

Two-Hour Introduction to Operating Systems

Dr. Mads Haahr
Department of Computer Science
Trinity College, Dublin



Some slides used here were adapted from notes
by William Stallings & Patty Roy – thanks!

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

1

What is an Operating System?



- OS = Operating System
- A program that controls the execution of application programs
 - An interface between applications and hardware
 - Makes the computer more convenient to use
- Manages the resources of a computer and controls the way they are used
 - Allows resources to be used in an efficient manner
- Examples of OS's?

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

2

Where are OS's Used?



- In more and more places!
- Desktop and Server Computers
 - DOS + Windows 95/98/ME
 - Windows NT/2000/XP
 - Free Unix variants: Linux, FreeBSD, NetBSD, etc.
 - Commercial Unix variants: Solaris, HP-UX, AIX, etc.
 - MacOS
- Some Game Consoles
 - Xbox: Cut-down Windows 2000

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

3

Where are OS's Used?



- Personal Digital Assistants (PDAs)
 - PalmOS
 - Windows CE → Windows Mobile
 - Embedded Linux
- Mobile Phones
 - Symbian OS
 - Windows Mobile
- Cars (fancy ones)

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

4

Where are OS's Used?



- In the future also:
 - Digital Cameras (fancy ones)
 - MP3 Players (iPods, etc.)
 - Refrigerators!
- Others?

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

5

Layers of a Computer System

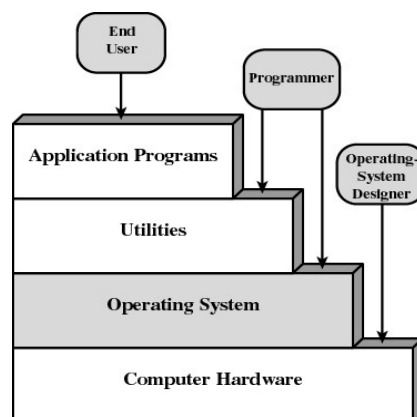


Figure 2.1 Layers and Views of a Computer System

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

6

Services Provided by the OS



- Program execution
- Access to Input/Output (I/O) devices
 - Disks, screens, keyboards, mice
 - Printers, cameras, speakers, etc.
- Controlled access to files
- System access
- Sometimes: Program development
 - Compilers, editors and debuggers

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

7

Services Provided by the OS



- Error detection and response
 - Internal and external hardware errors
 - Memory errors
 - Device failures
 - Software errors
 - Arithmetic overflow
 - Division by zero
 - Access to forbidden memory locations (why?)
 - Operating system cannot grant request of application

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

8

Services Provided by the OS



- Accounting
 - Collect statistics
 - Monitor performance
 - Used to anticipate future enhancements
 - Used for billing users

