CYENG 312/GECE 594 :
Trusted Operating System (OS)

Lecture 02: Booting and Management Programs

Instructor: Shayan (Sean) Taheri, Ph.D.

Assistant Professor

The Department of Electrical and Cyber Engineering (ECE)

The Institute for Health and Cyber Knowledge (I-HACK)

The Gannon University (GU)

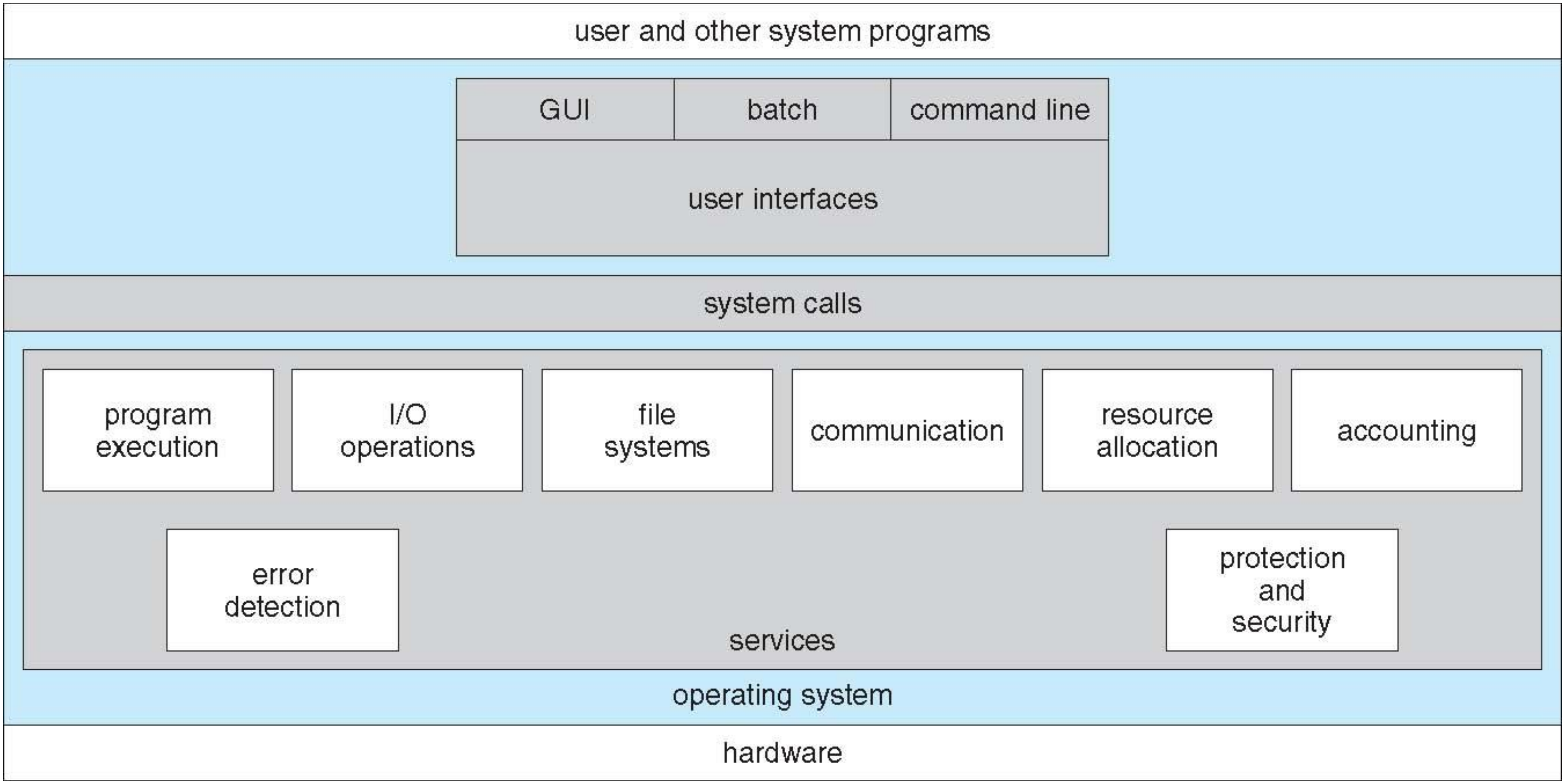
Personal Information

- Name: Shayan (Sean) Taheri.
- Date of Birth: July/28/1991.
- Past Position: Postdoctoral Fellow at University of Florida.
- Ph.D. Degree: Electrical Engineering from the University of Central Florida.
- M.S. Degree: Computer Engineering from the Utah State University.
- University Profile: https://www.gannon.edu/FacultyProfiles.aspx?profile=taheri001

Operating System Services

- One set of operating-system services provides functions that are helpful to the user:
  - *User interface* - *Almost all operating systems have a user interface (UI)*
    - Varies between Command-Line (CLI), Graphics User Interface (GUI), 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* - *The file system is of particular interest. Obviously, programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.*

A View of Operating System Services



user and other system programs

GUI

batch

command line

user interfaces

system calls

program execution

I/O operations

file systems

communication

resource allocation

accounting

error detection

services

protection and security

operating system

hardware

Operating System Services (Contd.)

- One set of operating-system services provides functions that are helpful to the user (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 (Contd.)

- Another set of OS functions exists for ensuring the efficient operation of the system itself via resource sharing
  - ***Resource allocation*** - *When multiple users or multiple jobs running concurrently, resources must be allocated to each of them*
    - Many types of resources - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code
  - ***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
    - If a system is to be protected and secure, precautions must be instituted throughout it. A chain is only as strong as its weakest link.

