

Operating-System Structures

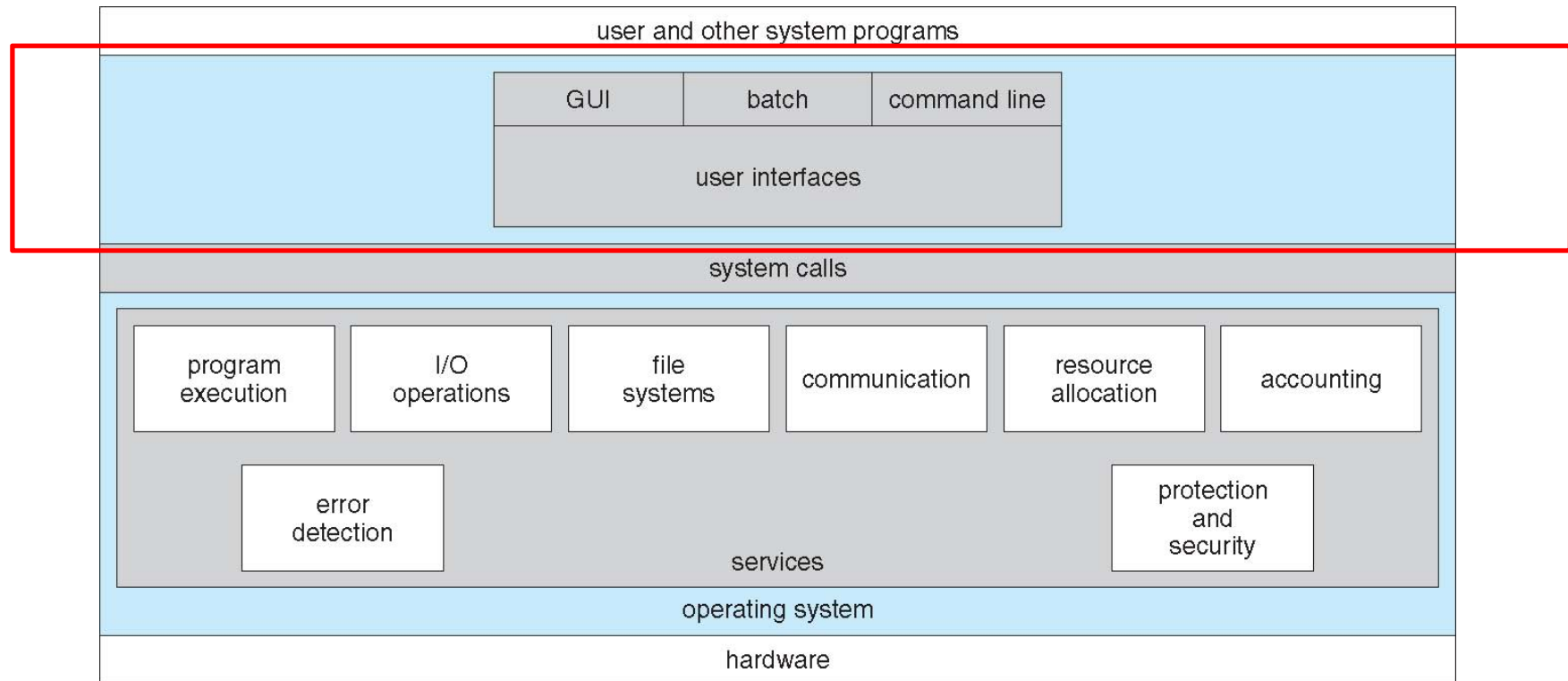
Fan Wu

Department of Computer Science and Engineering

Shanghai Jiao Tong University

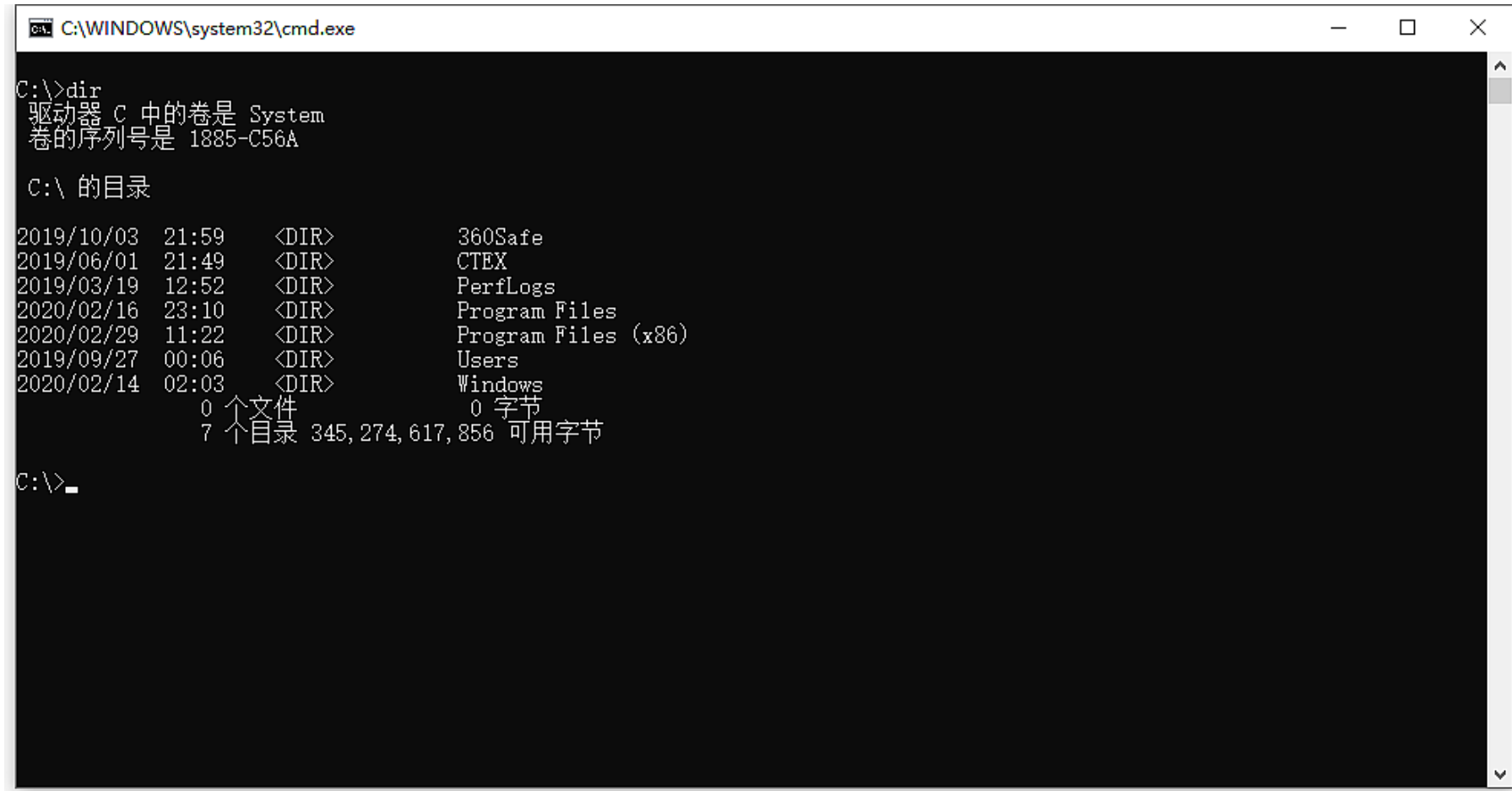
Spring 2022

Operating System Services Structure



Operating systems provide an environment for execution of programs and services to programs and users

Windows/DOS Command Line Interface



A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt shows the command 'C:\>dir' and its output. The output indicates that the drive C: is a System volume with serial number 1885-C56A. It then lists the contents of the C:\ directory, showing several subdirectories with their creation dates and times. At the bottom, it shows the total number of files and directories and the available space in bytes.

```
C:\>dir
驱动器 C 中的卷是 System
卷的序列号是 1885-C56A

C:\ 的目录

2019/10/03  21:59    <DIR>          360Safe
2019/06/01  21:49    <DIR>          CTEX
2019/03/19  12:52    <DIR>          PerfLogs
2020/02/16  23:10    <DIR>          Program Files
2020/02/29  11:22    <DIR>          Program Files (x86)
2019/09/27  00:06    <DIR>          Users
2020/02/14  02:03    <DIR>          Windows
               0 个文件             0 字节
               7 个目录  345,274,617,856 可用字节

C:\>_
```

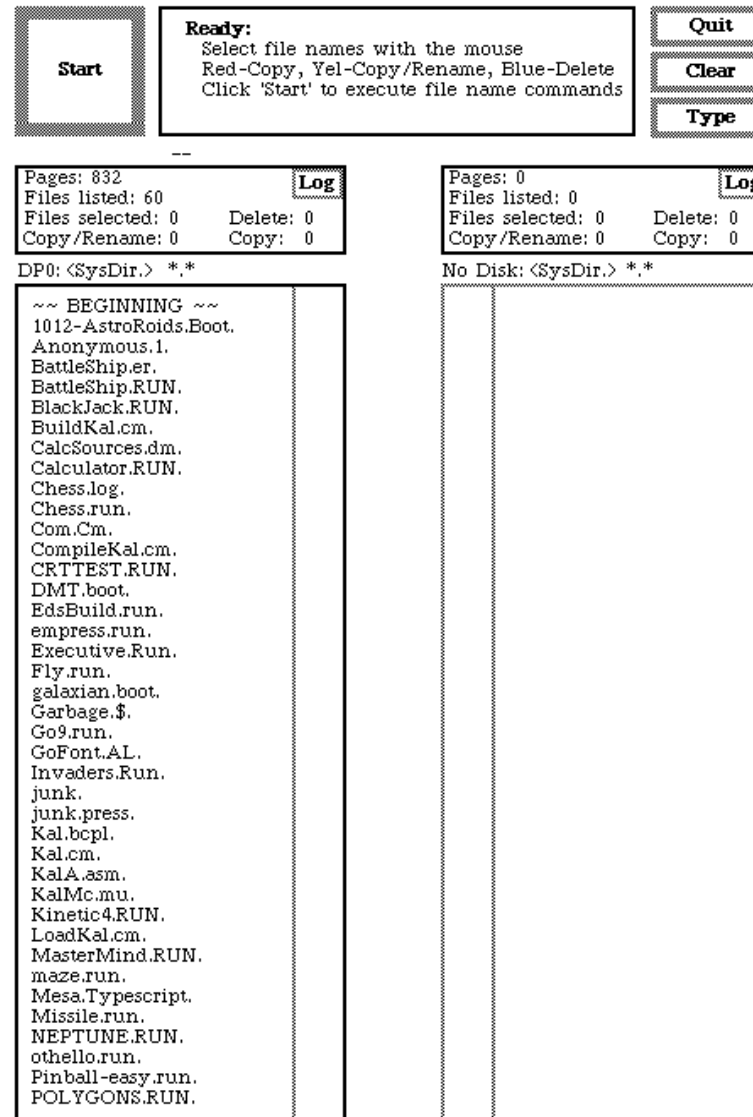
Bash shell in MacOS

```
1. root@r6181-d5-us01:~ (ssh)
X root@r6181-d5-u... 1 X ssh 2 X root@r6181-d5-us01... 3
Last login: Thu Jul 14 08:47:01 on ttys002
iMacPro:~ pbg$ ssh root@r6181-d5-us01
root@r6181-d5-us01's password:
Last login: Thu Jul 14 06:01:11 2016 from 172.16.16.162
[root@r6181-d5-us01 ~]# uptime
 06:57:48 up 16 days, 10:52,  3 users,  load average: 129.52, 80.33, 56.55
[root@r6181-d5-us01 ~]# df -kh
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg_ks-lv_root
                  50G   19G   28G  41% /
tmpfs            127G  520K  127G   1% /dev/shm
/dev/sda1        477M   71M  381M  16% /boot
/dev/dssd0000    1.0T  480G  545G  47% /dssd_xfs
tcp://192.168.150.1:3334/orangefs
                  12T   5.7T   6.4T  47% /mnt/orangefs
/dev/gpfs-test   23T   1.1T   22T   5% /mnt/gpfs
[root@r6181-d5-us01 ~]#
[root@r6181-d5-us01 ~]# ps aux | sort -nrk 3,3 | head -n 5
root      97653 11.2  6.6 42665344 17520636 ?    S<Ll  Jul13 166:23 /usr/lpp/mmfs/bin/mmfsd
root      69849  6.6  0.0      0      0 ?        S    Jul12 181:54 [vpthread-1-1]
root      69850  6.4  0.0      0      0 ?        S    Jul12 177:42 [vpthread-1-2]
root       3829  3.0  0.0      0      0 ?        S    Jun27 730:04 [rp_thread 7:0]
root       3826  3.0  0.0      0      0 ?        S    Jun27 728:08 [rp_thread 6:0]
[root@r6181-d5-us01 ~]# ls -l /usr/lpp/mmfs/bin/mmfsd
-r-x----- 1 root root 20667161 Jun  3  2015 /usr/lpp/mmfs/bin/mmfsd
[root@r6181-d5-us01 ~]#
```

