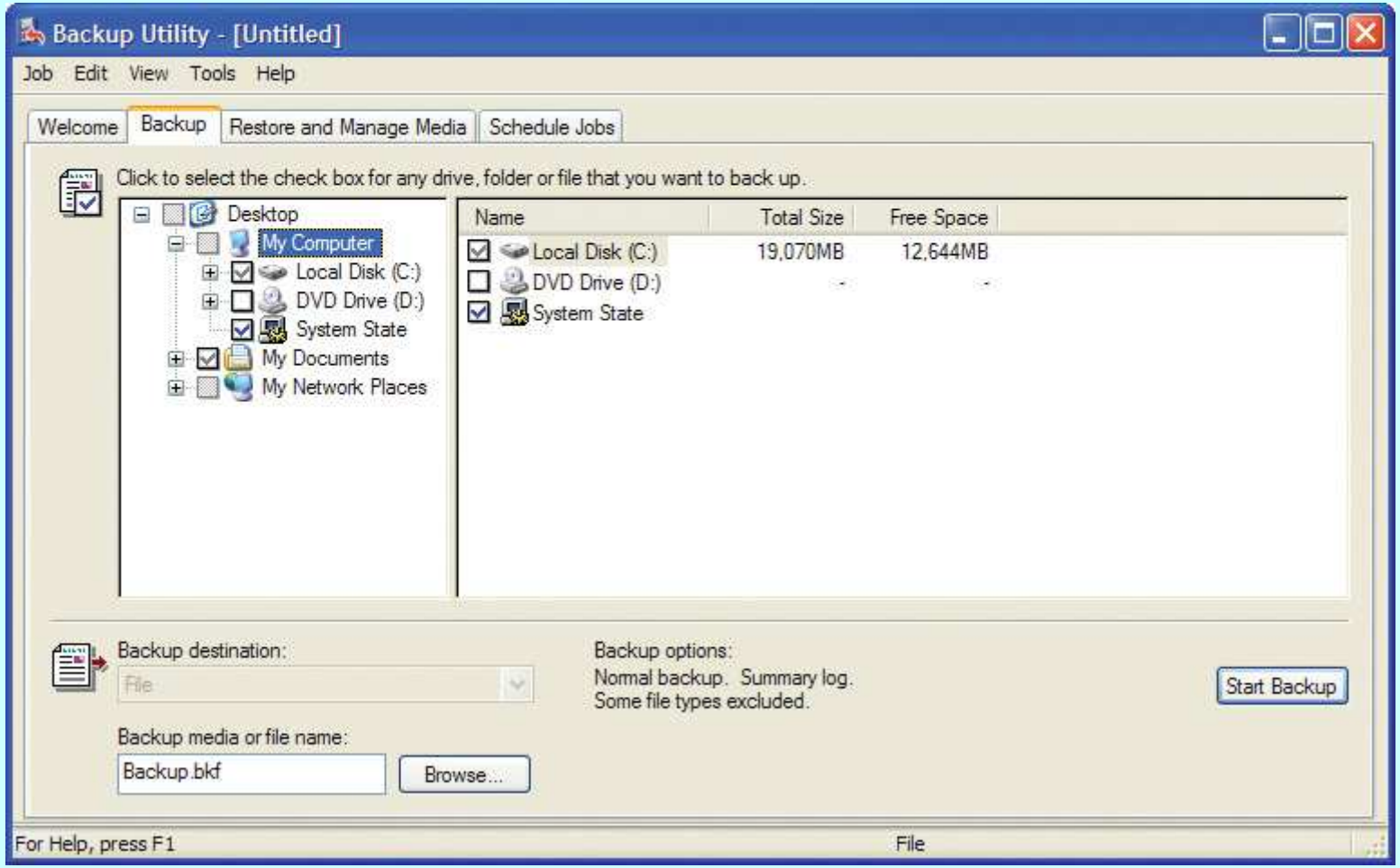
Enhancing an OS

- Utilities
  - Provide services not included with OS
  - Goes beyond the four functions
  - Firewall, anti-virus and compression
  - Prices vary

Enhancing an OS

- Backup software
  - Archives files onto removable media
  - Ensures data integrity
  - Most OS include a backup package
  - Many third party packages exist

Backup Software



Enhancing an OS

- Anti-virus software
  - Crucial utility
  - Finds, blocks and removes viruses
  - Must be updated regularly
  - McAfee and Norton Anti-Virus

Enhancing an OS

- Firewall
  - Crucial utility
  - Protects your computer from intruders
  - Makes computer invisible to hackers
  - Zone Labs is a home firewall
  - Cisco sells hardware firewalls

Enhancing an OS

- Intrusion detection
  - Often part of a firewall package
  - Announces attempts to breach security
  - Snort is a Linux based package

Enhancing an OS

- Screen savers
  - Crucial utility for command line systems
    - Prevents burn in
  - Merely fun for GUI systems
  - Screen saver decorates idle screens