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

9

Operating System as Software

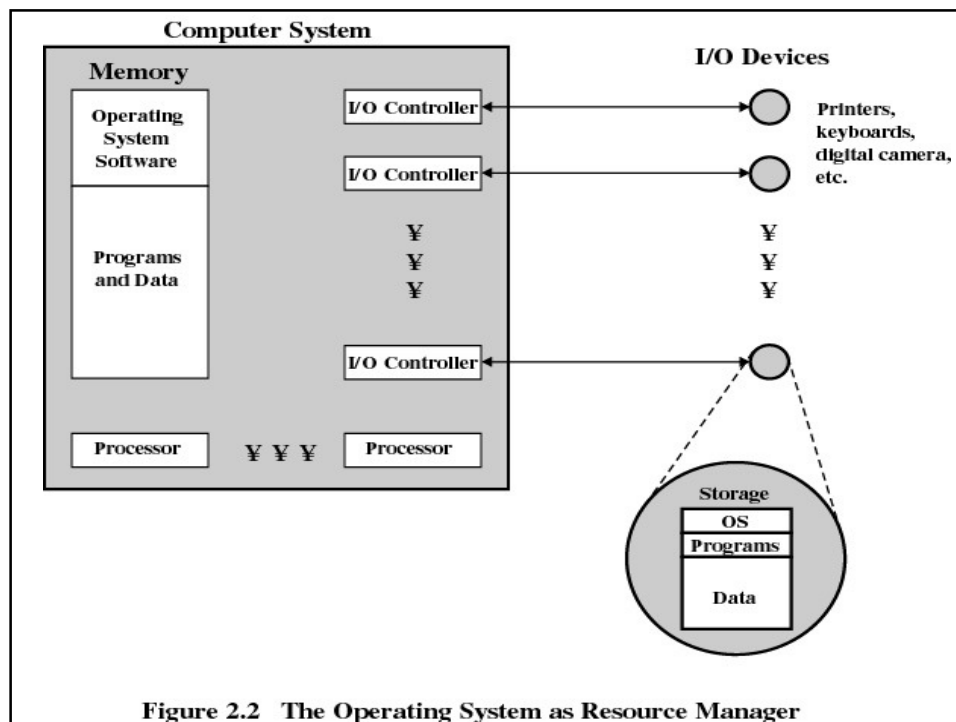


- The OS functions the same way as ordinary computer software
 - It is a program that is executed ...
 - ... but it has special privileges
- The OS relinquishes control of the processor to execute other programs

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

10



Kernel



- Portion of operating system that is always in main memory
- Contains most-frequently used functions
- Also called the nucleus
- Good performance of the kernel is very important

Before Operating Systems...



- Serial Processing
 - No operating system
 - Machines run from a console with display lights and toggle switches, input device, and printer
 - Schedule time
 - Setup included loading the compiler, source program, saving compiled program, and loading and linking

11 February 2005

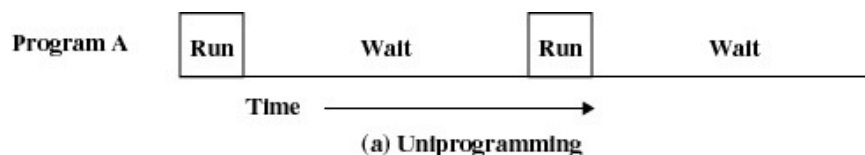
Mads Haahr (with material by William Stallings and Patty Roy)

13

Uniprogramming



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



- But! Processors are much faster than I/O devices... → inefficient use of CPU

11 February 2005

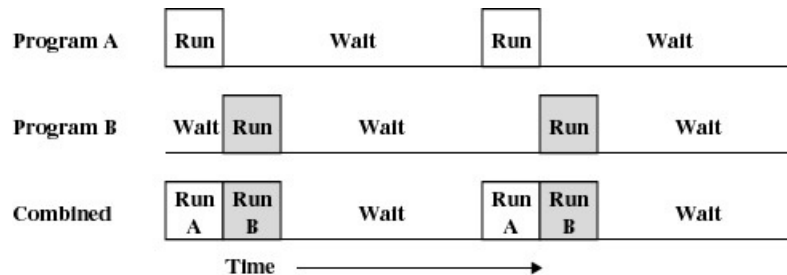
Mads Haahr (with material by William Stallings and Patty Roy)