Bourne Shell Command Interpreter

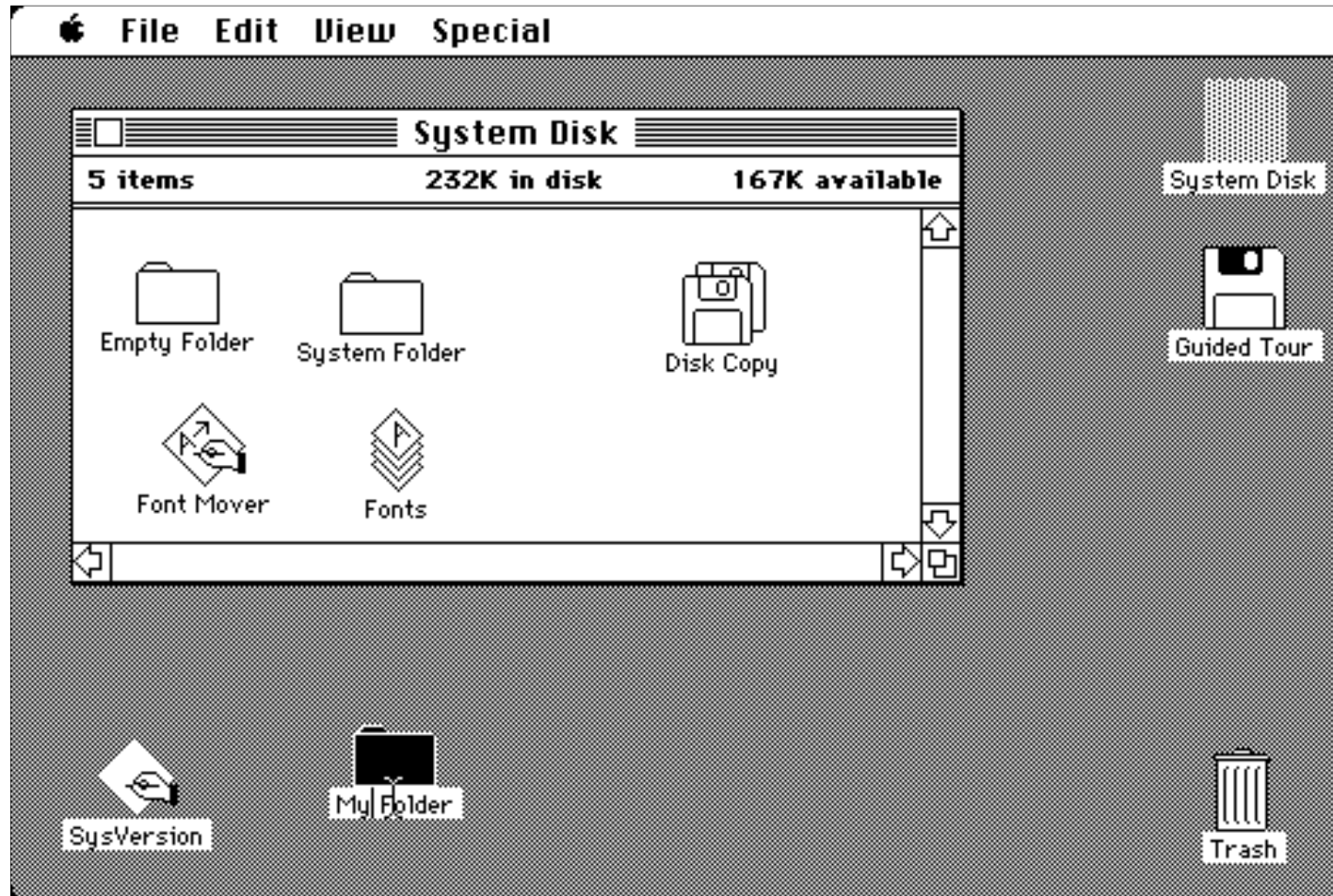
```
Terminal
File Edit View Terminal Tabs Help
fd0      0.0    0.0    0.0    0.0  0.0  0.0    0.0  0  0
sd0      0.0    0.2    0.0    0.2  0.0  0.0    0.4  0  0
sd1      0.0    0.0    0.0    0.0  0.0  0.0    0.0  0  0
          extended device statistics
device   r/s    w/s    kr/s    kw/s wait actv  svc_t  %w  %b
fd0      0.0    0.0    0.0    0.0  0.0  0.0    0.0  0  0
sd0      0.6    0.0   38.4    0.0  0.0  0.0    8.2  0  0
sd1      0.0    0.0    0.0    0.0  0.0  0.0    0.0  0  0
(root@pbg-nv64-vn)-(11/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# swap -sh
total: 1.1G allocated + 190M reserved = 1.3G used, 1.6G available
(root@pbg-nv64-vn)-(12/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# uptime
12:53am up 9 min(s), 3 users, load average: 33.29, 67.68, 36.81
(root@pbg-nv64-vn)-(13/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# w
4:07pm up 17 day(s), 15:24, 3 users, load average: 0.09, 0.11, 8.66
User      tty          login@ idle   JCPU   PCPU   what
root      console      15Jun07 18days 1      /usr/bin/ssh-agent -- /usr/bi
n/d
root      pts/3        15Jun07 18      4      w
root      pts/4        15Jun07 18days 1      w
(root@pbg-nv64-vn)-(14/pts)-(16:07 02-Jul-2007)-(global)
-(/var/tmp/system-contents/scripts)#
```

First GUI (1973)

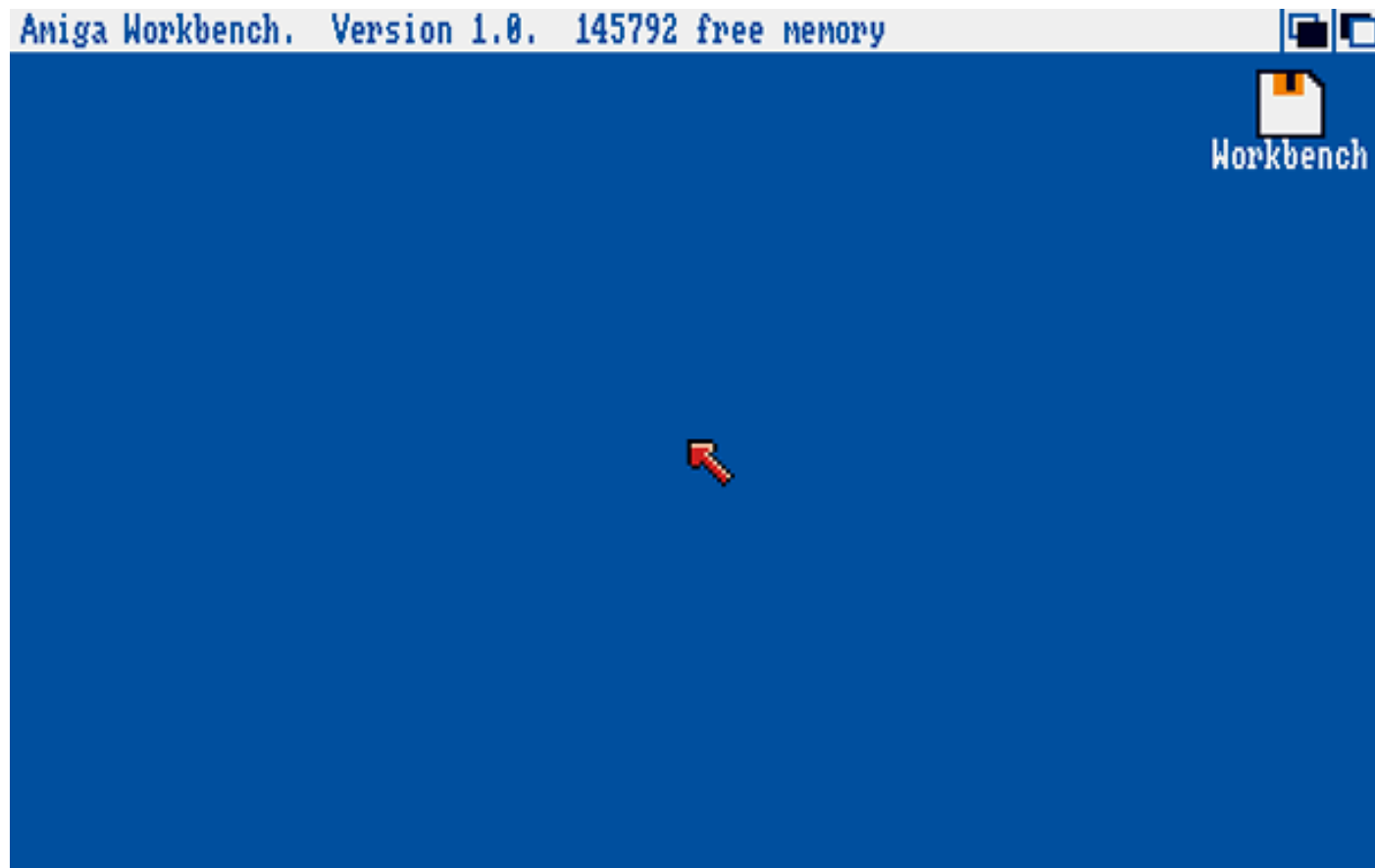


The first appeared
on the Xerox Alto
computer in 1973.

Mac OS System 1.0 (1984)

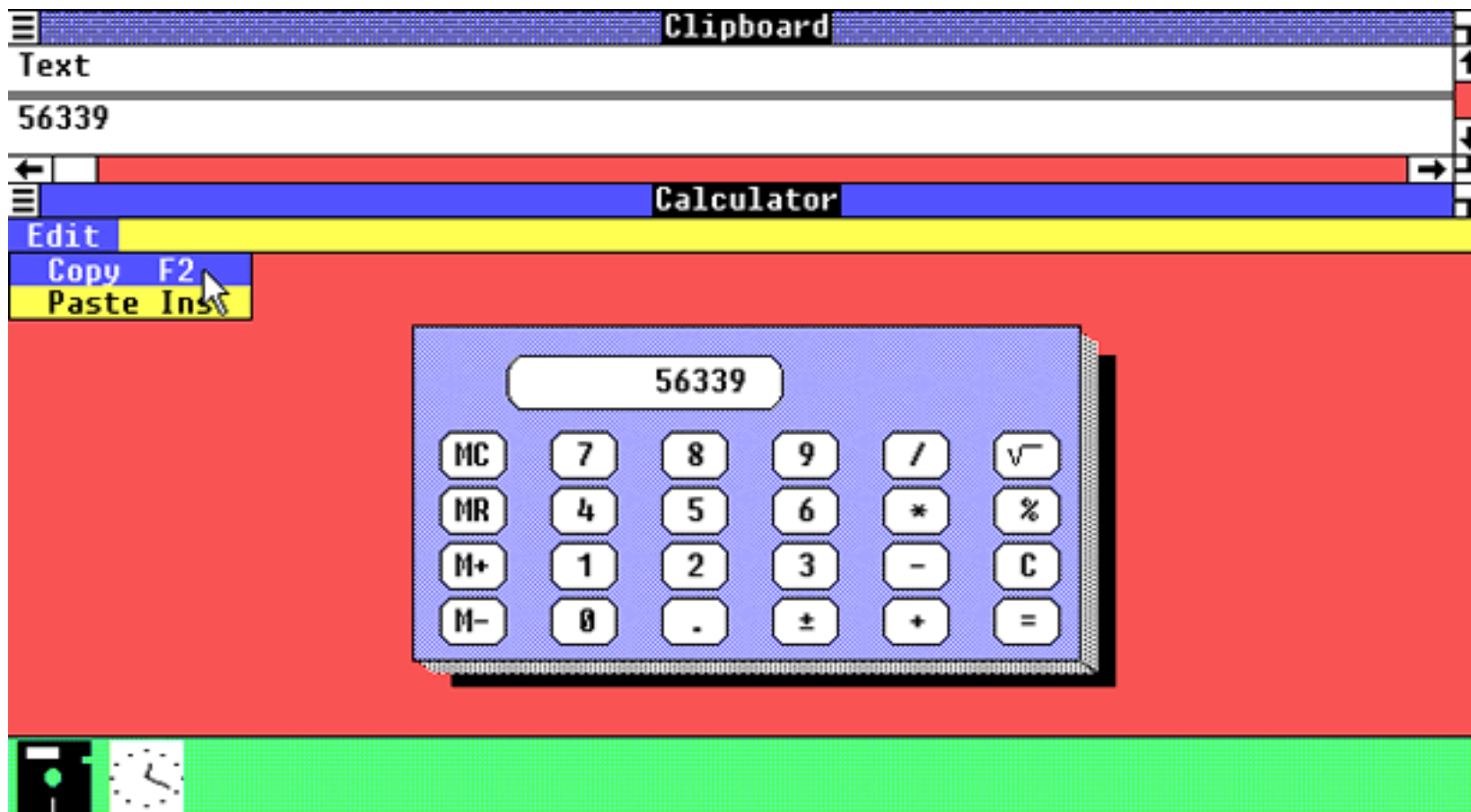


Amiga Workbench 1.0 (1985)

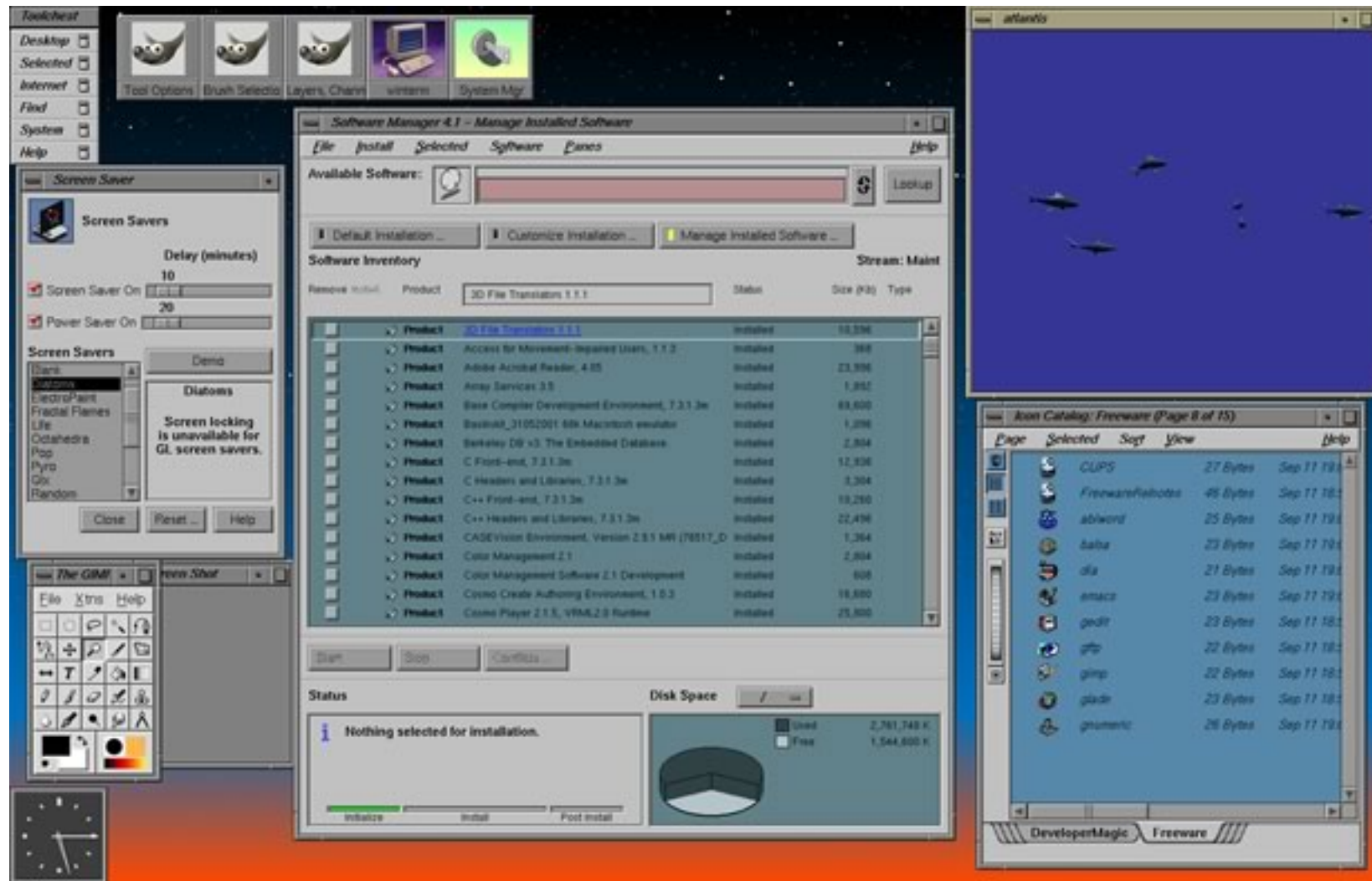


The first GUI with color graphics.

