

# Operating System Overview



# Operating System Objectives

- Operating systems are among the most complex pieces of software ever developed

## ☞ Convenience

- Makes the computer more convenient to use

## ☞ Efficiency

- Allows computer system resources to be used in an efficient manner

## ☞ Ability to evolve

- Permit effective development, testing, and introduction of new system functions without interfering with service



# Operating System Services

- Program development
  - ✧ Editors, debuggers, frameworks
- Program execution
  - ✧ Initialization, scheduling
- Access to I/O devices
  - ✧ Uniform interface, hides details
- Controlled access to files
  - ✧ Authorization, sharing, caching



# OS Services (continued...)

- System access
  - ✧ Protection, authorization, resolve conflicts
- Error detection and response
  - ✧ Hardware errors: memory error or device failure
  - ✧ Software errors: arithmetic errors, access forbidden memory locations, allocation errors
- Accounting
  - ✧ collect statistics (billing)
  - ✧ monitor performance
  - ✧ used to anticipate future enhancements



# OS as a Resource Manager

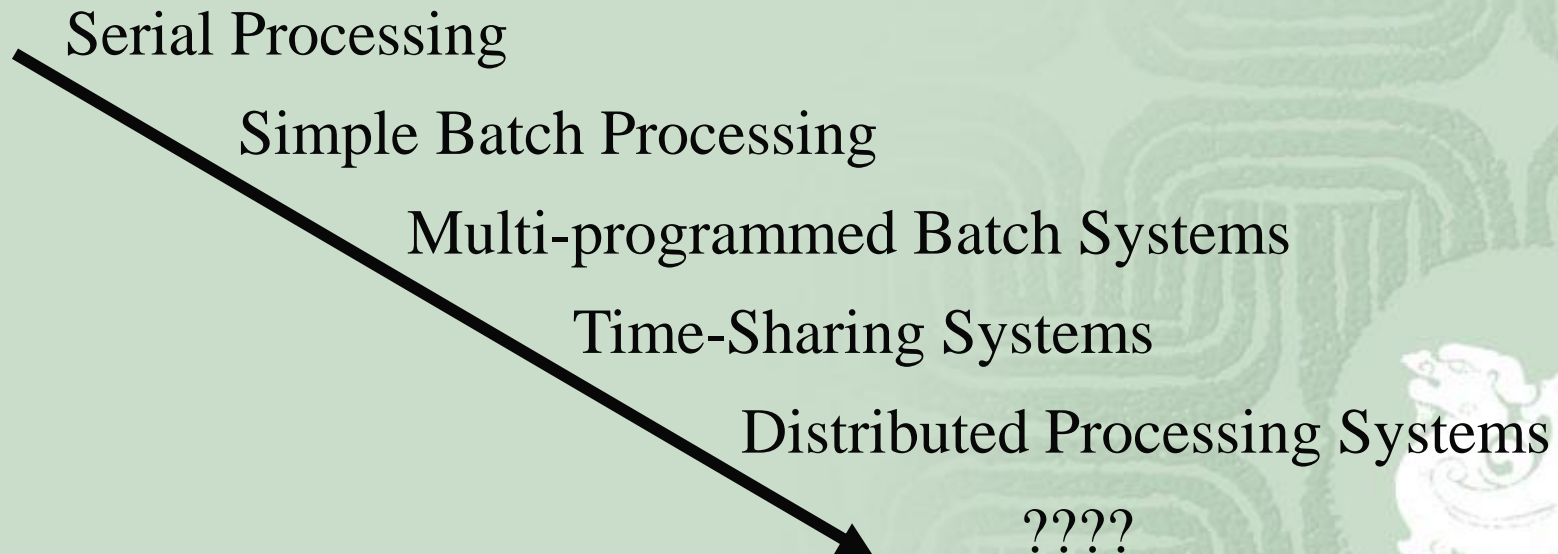
- OS executes same way as ordinary computer software - it is set of computer programs
- The key difference is
  - ❧ To manage resources
  - ❧ To Schedule processor to execute programs
- Kernel
  - ❧ Portion of operating system that is in main memory
  - ❧ Contains most-frequently used functions