14

Multiprogramming



- When one job needs to wait for I/O, the processor can switch to the other job



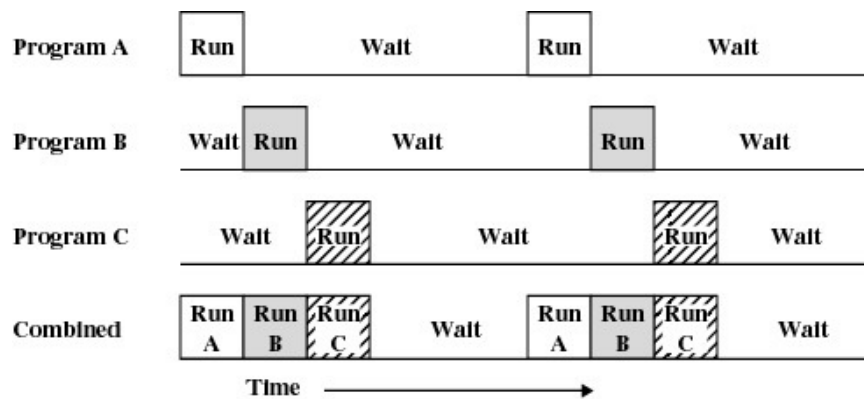
(b) Multiprogramming with two programs

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

15

Multiprogramming



(c) Multiprogramming with three programs

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

16

A More Advanced Example



	Job 1	Job 2	Job 3
Job type	heavy compute	heavy I/O	heavy I/O
Duration	5 minutes	15 minutes	10 minutes
CPU req'd	80%	10%	10%
Memory req'd	50 MB	100 MB	80 MB
Need disk?	no	no	yes
Need terminal?	no	yes	no
Need printer?	no	no	yes

↑
e.g., image analysis

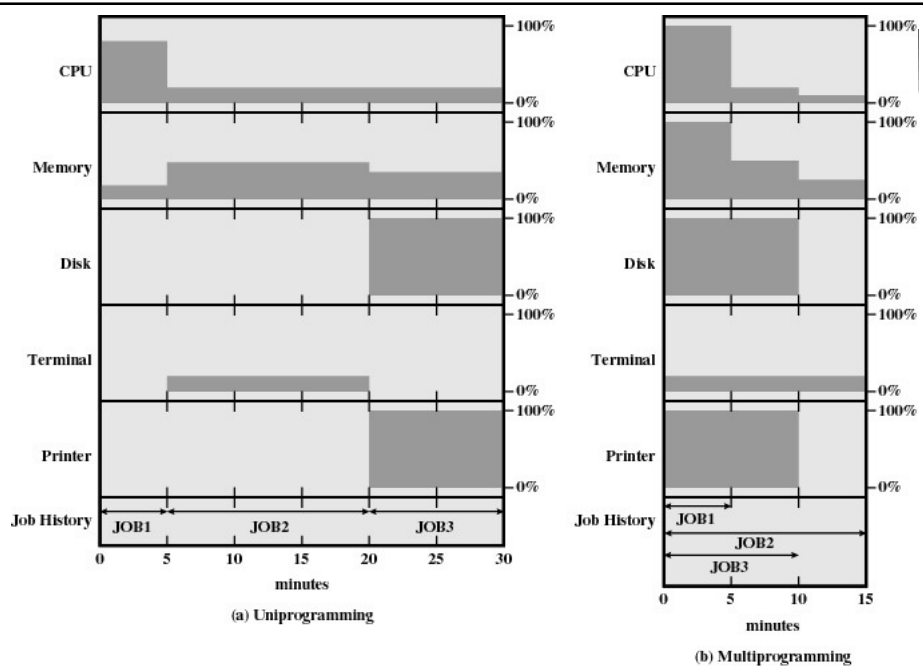
↑
e.g., visualisation

↑
e.g., printing from disk

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

17



Example Effects



	Uniprogramming	Multiprogramming
Processor use	22%	43%
Memory use	30%	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.	10 min.

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

19

Achievements in OS Research



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

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

20

Processes



- A program in execution
- An instance of a program running on a computer
- The entity that can be assigned to and executed on a processor
- A unit of activity characterized by
 - a single sequential thread of execution
 - a current state
 - an associated set of system resources