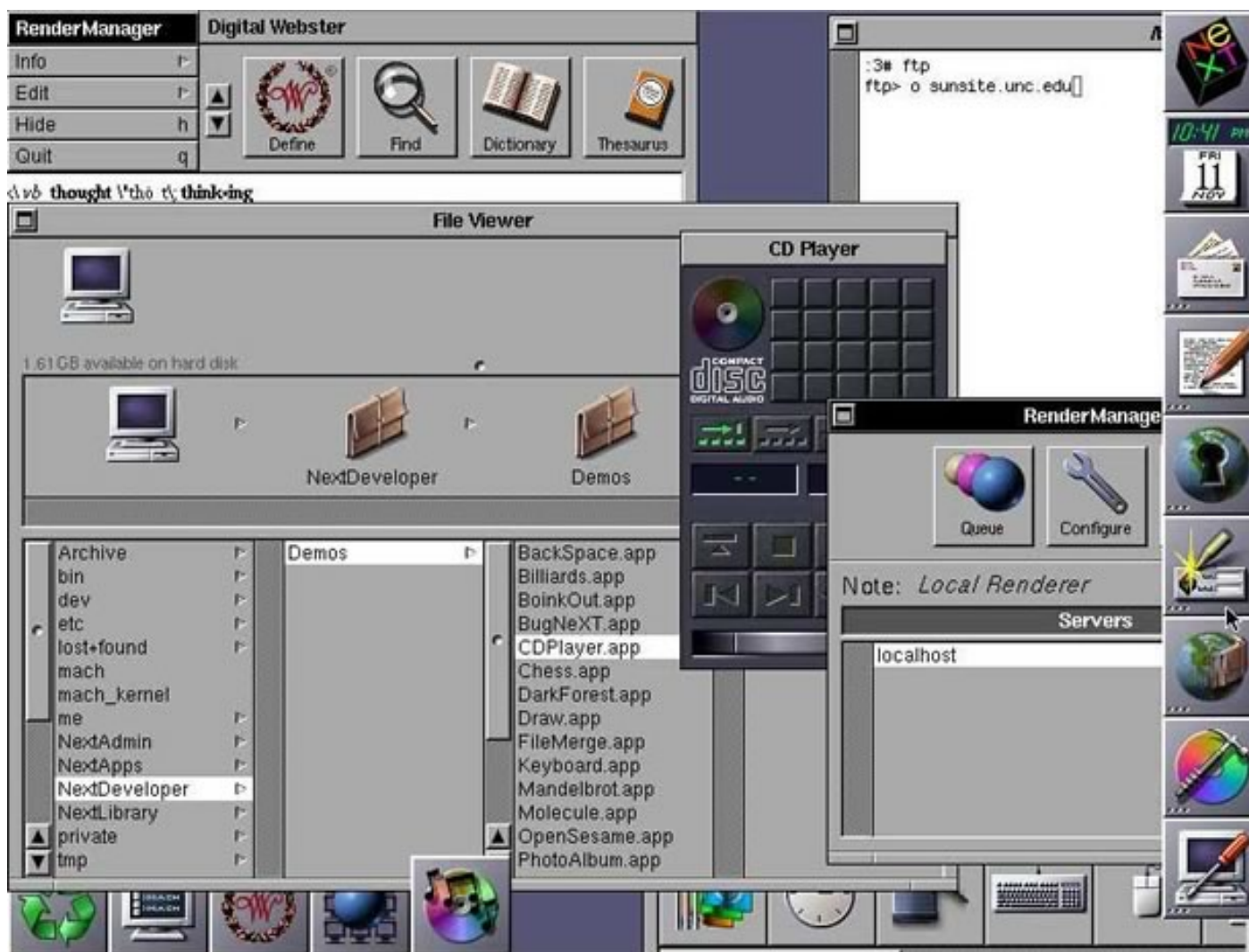
Windows 1.0x (1985)



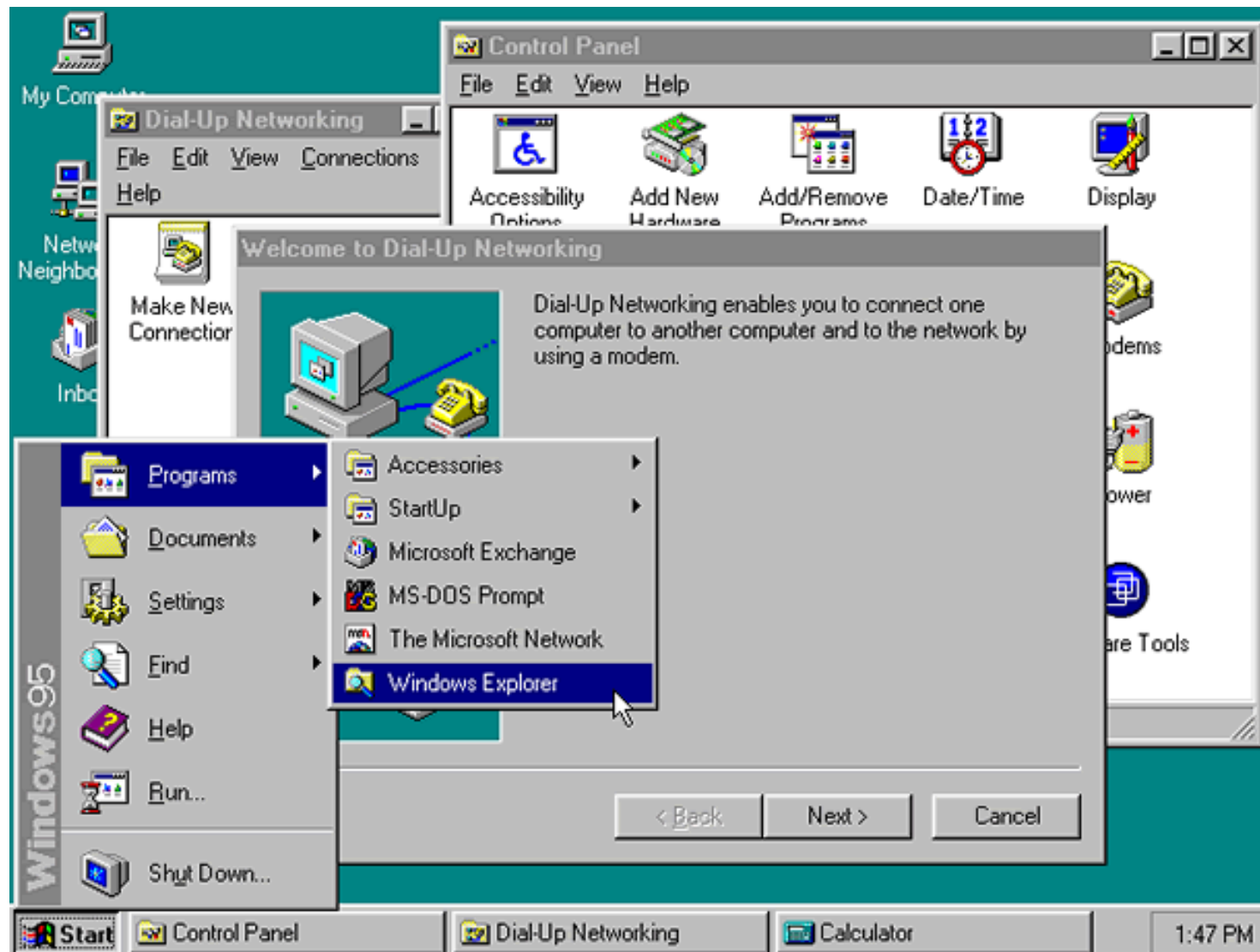
IRIX 3 (released in 1986, first release 1984)



NeXTSTEP / OPENSTEP 1.0 (1989)

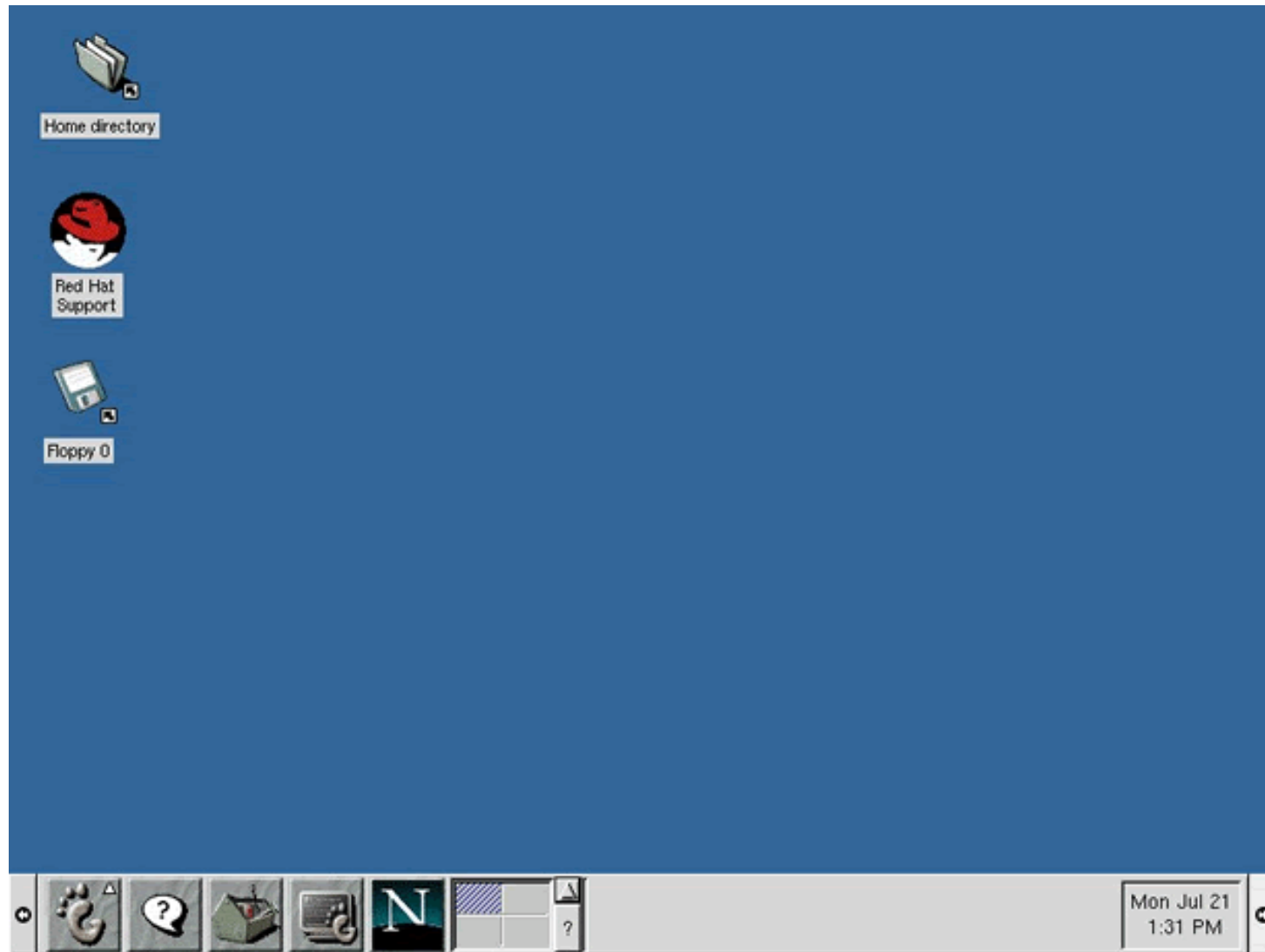


Windows 95 (1995)