# Evolution of Operating Systems

- Operating systems have evolved because
  - ∞ New types of hardware and hardware upgrades
  - ∞ Development of new services and needs
  - ∞ Fixes to OS faults



# Serial Processing

## ■ Serial Processing

- ❧ No operating system
- ❧ Machines run from a console with display lights and switches
- ❧ Schedule time
- ❧ Setup included loading the compiler, source program, saving compiled program, and loading and linking



# Simple Batch Systems

## ■ Simple Batch Systems

### ∞ Monitors

- Software that controls the running programs
- Batch jobs together
- Program branches back to monitor when finished
- Resident monitor is in main memory and available for execution

### ∞ Job Control Language (JCL)

- Special type of programming language
- Provides instructions to the monitor (what compiler/data to use)

### ∞ Hardware Features

- Memory protection - do not allow the memory area containing the monitor to be altered
- Timer - prevents a job from monopolizing the system



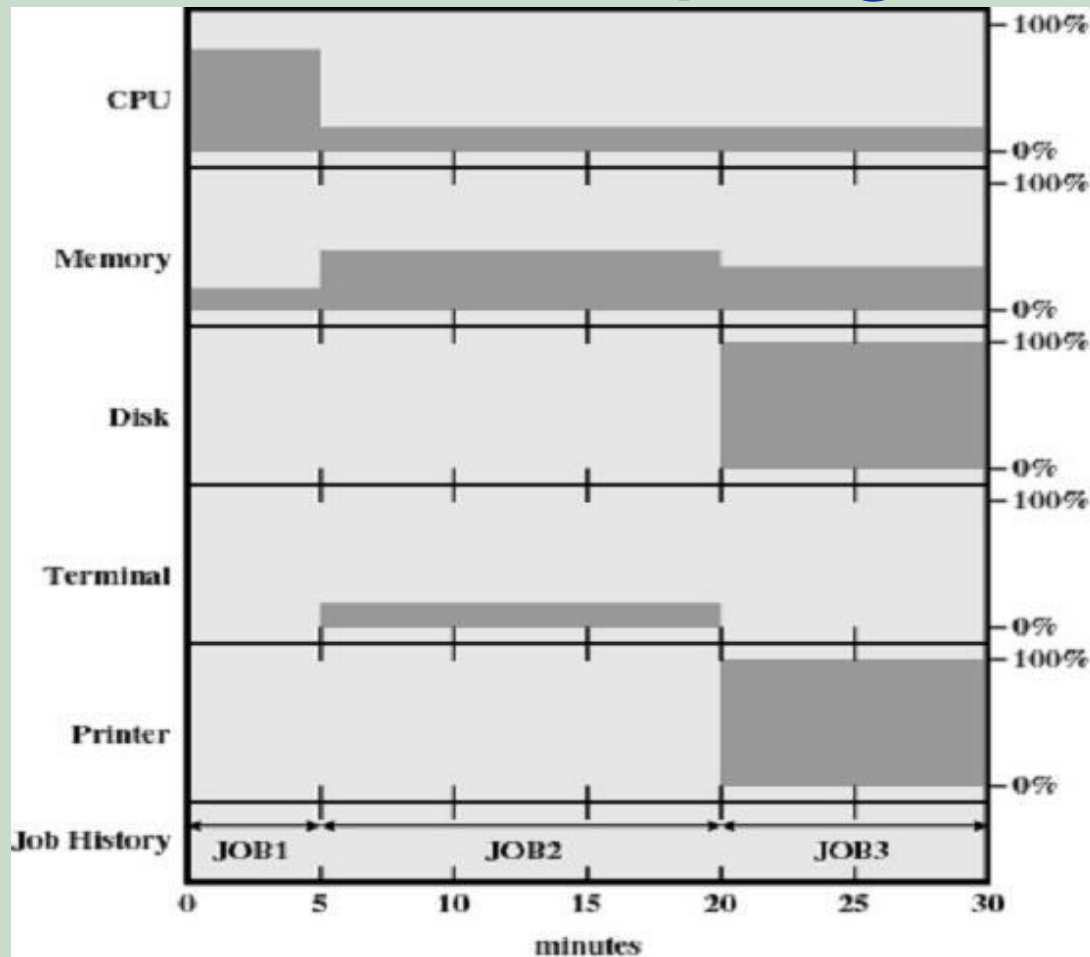


# Sample Example

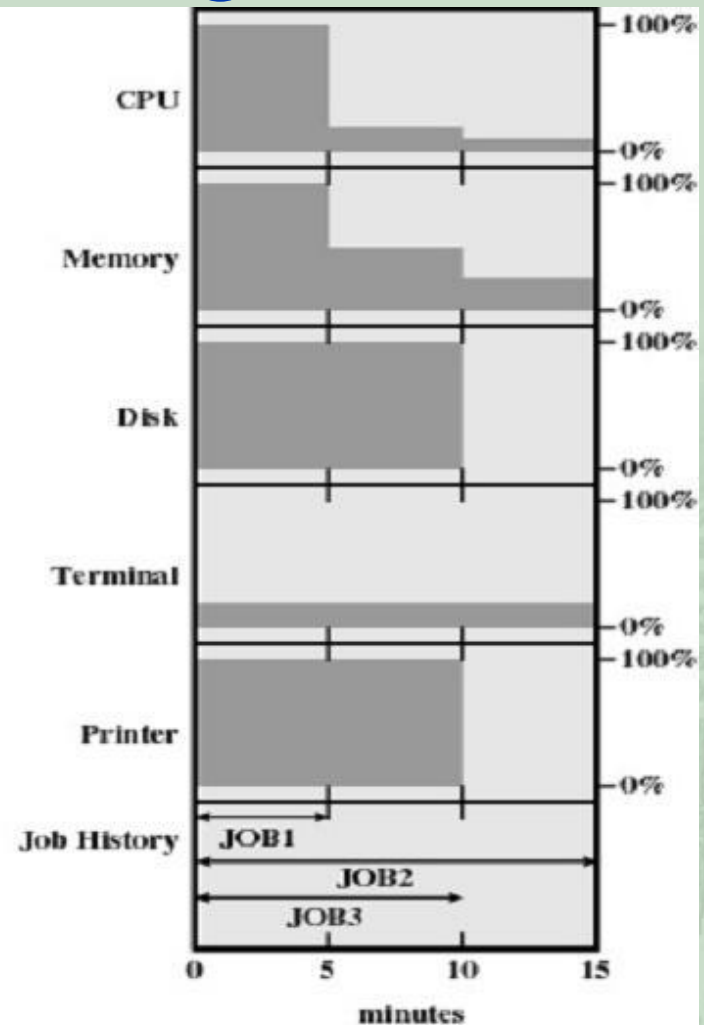
	Job 1	Job 2	Job 3
Type of Job	Heavy computing	Heavy memory	Heavy I/O
Duration	5 min	15 min	10 min
Memory	50 M	100 M	75 M
Need disk	No	No	Yes
Need terminal	No	Yes	No
Need printer	No	No	Yes