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

21

Processes



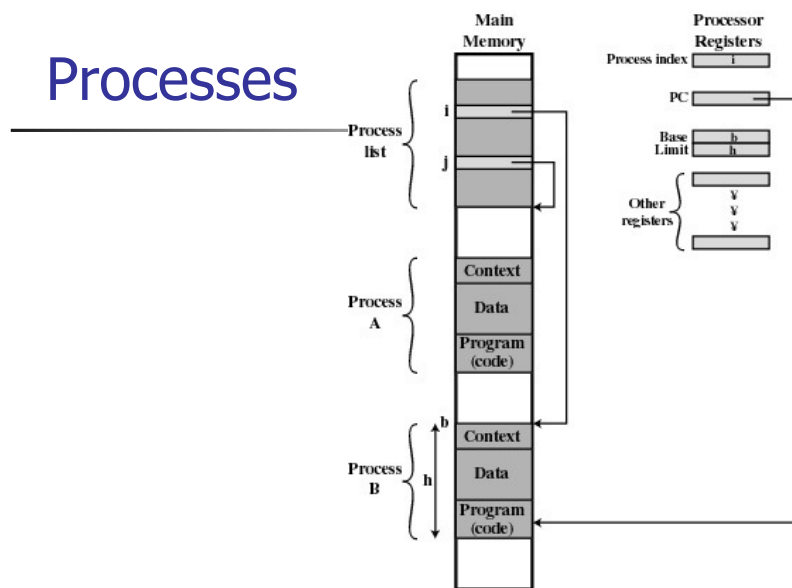
- Consists of three components
 - An executable program, i.e., some code
 - Associated data needed by the program
 - Execution context of the program
 - All information the operating system needs to manage the process
 - e.g., who owns the process, which priority does it have, what resources does it currently 'own'

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

22

Processes



11 February 2005

Mads Haahr (wit)

Figure 2.8 Typical Process Implementation

OS Support for Processes



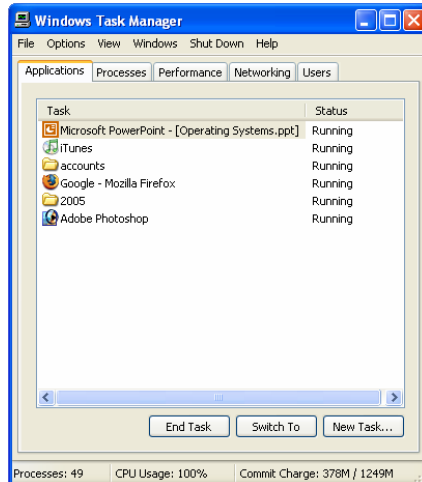
- Creating, destroying, starting, stopping, setting priorities, ...
- Process synchronisation
 - Semaphores (wait/signal)
 - Monitors
- Deadlock detection
- Inter-process communication
 - Shared memory

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

24

Processes in Windows XP



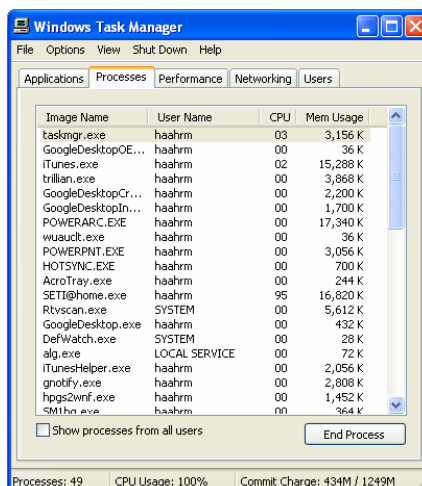
- Hit CTRL-ALT-DEL to start Task Manager
- Shows info about the PC's performance
- Info gathered by OS; updated continuously
- The Applications tab shows running applications and allows them to be ended

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

25

Processes in Windows XP



- The Processes tab shows the running processes
- CPU and memory usage
- Applications often consist of a number of processes
- Also, many system processes
- Individual processes can be ended → dangerous

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

26

Processes in Linux



The screenshot shows the KPM window with system statistics at the top and a list of processes below. The system statistics include CPU, memory, and swap usage. The process list shows columns for PID, USER, PRI, NICE, SIZE, RSS, SHARE, STAT, %CPU, %MEM, TIME, and CMDLINE.