The screenshot displays a KDE desktop environment. At the top, a taskbar shows several open applications: emacs, konsole, file manager, and others. The desktop background is a blue abstract pattern. On the left side, there is a vertical dock with icons for Trash, Templates, Autostart, netcage, kmail, kdel, k3b, k3d, k3t, and Super User. In the center, a terminal window is open, showing the command prompt and the output of the 'ls' command in the directory /usr/lib/qt2/. The output lists various files and directories including LICENSE.QPL, Makefile, bin, examples, configs, doc, extensions, gif, html, include, lib, propagate, src, tutorial, and variables. To the right of the terminal, a file manager window is open, displaying the contents of the /usr/lib/qt2/ directory. The file manager shows a list of files and folders, including CXX, Kde4, kdelibs, kdelibs4, kdelibs4c1, kdelibs4c2, kdelibs4c3, kdelibs4c4, kdelibs4c5, kdelibs4c6, kdelibs4c7, kdelibs4c8, kdelibs4c9, kdelibs4c10, kdelibs4c11, kdelibs4c12, kdelibs4c13, kdelibs4c14, kdelibs4c15, kdelibs4c16, kdelibs4c17, kdelibs4c18, kdelibs4c19, kdelibs4c20, kdelibs4c21, kdelibs4c22, kdelibs4c23, kdelibs4c24, kdelibs4c25, kdelibs4c26, kdelibs4c27, kdelibs4c28, kdelibs4c29, kdelibs4c30, kdelibs4c31, kdelibs4c32, kdelibs4c33, kdelibs4c34, kdelibs4c35, kdelibs4c36, kdelibs4c37, kdelibs4c38, kdelibs4c39, kdelibs4c40, kdelibs4c41, kdelibs4c42, kdelibs4c43, kdelibs4c44, kdelibs4c45, kdelibs4c46, kdelibs4c47, kdelibs4c48, kdelibs4c49, kdelibs4c50, kdelibs4c51, kdelibs4c52, kdelibs4c53, kdelibs4c54, kdelibs4c55, kdelibs4c56, kdelibs4c57, kdelibs4c58, kdelibs4c59, kdelibs4c60, kdelibs4c61, kdelibs4c62, kdelibs4c63, kdelibs4c64, kdelibs4c65, kdelibs4c66, kdelibs4c67, kdelibs4c68, kdelibs4c69, kdelibs4c70, kdelibs4c71, kdelibs4c72, kdelibs4c73, kdelibs4c74, kdelibs4c75, kdelibs4c76, kdelibs4c77, kdelibs4c78, kdelibs4c79, kdelibs4c80, kdelibs4c81, kdelibs4c82, kdelibs4c83, kdelibs4c84, kdelibs4c85, kdelibs4c86, kdelibs4c87, kdelibs4c88, kdelibs4c89, kdelibs4c90, kdelibs4c91, kdelibs4c92, kdelibs4c93, kdelibs4c94, kdelibs4c95, kdelibs4c96, kdelibs4c97, kdelibs4c98, kdelibs4c99, kdelibs4c100, kdelibs4c101, kdelibs4c102, kdelibs4c103, kdelibs4c104, kdelibs4c105, kdelibs4c106, kdelibs4c107, kdelibs4c108, kdelibs4c109, kdelibs4c110, kdelibs4c111, kdelibs4c112, kdelibs4c113, kdelibs4c114, kdelibs4c115, kdelibs4c116, kdelibs4c117, kdelibs4c118, kdelibs4c119, kdelibs4c120, kdelibs4c121, kdelibs4c122, kdelibs4c123, kdelibs4c124, kdelibs4c125, kdelibs4c126, kdelibs4c127, kdelibs4c128, kdelibs4c129, kdelibs4c130, kdelibs4c131, kdelibs4c132, kdelibs4c133, kdelibs4c134, kdelibs4c135, kdelibs4c136, kdelibs4c137, kdelibs4c138, kdelibs4c139, kdelibs4c140, kdelibs4c141, kdelibs4c142, kdelibs4c143, kdelibs4c144, kdelibs4c145, kdelibs4c146, kdelibs4c147, kdelibs4c148, kdelibs4c149, kdelibs4c150, kdelibs4c151, kdelibs4c152, kdelibs4c153, kdelibs4c154, kdelibs4c155, kdelibs4c156, kdelibs4c157, kdelibs4c158, kdelibs4c159, kdelibs4c160, kdelibs4c161, kdelibs4c162, kdelibs4c163, kdelibs4c164, kdelibs4c165, kdelibs4c166, kdelibs4c167, kdelibs4c168, kdelibs4c169, kdelibs4c170, kdelibs4c171, kdelibs4c172, kdelibs4c173, kdelibs4c174, kdelibs4c175, kdelibs4c176, kdelibs4c177, kdelibs4c178, kdelibs4c179, kdelibs4c180, kdelibs4c181, kdelibs4c182, kdelibs4c183, kdelibs4c184, kdelibs4c185, kdelibs4c186, kdelibs4c187, kdelibs4c188, kdelibs4c189, kdelibs4c190, kdelibs4c191, kdelibs4c192, kdelibs4c193, kdelibs4c194, kdelibs4c195, kdelibs4c196, kdelibs4c197, kdelibs4c198, kdelibs4c199, kdelibs4c200, kdelibs4c201, kdelibs4c202, kdelibs4c203, kdelibs4c204, kdelibs4c205, kdelibs4c206, kdelibs4c207, kdelibs4c208, kdelibs4c209, kdelibs4c210, kdelibs4c211, kdelibs4c212, kdelibs4c213, kdelibs4c214, kdelibs4c215, kdelibs4c216, kdelibs4c217, kdelibs4c218, kdelibs4c219, kdelibs4c220, kdelibs4c221, kdelibs4c222, kdelibs4c223, kdelibs4c224, kdelibs4c225, kdelibs4c226, kdelibs4c227, kdelibs4c228, kdelibs4c229, kdelibs4c230, kdelibs4c231, kdelibs4c232, kdelibs4c233, kdelibs4c234, kdelibs4c235, kdelibs4c236, kdelibs4c237, kdelibs4c238, kdelibs4c239, kdelibs4c240, kdelibs4c241, kdelibs4c242, kdelibs4c243, kdelibs4c244, kdelibs4c245, kdelibs4c246, kdelibs4c247, kdelibs4c248, kdelibs4c249, kdelibs4c250, kdelibs4c251, kdelibs4c252, kdelibs4c253, kdelibs4c254, kdelibs4c255, kdelibs4c256, kdelibs4c257, kdelibs4c258, kdelibs4c259, kdelibs4c260, kdelibs4c261, kdelibs4c262, kdelibs4c263, kdelibs4c264, kdelibs4c265, kdelibs4c266, kdelibs4c267, kdelibs4c268, kdelibs4c269, kdelibs4c270, kdelibs4c271, kdelibs4c272, kdelibs4c273, kdelibs4c274, kdelibs4c275, kdelibs4c276, kdelibs4c277, kdelibs4c278, kdelibs4c279, kdelibs4c280, kdelibs4c281, kdelibs4c282, kdelibs4c283, kdelibs4c284, kdelibs4c285, kdelibs4c286, kdelibs4c287, kdelibs4c288, kdelibs4c289, kdelibs4c290, kdelibs4c291, kdelibs4c292, kdelibs4c293, kdelibs4c294, kdelibs4c295, kdelibs4c296, kdelibs4c297, kdelibs4c298, kdelibs4c299, kdelibs4c300, kdelibs4c301, kdelibs4c302, kdelibs4c303, kdelibs4c304, kdelibs4c305, kdelibs4c306, kdelibs4c307, kdelibs4c308, kdelibs4c309, kdelibs4c310, kdelibs4c311, kdelibs4c312, kdelibs4c313, kdelibs4c314, kdelibs4c315, kdelibs4c316, kdelibs4c317, kdelibs4c318, kdelibs4c319, kdelibs4c320, kdelibs4c321, kdelibs4c322, kdelibs4c323, kdelibs4c324, kdelibs4c325, kdelibs4c326, kdelibs4c327, kdelibs4c328, kdelibs4c329, kdelibs4c330, kdelibs4c331, kdelibs4c332, kdelibs4c333, kdelibs4c334, kdelibs4c335, kdelibs4c336, kdelibs4c337, kdelibs4c338, kdelibs4c339, kdelibs4c340, kdelibs4c341, kdelibs4c342, kdelibs4c343, kdelibs4c344, kdelibs4c345, kdelibs4c346, kdelibs4c347, kdelibs4c348, kdelibs4c349, kdelibs4c350, kdelibs4c351, kdelibs4c352, kdelibs4c353, kdelibs4c354, kdelibs4c355, kdelibs4c356, kdelibs4c357, kdelibs4c358, kdelibs4c359, kdelibs4c360, kdelibs4c361, kdelibs4c362, kdelibs4c363, kdelibs4c364, kdelibs4c365, kdelibs4c366, kdelibs4c367, kdelibs4c368, kdelibs4c369, kdelibs4c370, kdelibs4c371, kdelibs4c372, kdelibs4c373, kdelibs4c374, kdelibs4c375, kdelibs4c376, kdelibs4c377, kdelibs4c378, kdelibs4c379, kdelibs4c380, kdelibs4c381, kdelibs4c382, kdelibs4c383, kdelibs4c384, kdelibs4c385, kdelibs4c386, kdelibs4c387, kdelibs4c388, kdelibs4c389, kdelibs4c390, kdelibs4c391, kdelibs4c392, kdelibs4c393, kdelibs4c394, kdelibs4c395, kdelibs4c396, kdelibs4c397, kdelibs4c398, kdelibs4c399, kdelibs4c400, kdelibs4c401, kdelibs4c402, kdelibs4c403, kdelibs4c404, kdelibs4c405, kdelibs4c406, kdelibs4c407, kdelibs4c408, kdelibs4c409, kdelibs4c410, kdelibs4c411, kdelibs4c412, kdelibs4c413, kdelibs4c414, kdelibs4c415, kdelibs4c416, kdelibs4c417, kdelibs4c418, kdelibs4c419, kdelibs4c420, kdelibs4c421, kdelibs4c422, kdelibs4c423, kdelibs4c424, kdelibs4c425, kdelibs4c426, kdelibs4c427, kdelibs4c428, kdelibs4c429, kdelibs4c430, kdelibs4c431, kdelibs4c432, kdelibs4c433, kdelibs4c434, kdelibs4c435, kdelibs4c436, kdelibs4c437, kdelibs4c438, kdelibs4c439, kdelibs4c440, kdelibs4c4

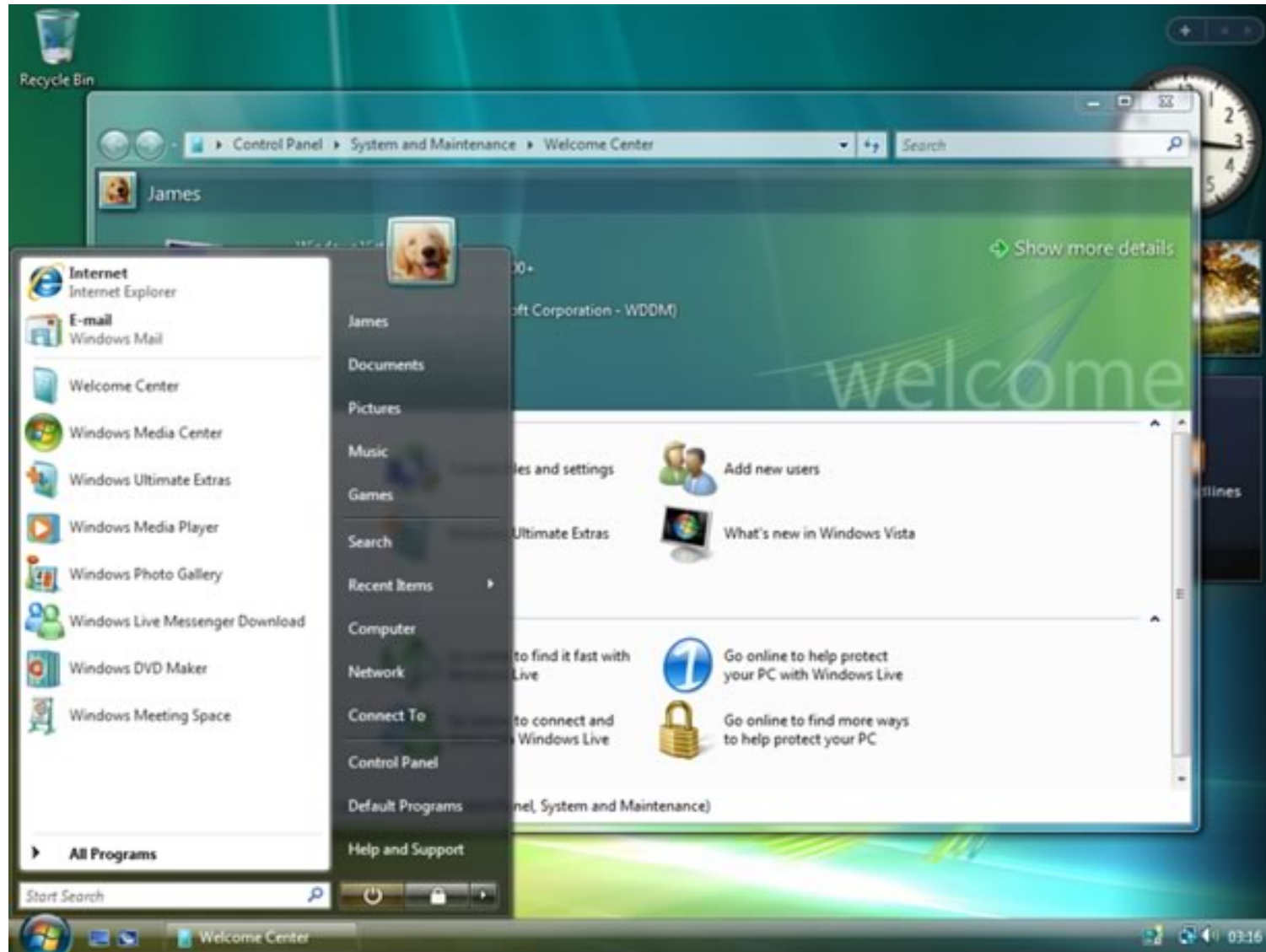
GNOME 1.0 (1999)



Windows XP (released in 2001)