# Multiprogramming



(a) Uniprogramming



(b) Multiprogramming

# Effects of Multiprogramming

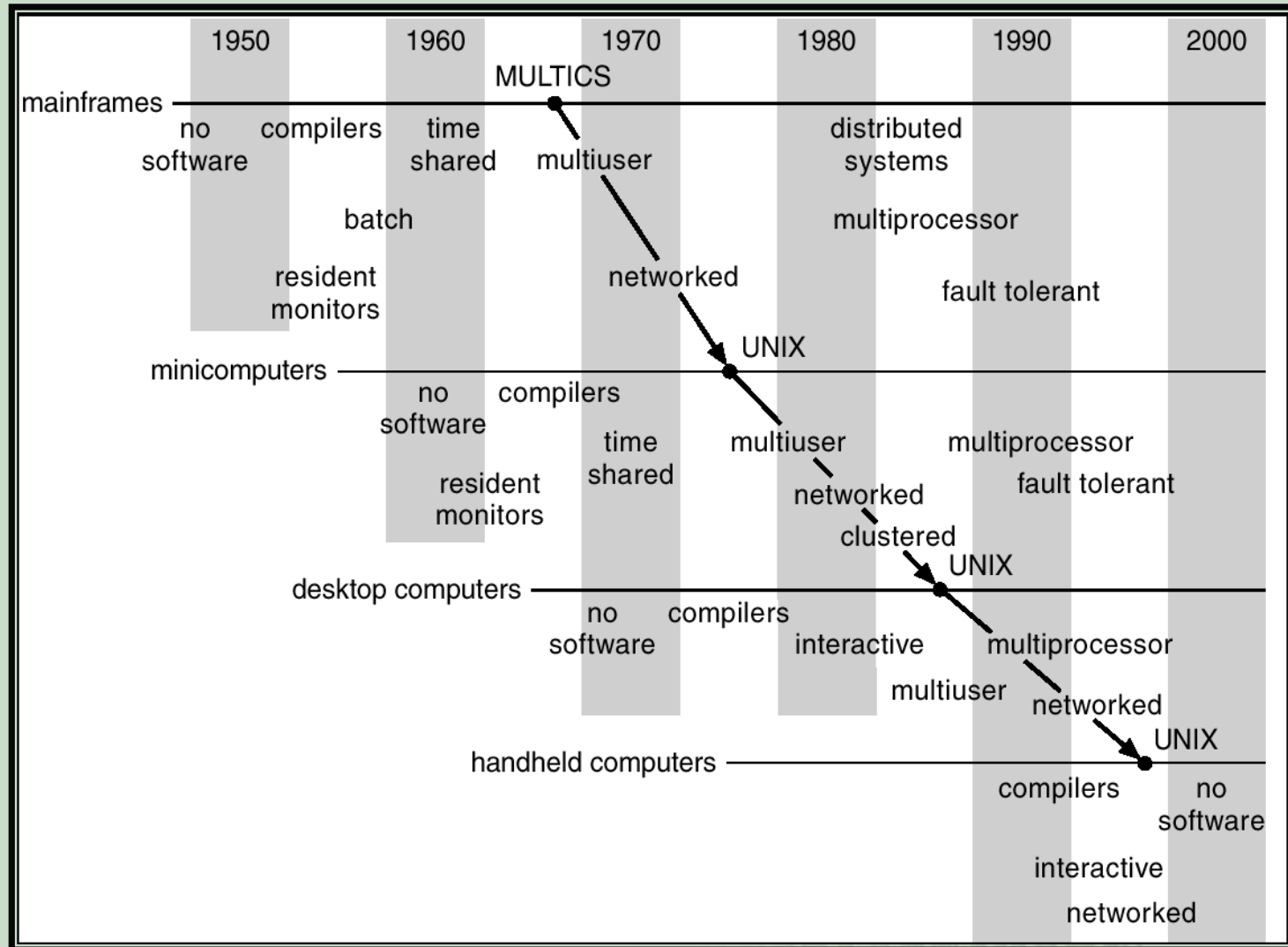
	Uniprogramming	Multiprogramming
Processor use	20%	40%
Memory use	33%	67%
Disk use	33%	67%
Printer use	33%	67%
Elapsed time	30 min.	15 min.
Throughput rate	6 jobs/hr	12 jobs/hr
Mean response time	18 min. $(5+20+30)/3$	10 min. $(5+10+15)/3$

# Time-Sharing Systems

- Allow several users to interact at the same time
- Emphasizes response time over processor use
- Compatible time sharing system (CTSS) – 1962
  - ⌘ 32K 36-bit words, switched users every 0.2 seconds
  - ⌘ Supported up to 32 users
- MULTICS (1965) – Computer service for Boston
- Systems grew as user requirements expanded
  - ⌘ MS-DOS 1.0 (8k, 1981)
  - ⌘ DOS 3.1 – Networking (1984)
  - ⌘ Windows 3.1 – Graphical interface (1990)
  - ⌘ Windows 95 – 32-bit internals (1995)
  - ⌘ ...