PID	USER	PRI	NICE	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	CMDLINE
561	root	11	-10356252	40400	1816	S	<	1.38	15.85	55.55h	/usr/X11R6/bin/X -dpi 100 -nolisten tcp vt17 -auth /var/
631	haahr	19	0	3224	2444	1096	R	1.98	0.96	83.41	/usr/bin/atsd -F 10 -S 4096 -s 60 -m artsmessage -l 3
29549	haahr	15	0	6896	6892	5760	R	0.99	2.70	2.93s	kpm -icon kpm -minicon kpm
635	haahr	10	0	3900	3300	2240	S	0.20	1.29	4.41	kdeinit: kwin-session 1186e224420001029924315000
29554	haahr	12	0	8500	8500	3944	S	0.40	3.33	1.78s	gimp
12889	haahr	9	0	57824	43132	10548	S	0.00	16.92	19.29	/usr/lib/mozilla-firefox/firefox-bin -a firefox
639	haahr	9	0	13092	12032	3112	S	0.20	4.72	10.47	kdeinit: kicker
633	haahr	10	0	2656	1824	1452	S	0.20	0.72	3.06	kdeinit: knotify
637	haahr	9	0	3860	3228	2436	S	0.00	1.27	5.12	kdeinit: kdesktop
6	root	9	0	0	0	0	SW	0.00	0.00	2.32h	(kupdated)
29555	haahr	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	haahr	9	0	1408	808	676	S	0.00	0.32	6.78s	kdeinit: dcooserver --nosid
22103	haahr	9	0	1560	1340	1020	S	0.00	0.53	0.51s	xterm
11037	haahr	9	0	1608	708	548	S	0.00	0.28	4.40s	kdeinit: kcookiejar
18358	root	9	0	1064	748	596	S	0.00	0.29	9.06	/usr/sbin/nmbd -D
4	root	9	0	0	0	0	SW	0.00	0.00	89.38	(kswapd)
24255	haahr	9	0	23940	10132	2572	S	0.00	3.97	3.30	pine
22104	haahr	8	0	1352	1036	660	S	0.00	0.41	0.13s	-sh
12921	haahr	9	0	57824	43132	10548	S	0.00	16.92	9.71s	/usr/lib/mozilla-firefox/firefox-bin -a firefox
2	root	9	0	0	0	0	SW	0.00	0.00	0.33	(kewntd)
355	root	9	0	17480	1784	520	S	0.00	0.70	1.17	/usr/lib/X11/xfs-daemon
12923	haahr	9	0	57824	43132	10548	S	0.00	16.92	0.15	/usr/lib/mozilla-firefox/firefox-bin -a firefox
386	root	9	0	0	0	0	SW	0.00	0.00	3.53	(rpciod)
27413	haahr	9	0	1284	956	788	S	0.00	0.37	0.11s	xterm
650	haahr	9	0	1864	1064	748	S	0.00	0.42	10.51	xterm

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

27

Memory Management



- Process isolation
 - Processes do not share memory
 - This prevents processes from affecting each other
- Protection and access control
- Automatic allocation and management

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

28

Virtual Memory and Paging



- Virtual Memory
 - Provides a logical rather than actual view of memory
 - Allows process to be comprised of a number of fixed-size blocks, called pages
- Virtual address
 - A page number and an offset within the page
 - Each page may be located anywhere in main memory
- Real address
 - The physical address in main memory

11 February 2005

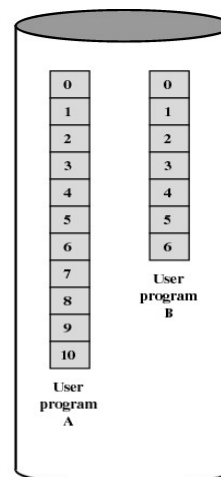
Mads Haahr (with material by William Stallings and Patty Roy)

29

A.1			
	A.0	A.2	
	A.5		
B.0	B.1	B.2	B.3
		A.7	
	A.9		
		A.8	
B.4	B.5	B.6	

Main Memory

Main memory consists of a number of fixed-length frames, equal to the size of a page. For a program to execute, some or all of its pages must be in main memory.