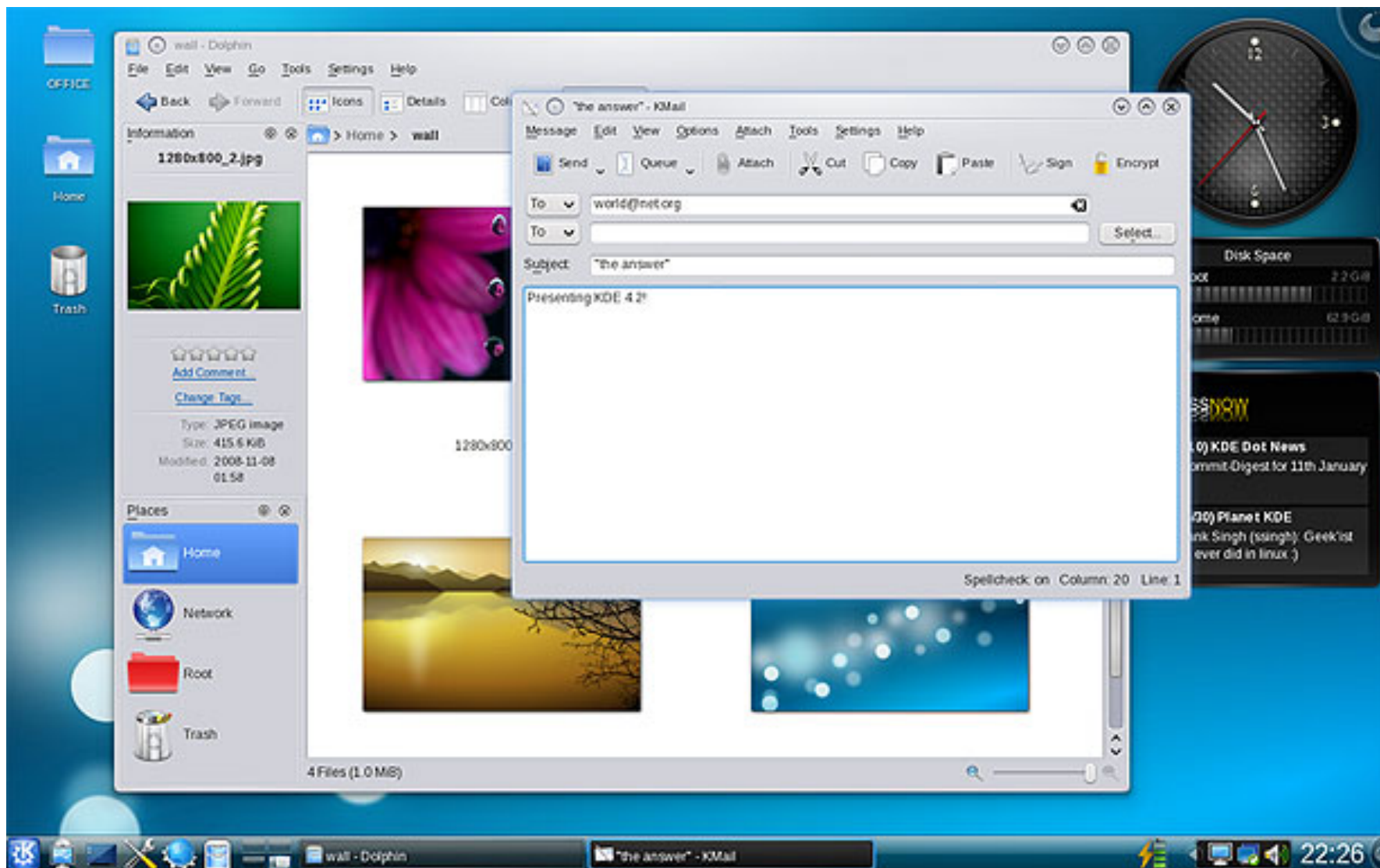
Windows Vista (released in 2007)



Mac OS X Leopard (released in 2007)



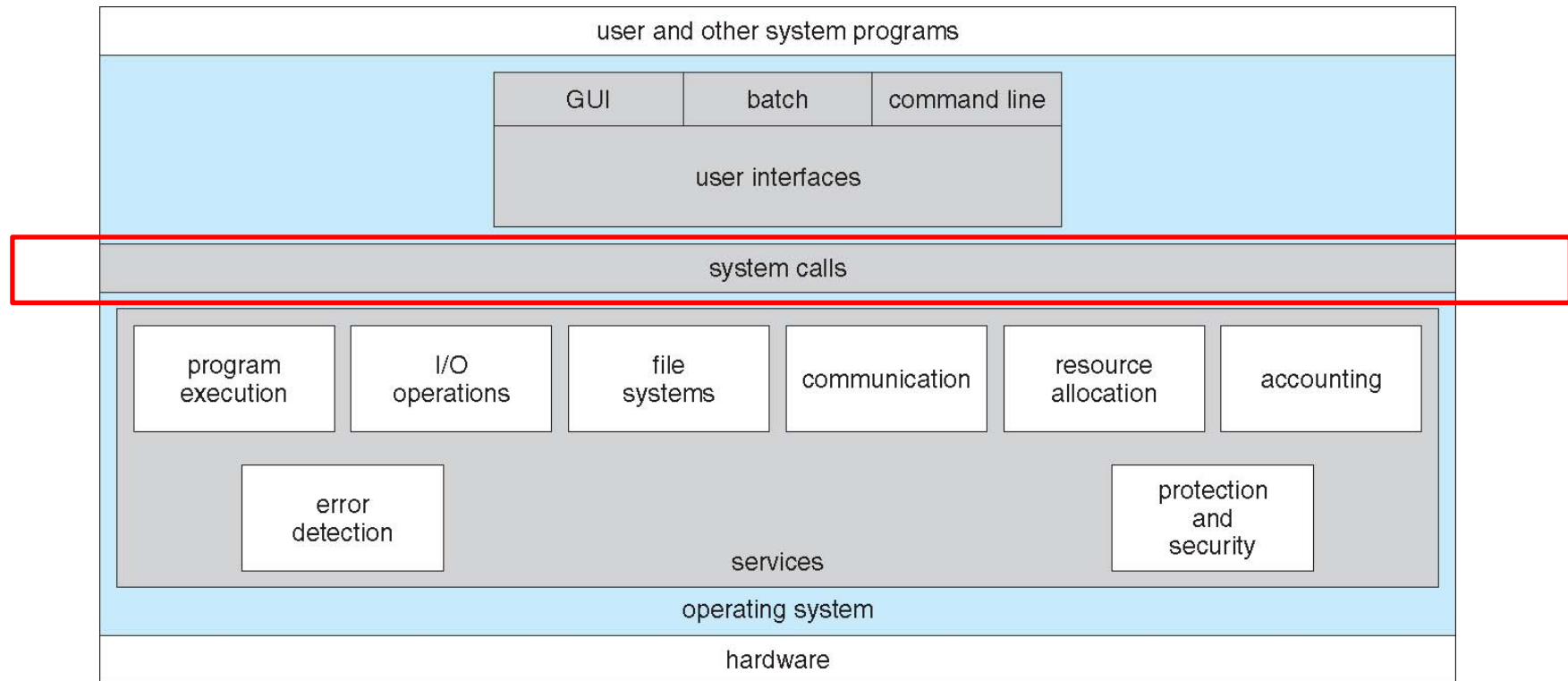
KDE (v4.0 Jan. 2009, v4.2 Mar. 2009)



Windows 10 (July 2015)

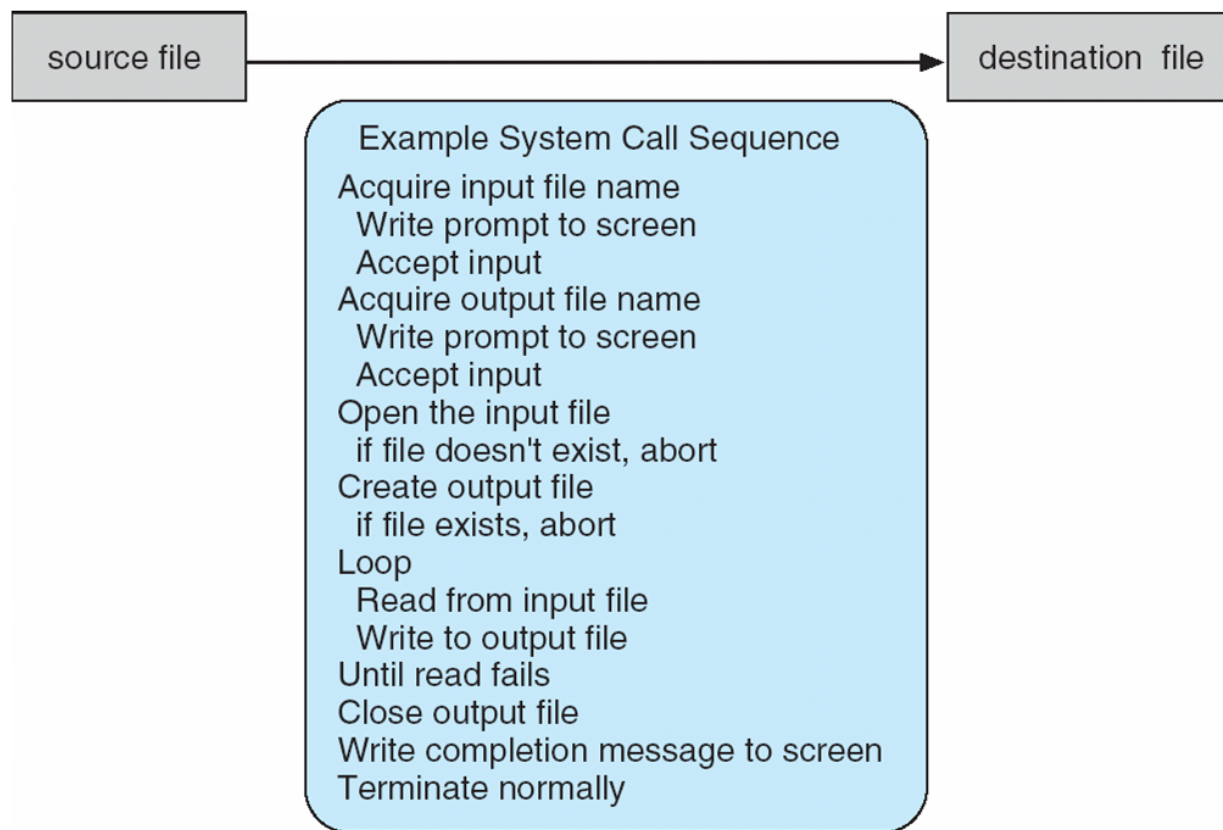


A View of Operating System Services

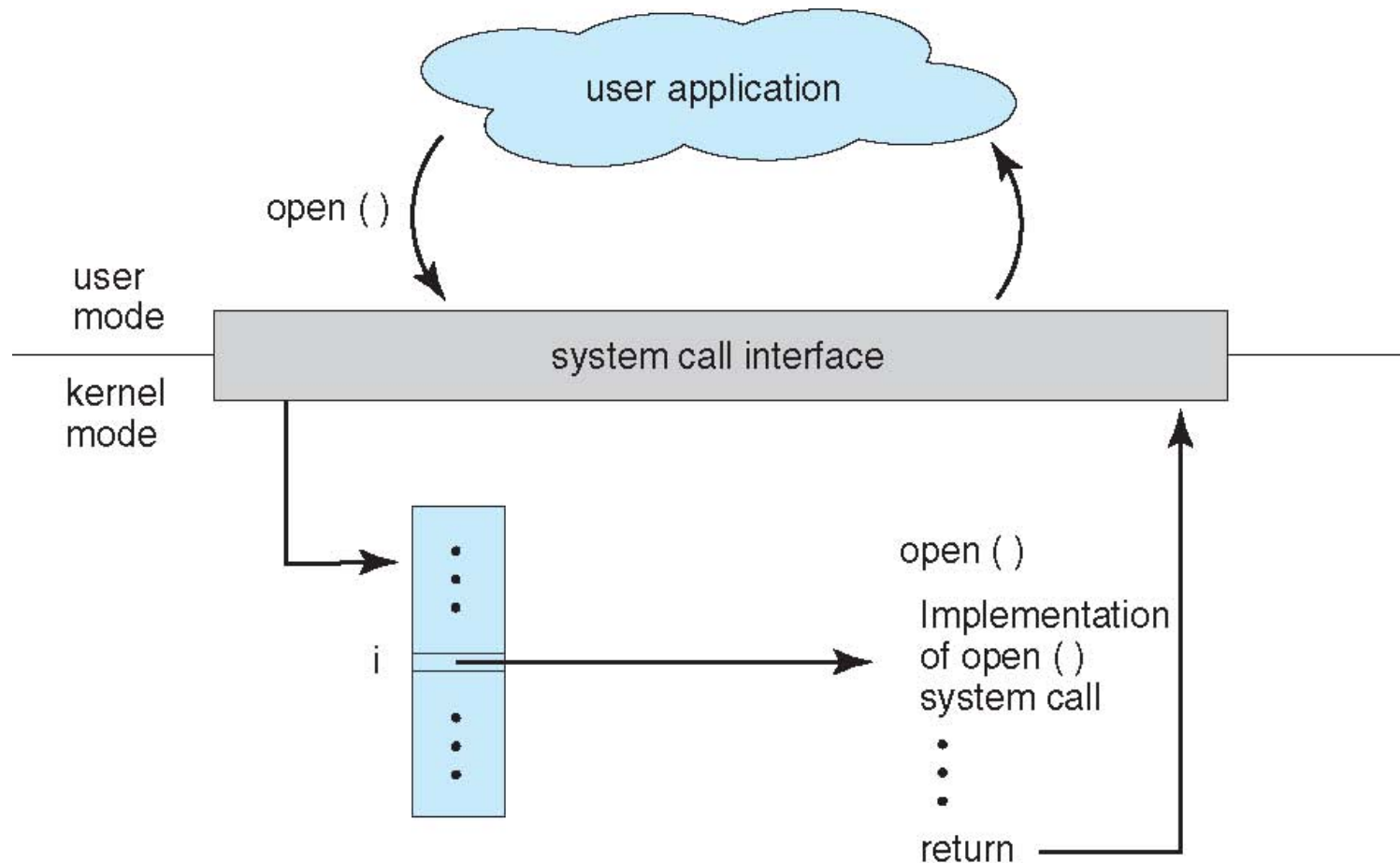


System Call

- Programming interface to the services provided by the OS
- Typically written in a high-level language (C or C++)
- Example: System call sequence to copy the contents of one file to another file