# Migration of Operating-System Concepts and Features





# Distributed Processing Systems

- Symmetric multiprocessing
  - ☞ there are multiple processors
  - ☞ these processors share same main memory and I/O facilities
  - ☞ All processors can perform the same functions
- Distributed operating systems
  - ☞ provides the illusion of a single main memory and single secondary memory space
  - ☞ used for distributed computing



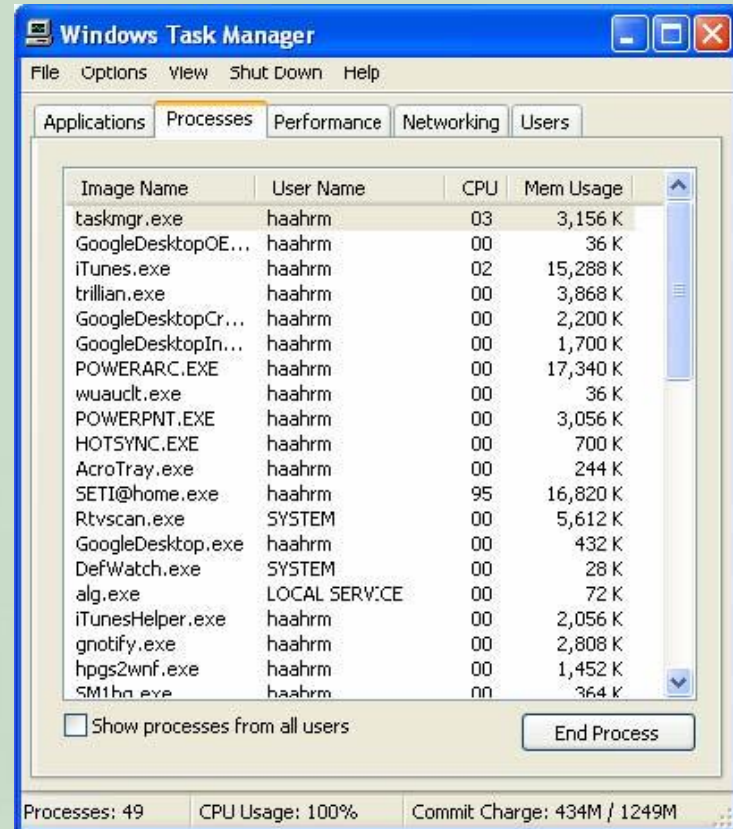
# Major Achievements

- Processes
- Memory Management
- Information protection and security
- Scheduling and resource management
- System structure



# Processes in Windows XP



- Hit CTRL-ALT-DEL to start Task Manager
- Shows info about the PC's performance
- Info gathered by OS; updated continuously
- The Applications tab shows running applications and allows them to be ended



# Processes in Linux

kap@cyanide - KPM

File View Process Signal Settings Help

cpu  mem  swap (none)

user nice sys idle used buff cach free up 282 days, 05:54

PID	USER	PRI	NICE	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	CMDLINE
561	root	11	-1035	8252	40400	1816	S<	1.38	15.85	55:55h	/usr/X11R6/bin/X -dpi 100 -nolisten tcp vt7 -auth /var/
631	haahrm	19	0	3224	2444	1096	R	1.98	3.96	83:41	/usr/bin/artsd -F 10 -S 4096 -s 60 -m artsmessage -l 3
29549	haahrm	15	0	6896	6892	5760	R	0.99	2.70	2.93s	kpm -icon kpm -miniicon kpm
635	haahrm	10	0	3900	3300	2240	S	0.20	1.29	4:41	kdeinit: kwin -session 1186e224420001029924315000
29554	haahrm	12	0	8500	8500	3944	S	0.40	3.33	1.78s	gimp
12689	haahrm	9	0	57824	43132	10548	S	0.00	13.92	19:28	/usr/lib/mozilla-firefox/firefox-bin -a firefox
639	haahrm	9	0	13092	12032	3112	S	0.20	4.72	10:47	kdeinit: kicker
633	haahrm	10	0	2656	1824	1452	S	0.20	3.72	3:06	kdeinit: knotify
637	haahrm	9	0	3860	3228	2436	S	0.00	1.27	5:12	kdeinit: kdesktop
6	root	9	0	0	0	0	SW	0.00	3.00	2:32h	(kupdated)
29555	haahrm	9	0	2960	2960	1444	S	0.00	1.16	0.43s	/usr/lib/gimp/1.2/plugin-ins/script-fu -gimp 9 8 -run 0
616	haahrm	9	0	1408	808	676	S	0.00	3.32	8.78s	kdeinit: dcopserver --nosid
22103	haahrm	9	0	1560	1340	1020	S	0.00	3.53	0.51s	xterm
11037	haahrm	9	0	1608	708	548	S	0.00	3.28	4.40s	kdeinit: kcookiejar
18358	root	9	0	1064	748	596	S	0.00	3.29	9:06	/usr/sbin/nmbd -D
4	root	9	0	0	0	0	SW	0.00	3.00	89:38	(kswapd)
24255	haahrm	9	0	23940	10132	2572	S	0.00	3.97	3:30	pine
22104	haahrm	8	0	1352	1036	660	S	0.00	3.41	0.13s	-sh
12921	haahrm	9	0	57824	43132	10548	S	0.00	13.92	9.71s	/usr/lib/mozilla-firefox/firefox-bin -a firefox
2	root	9	0	0	0	0	SW	0.00	3.00	0:33	(keventd)
355	root	9	0	17480	1784	520	S	0.00	3.70	1:17	/usr/bin/X11/xfs -daemon
12923	haahrm	9	0	57824	43132	10548	S	0.00	13.92	0:15	/usr/lib/mozilla-firefox/firefox-bin -a firefox
386	root	9	0	0	0	0	SW	0.00	3.00	3:53	(rpciod)
27413	haahrm	9	0	1284	956	788	S	0.00	3.37	0.11s	xterm
650	haahrm	9	0	1664	1064	748	S	0.00	3.42	10:51	xterm