Disk

Secondary memory (disk) can hold many fixed-length pages. A user program consists of some number of pages. Pages for all programs plus the operating system are on disk, as are files.

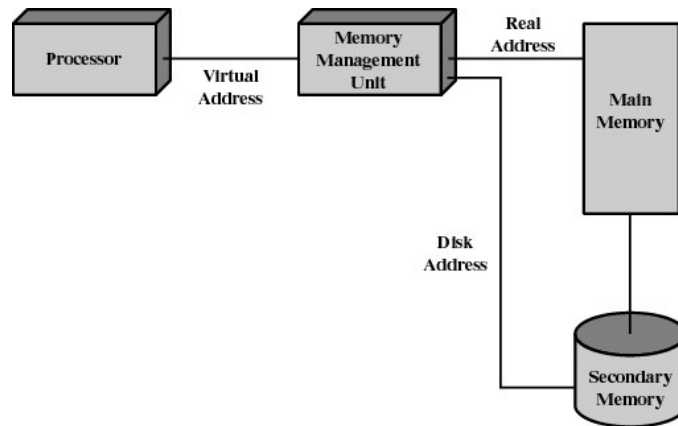


11 February 2005

Figure 2.9 Virtual Memory Concepts

30

Virtual Memory Addressing

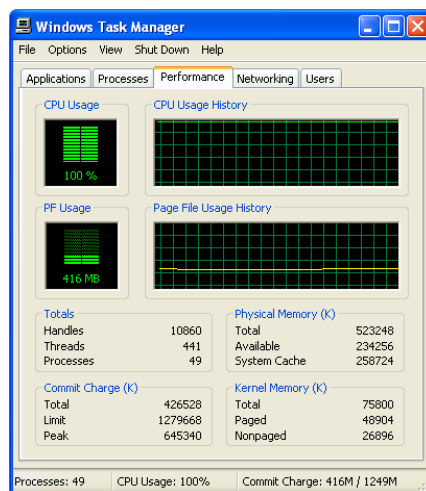


11 February 20

Figure 2.10 Virtual Memory Addressing

31

Memory in Windows XP



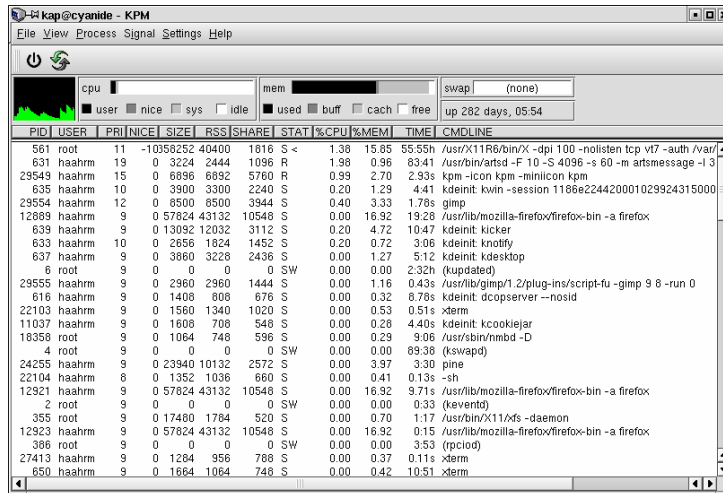
- The Performance tab shows CPU and virtual memory (page file) usage over time
- Memory statistics also shown
- SETI@home client used my spare CPU cycles in example
- Typically, workstations use very few CPU cycles

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

32

Memory in Linux



11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

33

File System



- Implements long-term store
- Information stored in named objects called files
- OS typically offers the following file operations:
 - creating, deleting, renaming
 - reading, writing
 - locking, unlocking