System Call – OS Relationship



Examples of Windows and Unix System Calls

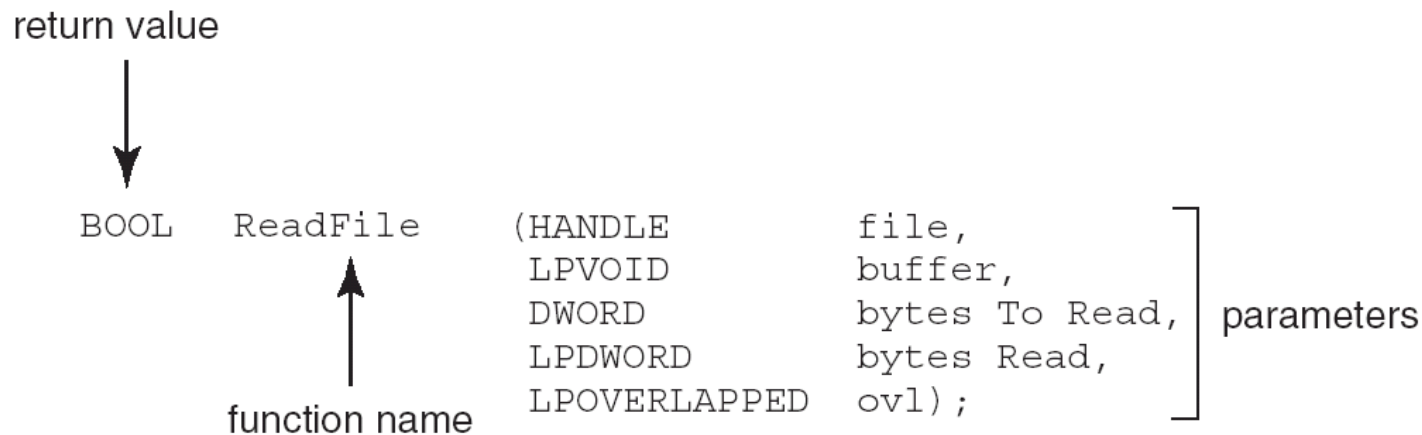
	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

API

- Mostly accessed by programs via a high-level **Application Program Interface (API)** rather than direct system call use
- Three most common APIs
 - Win32 API for Windows
 - POSIX API for POSIX-based systems (UNIX, Linux, and Mac OS X)
 - Java API for the Java virtual machine (JVM)
- Why use APIs rather than system calls?
 - System calls are often more detailed and difficult to work with than the API
 - Program portability

Example of Standard API

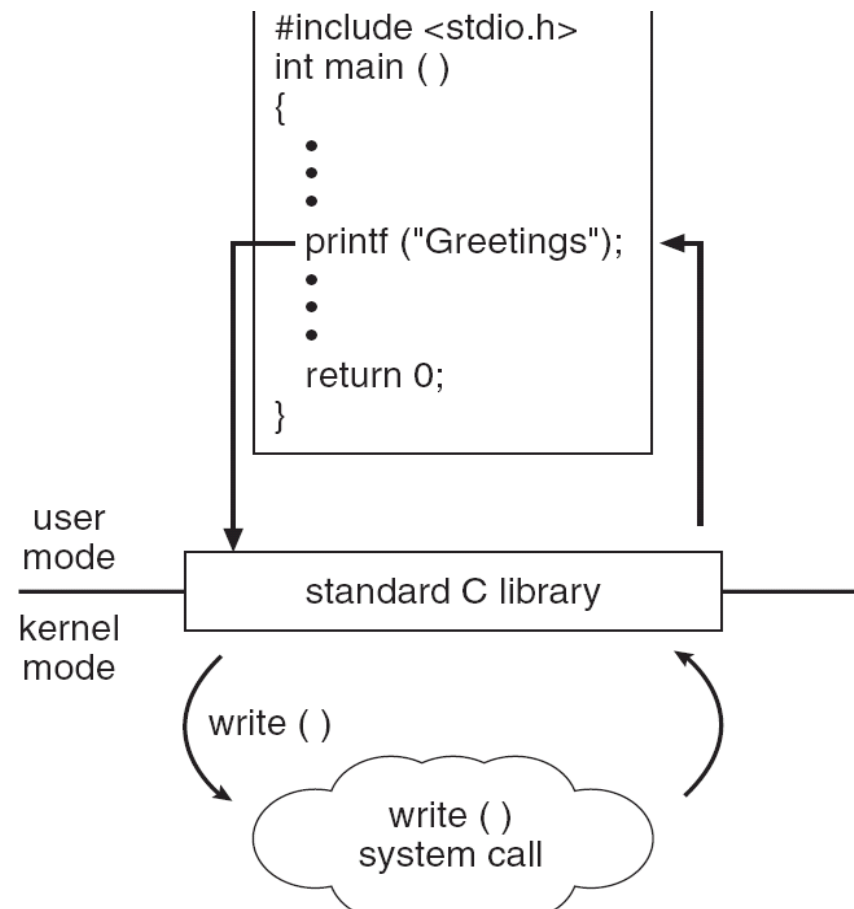
- Consider the ReadFile() function in the
- Win32 API—a function for reading from a file



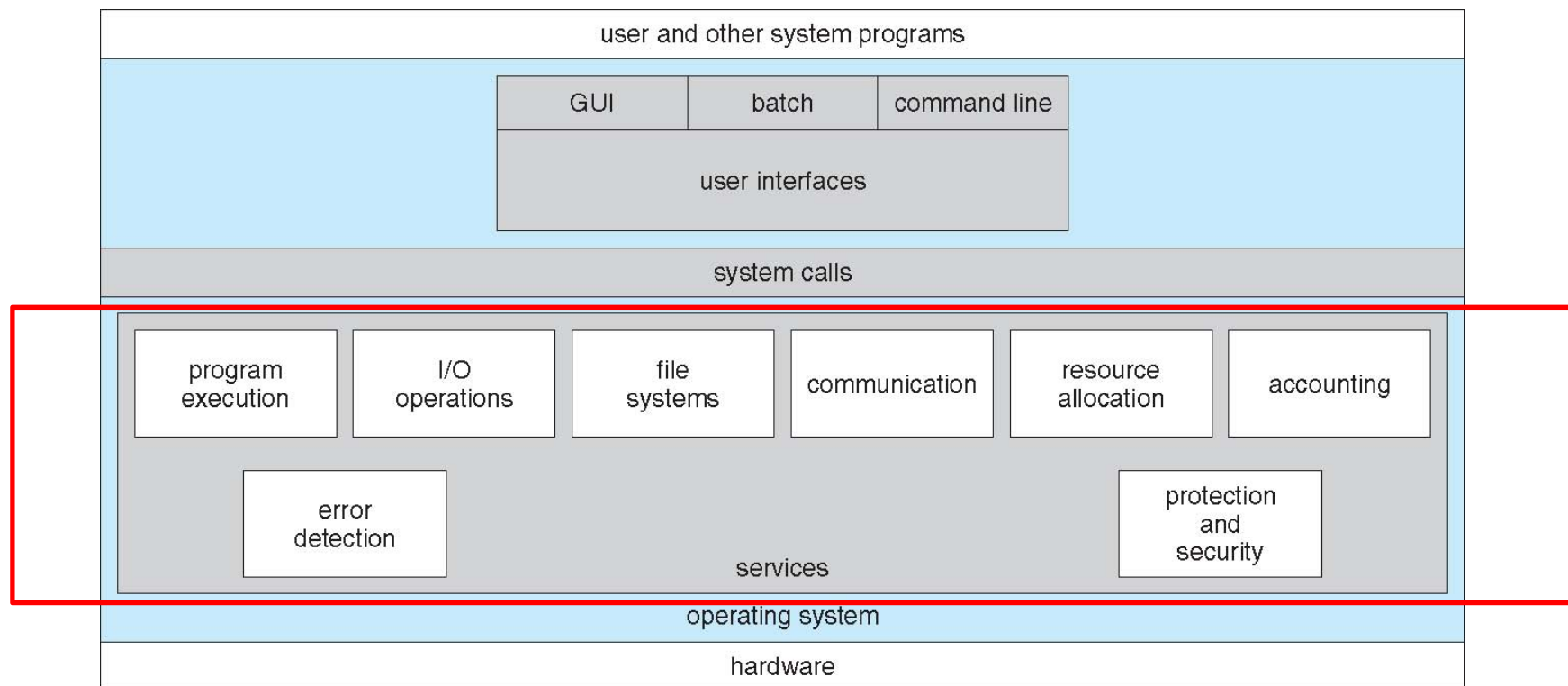
- A description of the parameters passed to ReadFile()
 - HANDLE file—the file to be read
 - LPVOID buffer—a buffer where the data will be read into and written from
 - DWORD bytesToRead—the number of bytes to be read into the buffer
 - LPDWORD bytesRead—the number of bytes read during the last read
 - LPOVERLAPPED ovl—indicates if overlapped I/O is being used

Standard C Library Example

- C program invoking printf() library call, which calls write() system call



A View of Operating System Services



Operating System Services

- Operating-system services:
 - **User interface** - Almost all operating systems have a user interface (UI).
 - ▶ **Graphics User Interface (GUI), Command-Line (CLI), Batch**
 - **Program execution** - The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)
 - **I/O operations** - A running program may require I/O, which may involve a file or an I/O device
 - **File-system manipulation** - Programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.

Operating System Services (Cont.)

- **Communications** – Processes may exchange information, on the same computer or between computers over a network
 - ▶ Communications may be via shared memory or through message passing (packets moved by the OS)
- **Error detection** – OS needs to be constantly aware of possible errors
 - ▶ May occur in the CPU and memory hardware, in I/O devices, in user program
 - ▶ For each type of error, OS should take the appropriate action to ensure correct and consistent computing
 - ▶ Debugging facilities can greatly enhance the user's and programmer's abilities to efficiently use the system

Operating System Services (Cont.)

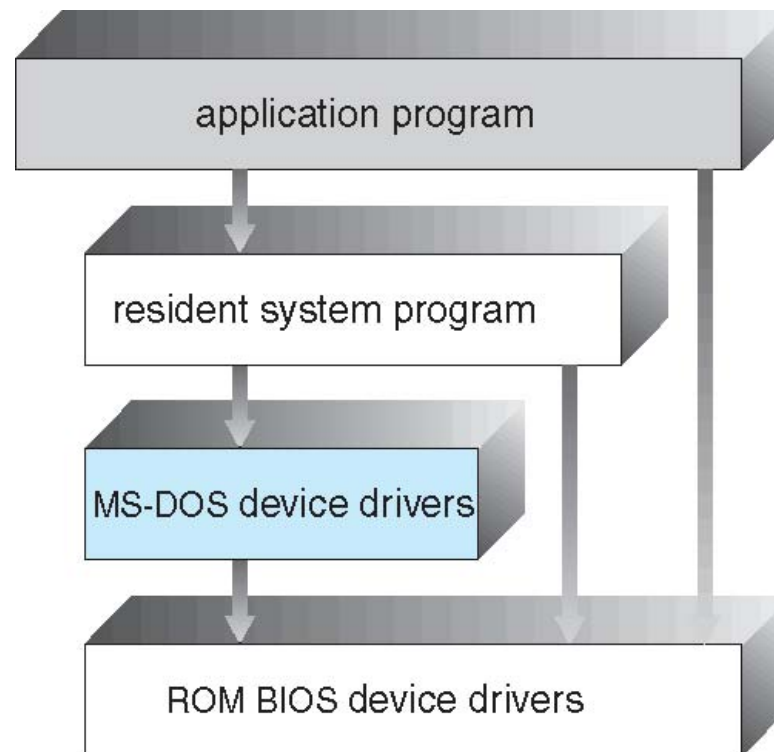
- **Resource allocation** - When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
- **Accounting** - To keep track of which users use how much and what kinds of computer resources
- **Protection and security** - The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other
 - ▶ **Protection** involves ensuring that all access to system resources is controlled
 - ▶ **Security** of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts

Operating-System Structure

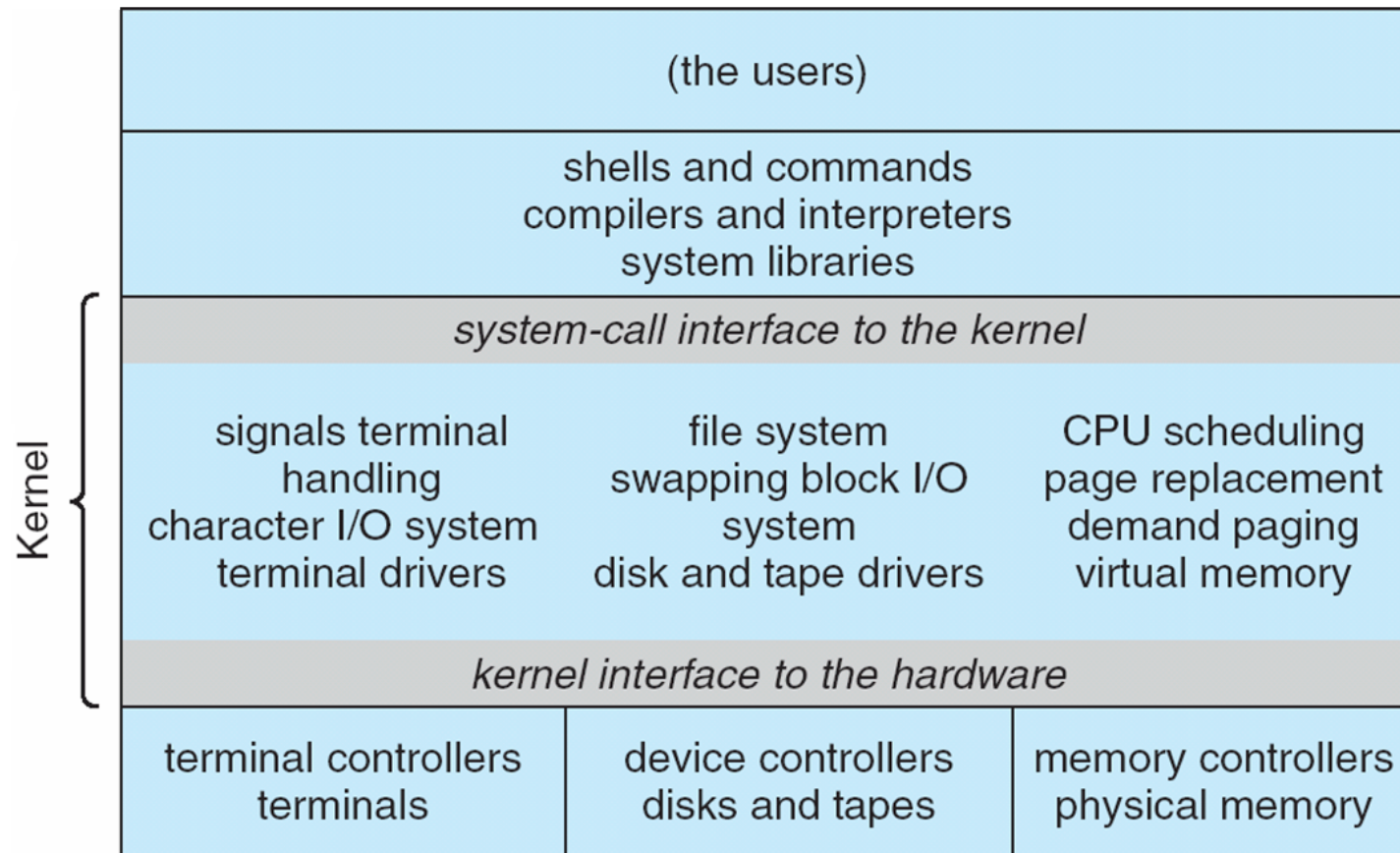
Structure of Components and
Interconnections