# Memory Management

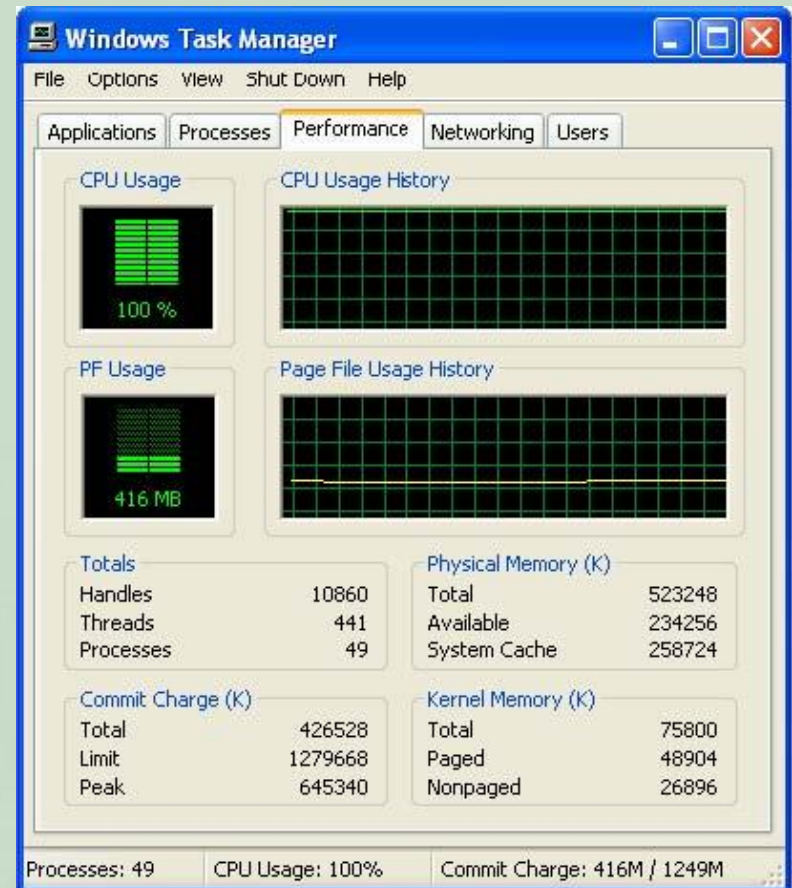
- Responsibilities of memory management
  - ❧ Process isolation
  - ❧ Automatic allocation and management
  - ❧ Support of modular programming
  - ❧ Protection and access control
  - ❧ Long-term storage
- These requirements typically met by
  - ❧ Virtual memory
  - ❧ File system facilities





# Memory in Windows XP

- The Performance tab shows CPU and virtual memory usage over time
- Memory statistics also shown



# Information Protection and Security

- Time-sharing and computer networks require
  - ∞ Availability
  - ∞ Confidentiality
  - ∞ Data integrity
  - ∞ Authenticity