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

34

Information Protection and Security



- Access control
 - regulate user access to the system
- Information flow control
 - regulate flow of data within the system and its delivery to users
- Certification
 - proving that access and flow control perform according to specifications

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

35

Scheduling and Resource Management



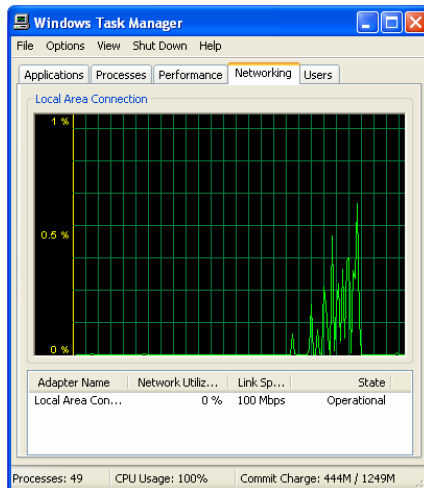
- Fairness
 - give equal and fair access to all processes
- Differential responsiveness
 - discriminate between different classes of jobs
 - e.g., long-running number crunching jobs vs high-priority interactive jobs
- Efficiency
 - maximize throughput, minimize response time, and accommodate as many uses as possible

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

36

Windows XP



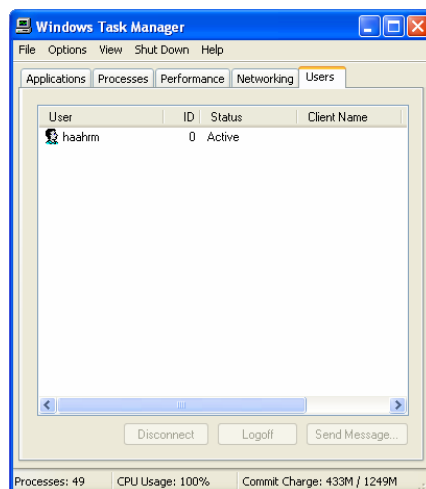
- The Networking tab shows network bandwidth used over time
- Several network interfaces ('connections' here) are shown separately

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

37

Windows XP



- The Users tab shows the users currently logged in

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

38

Major OS Elements

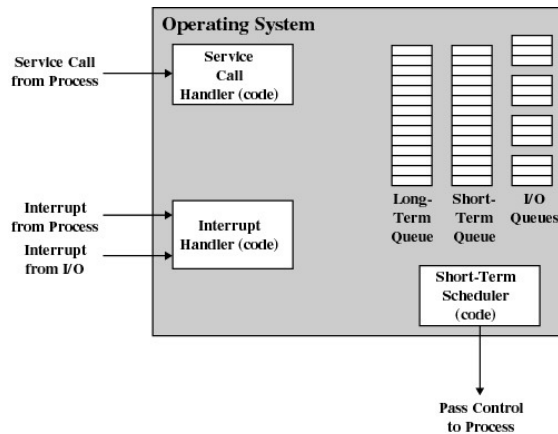


Figure 2.11 Key Elements of an Operating System for Multiprogramming

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

39

System Structure



- View the system as a series of levels
- Each level performs a related subset of functions
- Each level relies on the next lower level to perform more primitive functions
- This decomposes a problem into a number of more manageable subproblems

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

40

Modern Operating Systems



- Multithreading
 - Each process is divided into threads that can run simultaneously
- Thread
 - Dispatchable unit of work
 - Executes sequentially and is interruptible
- A process is a collection of one or more threads

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

41

Modern Operating Systems



- Symmetric multiprocessing
 - There are multiple processors
 - These processors share same main memory and I/O facilities
 - All processors can perform the same functions

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

42

Modern Operating Systems



- Distributed operating systems
 - Provides the illusion of a single main memory and single secondary memory space
 - Distributed shared memory
 - Distributed file systems