Simple Structure

- MS-DOS – written to provide the most functionality in the least space
 - Not divided into modules
 - Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated

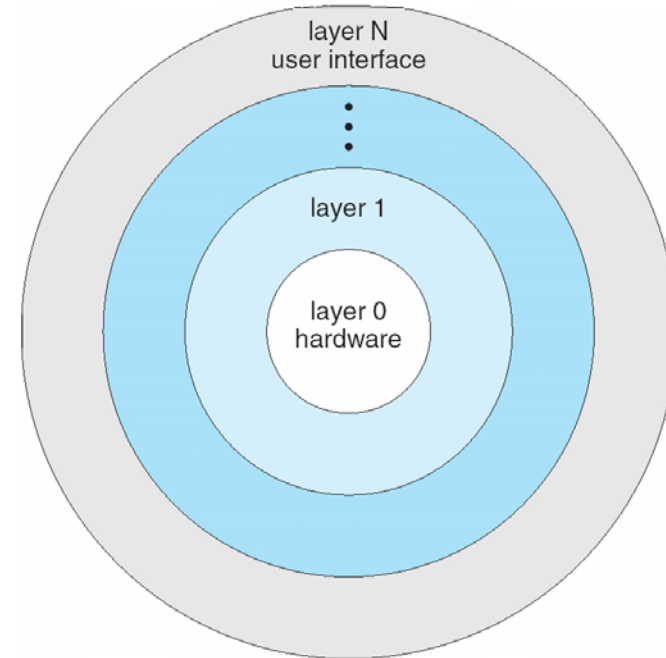


Traditional UNIX System Structure



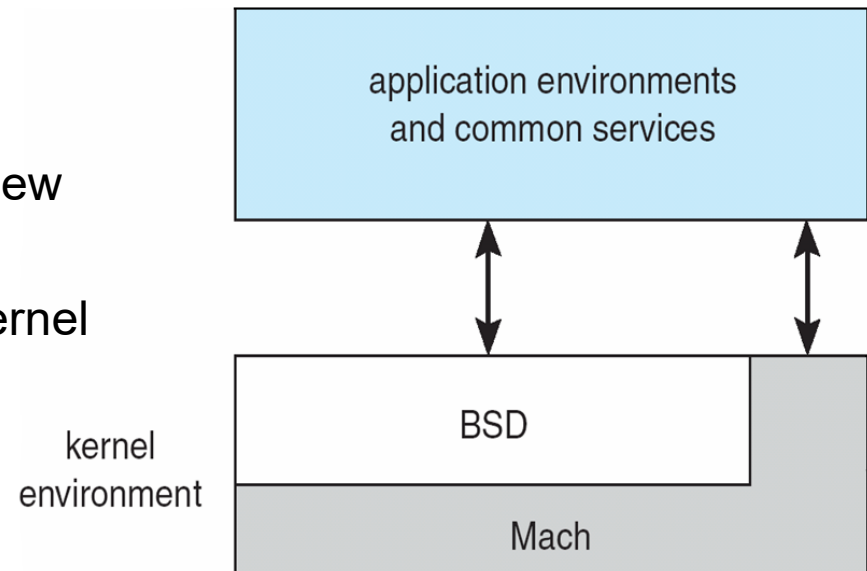
Layered Approach

- The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers
- The main advantage of the layered approach is simplicity of construction and debugging



Microkernel System Structure

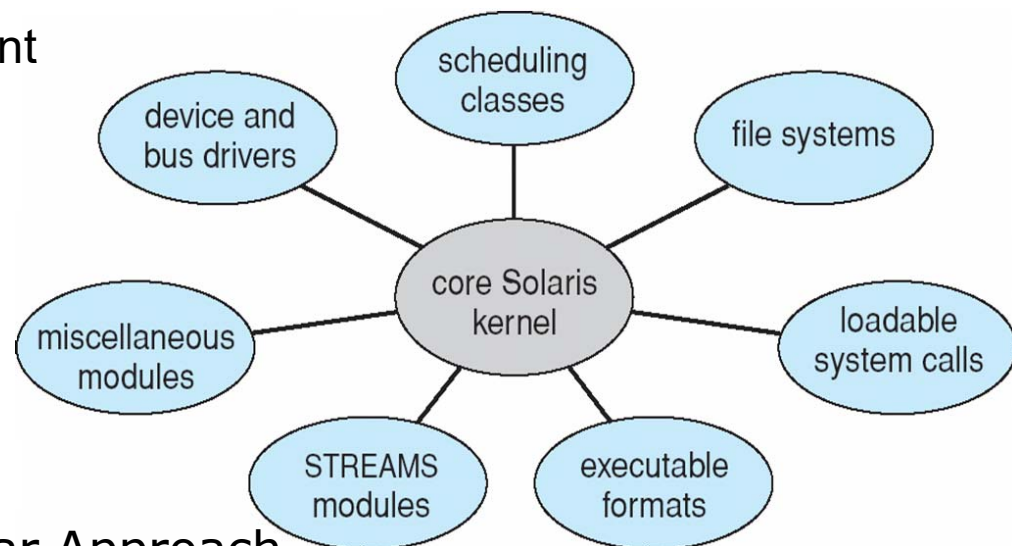
- Moves as much from the kernel into “*user*” space
- Communication takes place between user modules using message passing
- Benefits:
 - Easier to extend a microkernel
 - Easier to port the operating system to new architectures
 - More reliable (less code is running in kernel mode)
 - More secure
- Detriments:
 - Performance overhead of user space to kernel space communication



Mac OS X Structure

Modules

- Most modern operating systems implement kernel modules
 - Uses object-oriented approach
 - Each core component is separated
 - Each talks to the others over known interfaces
 - Each is loadable as needed within the kernel
- Overall, similar to layers but with more flexibility
- Like microkernel but more efficient

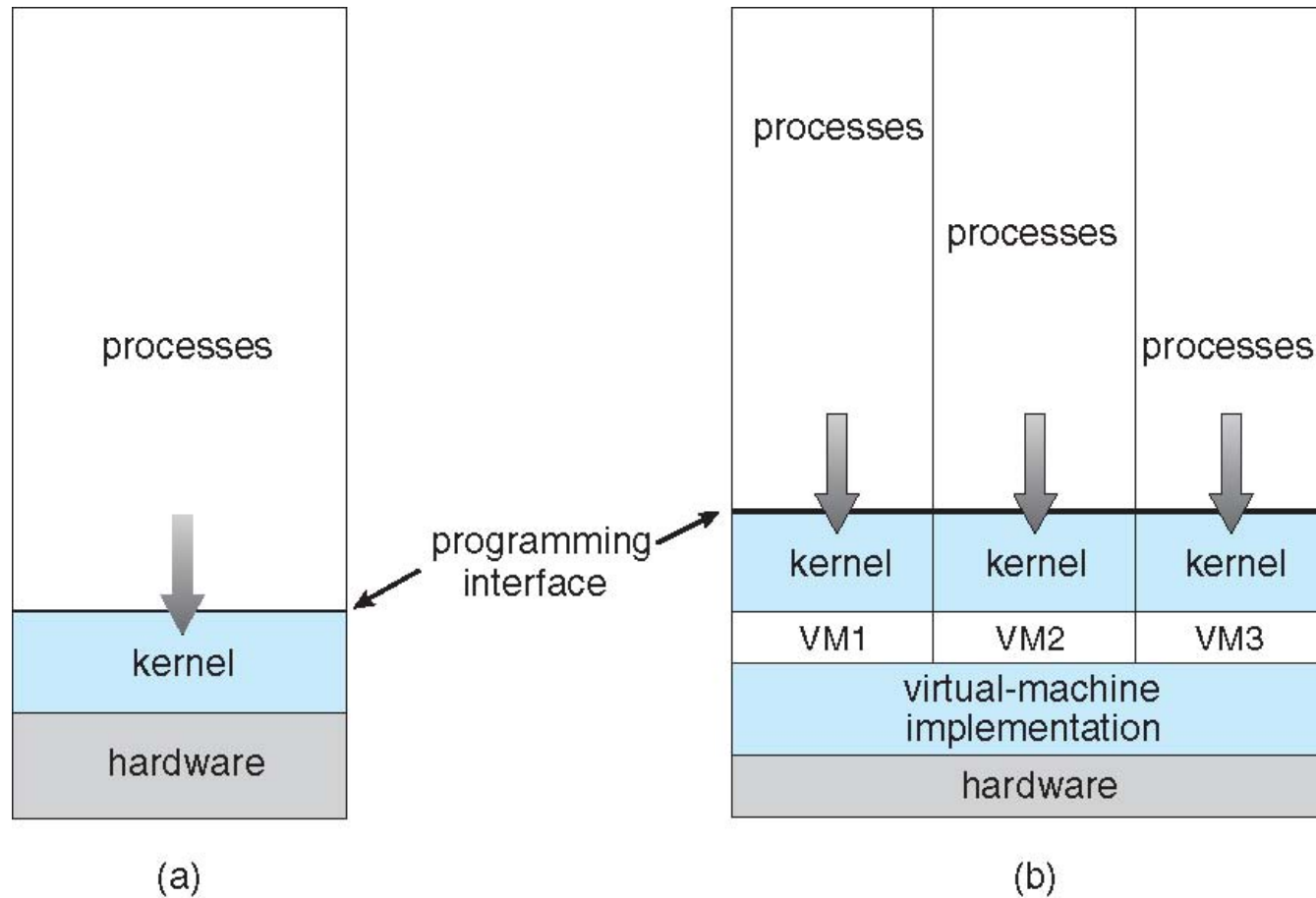


Solaris Modular Approach

Virtual Machines

- A **virtual machine** takes the layered approach to its logical conclusion. It treats hardware and the operating system kernel as though they were all hardware.
- A virtual machine provides an interface *identical* to the underlying bare hardware.
- The **host** creates the illusion that a process has its own processor and (virtual) memory.
- Each **guest** is provided with a (virtual) copy of underlying computer.

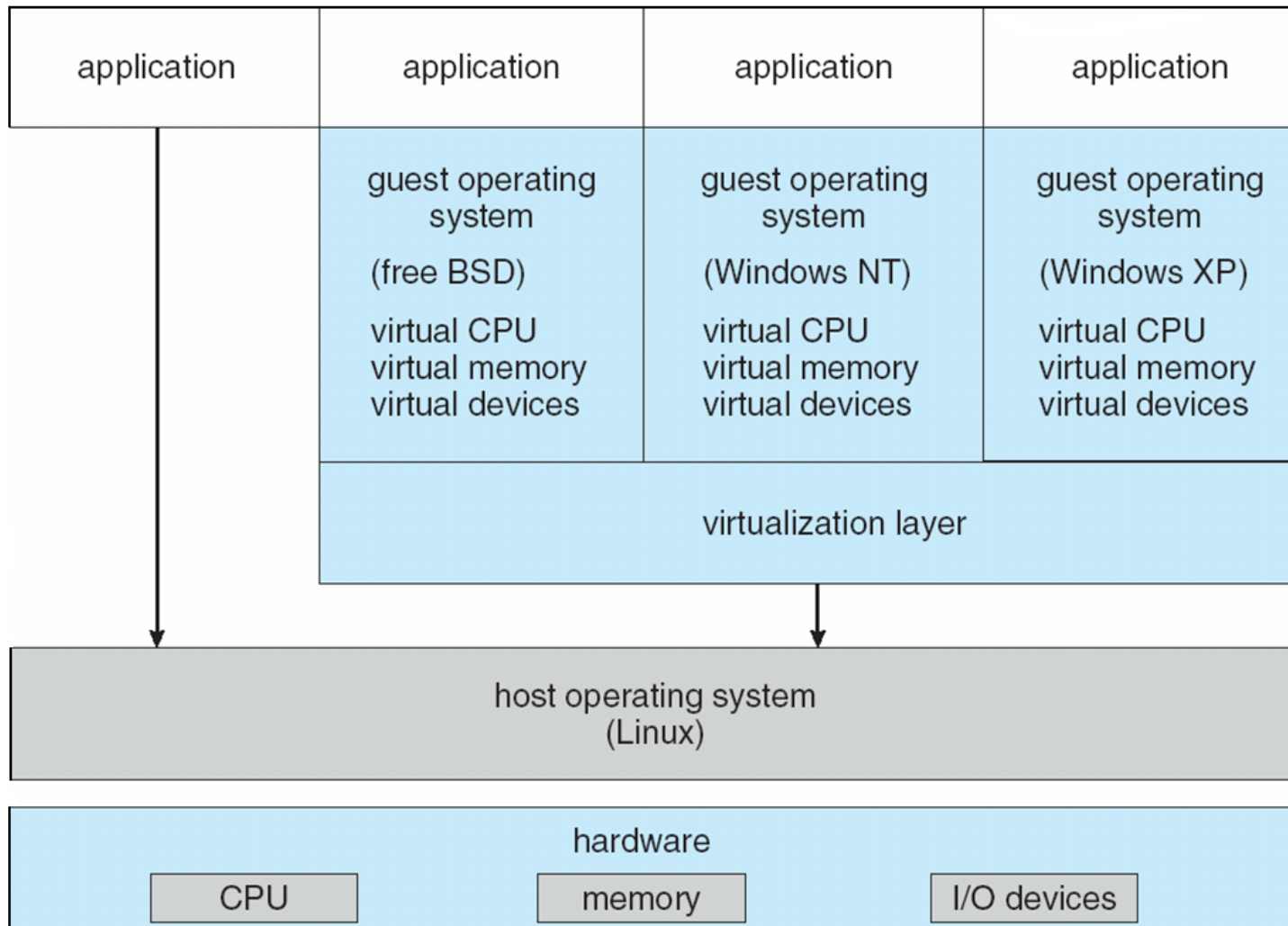
Virtual Machines (Cont.)