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

43

Example OS: PalmOS



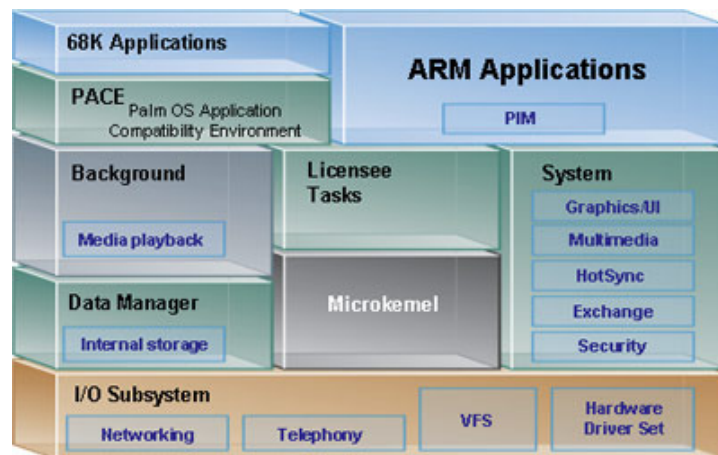
- Used for PalmPilot PDAs and successors
- Multitasking since PalmOS 5
- CPUs: Intel XScale, Texas Instruments OMAP, Motorola Dragonball MX
- Wireless: 802.11b, Bluetooth, GSM, CDMA
- 320×320+ displays
- Good battery utilisation

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

44

Example OS: PalmOS



11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

45

Example OS: Symbian OS



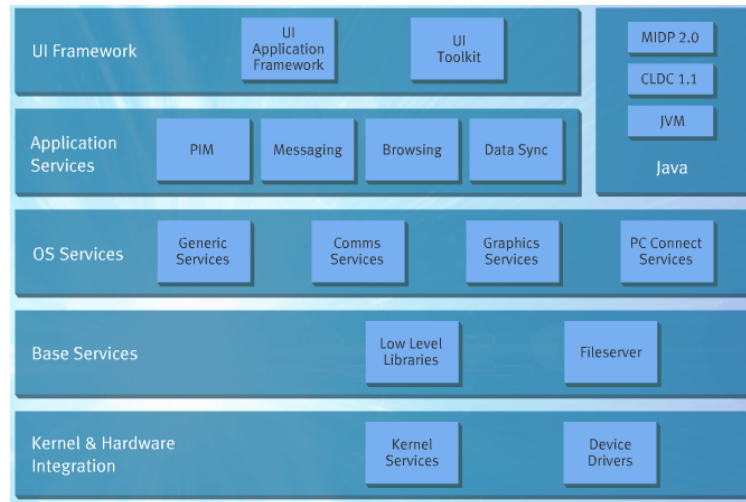
- Designed for mobile phones
- Gives access to graphics, multimedia, networking, telephony, crypto, PC connectivity, etc.

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

46

Example OS: Symbian OS



11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

47

The OS Wars



- Why is the OS such a big deal?
 - Windows vs Linux vs Mac
 - Symbian vs PalmOS vs Windows Mobile
- Incompatibilities
 - OS's have different interfaces
 - → programs must be written differently
 - → applications for one OS don't run on another
- Tendency to bloatware
 - Applications tend to move into the OS
 - Internet Explorer, Media Player, Search?

11 February 2005

Mads Haahr (with material by William Stallings and Patty Roy)

48

Security: Windows vs Linux



- Windows Family
 - Developed from DOS
 - Originally single-user machines
 - No network → few threats
- Linux
 - Developed from Unix
 - Originally multi-user networked servers
 - Designed to withstand with security threats from the beginning

Further Reading



- William Stallings. *Operating Systems: Internals and Design Principles, 4th edition*. Prentice Hall, 2001.
- Andrew Tanenbaum. *Modern Operating Systems, 2nd edition*. Prentice Hall, 2001.