(a) Nonvirtual machine

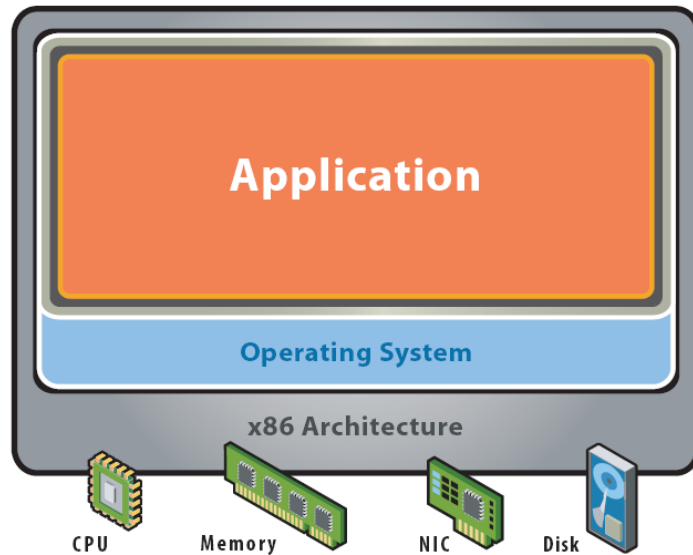
(b) Virtual machine

vmware® Architecture



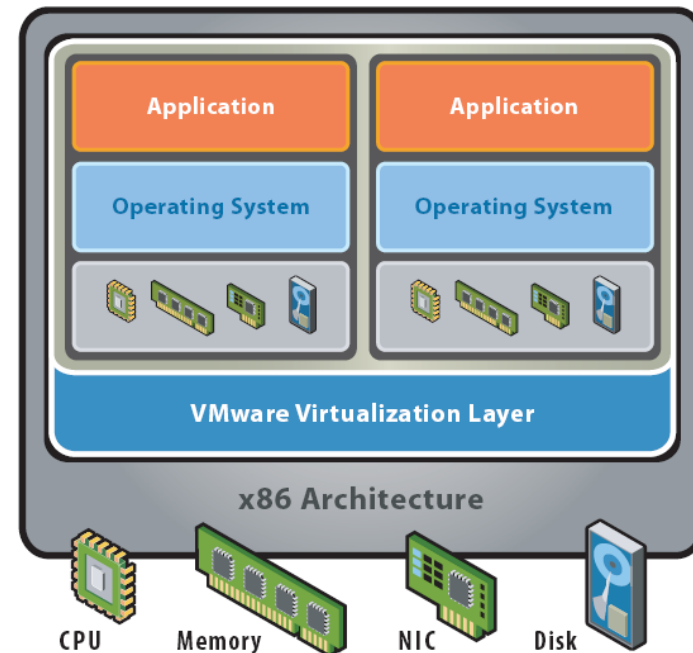
Benefits of Virtualization

Before Virtualization



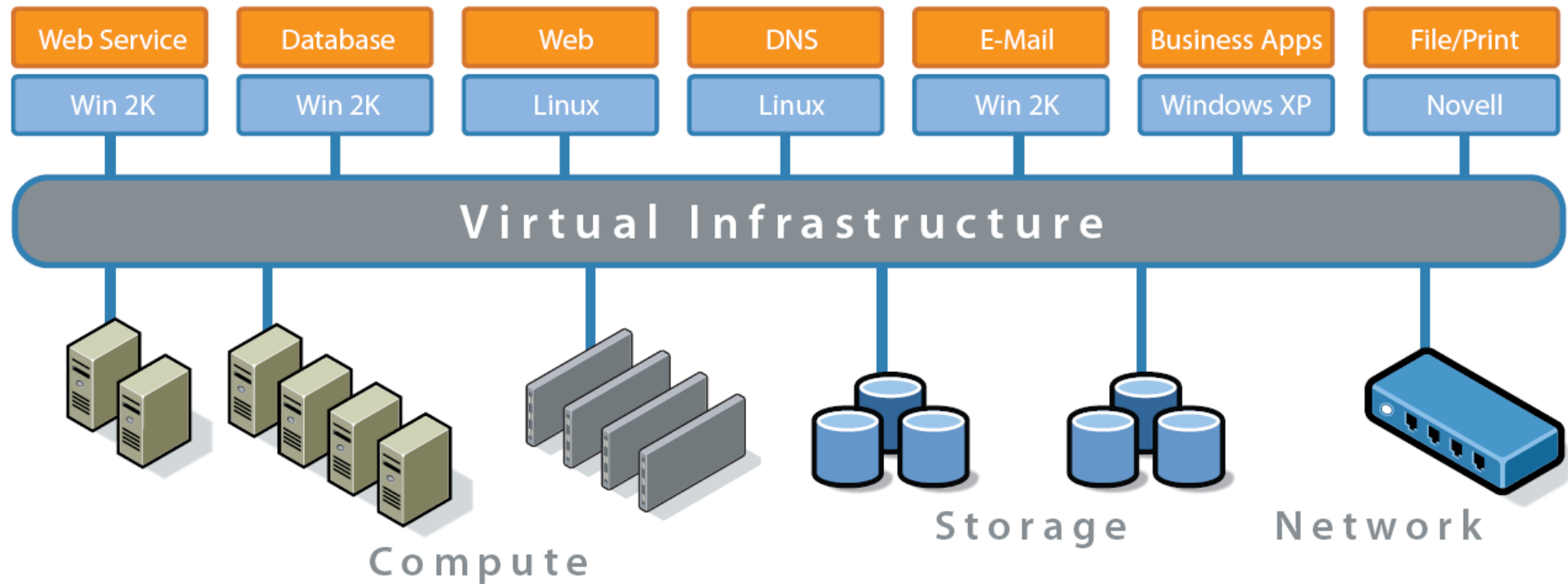
- Single OS image per machine
- Software and hardware tightly coupled
- Underutilized resources
- Inflexible and costly infrastructure

After Virtualization

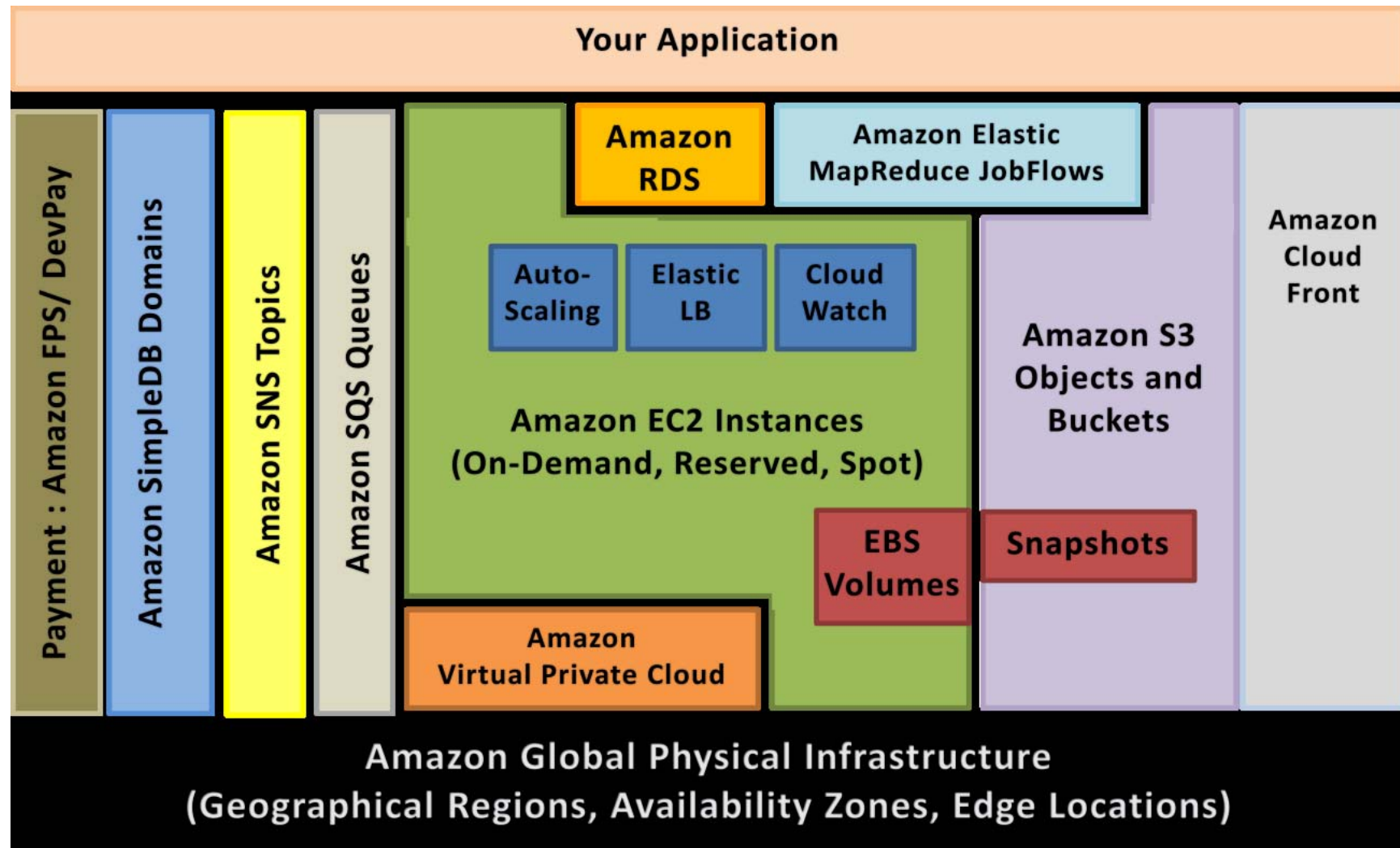


- Multiple OSs on a single machine
- Hardware-independence of operating system and applications
- Better utilization of resources
- Encapsulating OS and application into virtual machines

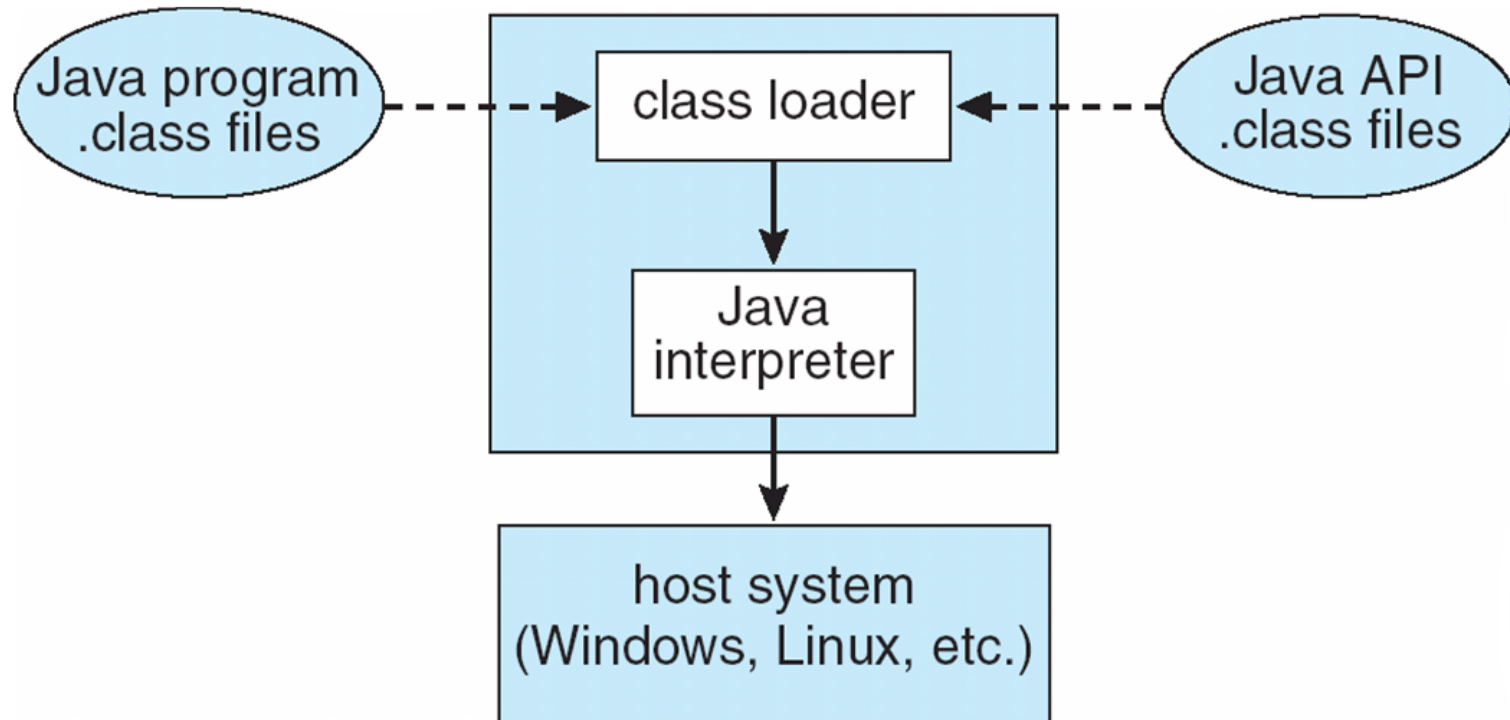
Virtual Infrastructure for Data Center



Amazon Elastic Compute Cloud (EC2)



The Java Virtual Machine



Homework

- Reading
 - Chapter 2: Operating-System Structures