System Boot

- When power initialized on system, execution starts at a fixed memory location
  - *Firmware ROM used to hold initial boot code*
- Operating system must be made available to hardware so hardware can start it
  - *Small piece of code – **bootstrap loader**, stored in **ROM** or **EEPROM** locates the kernel, loads it into memory, and starts it*
  - *Sometimes two-step process where **boot block** at fixed location loaded by ROM code, which loads bootstrap loader from disk*
- Common bootstrap loader, **GRUB**, allows selection of kernel from multiple disks, versions, kernel options
- Kernel loads and system is then **running**

Starting the Computer

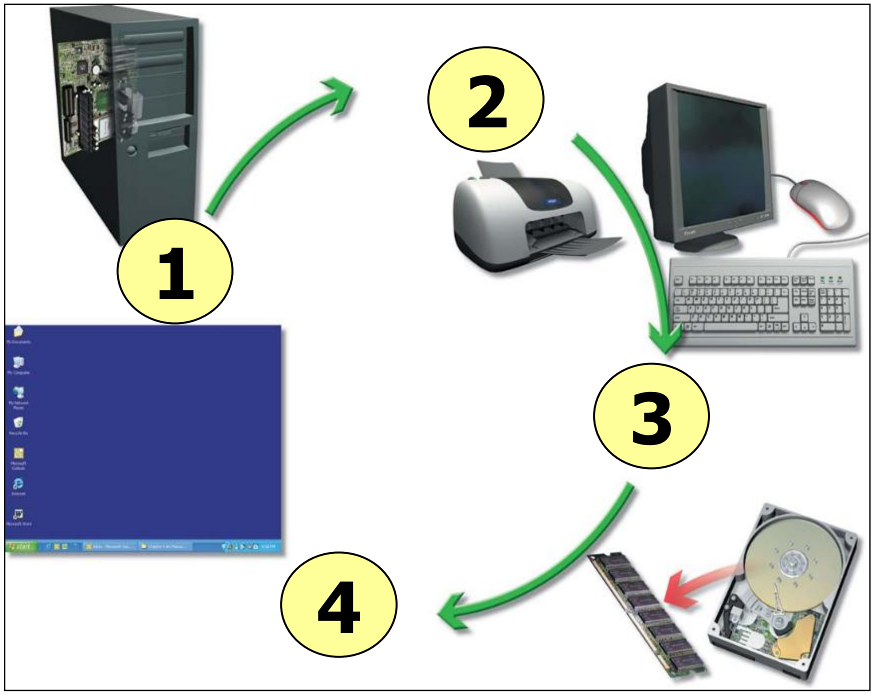
The Boot Process

Step 1: The basic input/output system (BIOS) is activated.

Step 2: A Power-on self-test (POST) checks attached hardware.

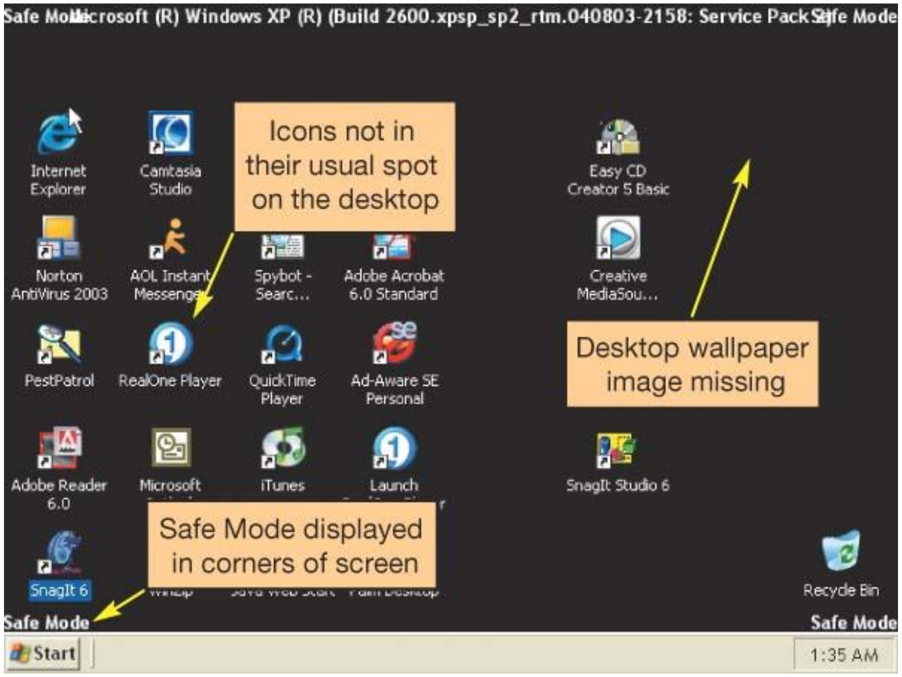
Step 3: The operating system loads into memory.

Step 4: Configuration and customization settings are checked.



Handling Errors in the Boot Process

- Non-system disk or disk error
  - ❑ *Remove the floppy from the drive and press any key*
- POST errors
  - ❑ *Single beep: Everything is loading properly*
  - ❑ *Series of beeps: Hardware problem*
- Safe mode
  - ❑ *Windows does not boot properly*
  - ❑ *Uninstall any new devices or software*



Icons not in their usual spot on the desktop

Desktop wallpaper image missing

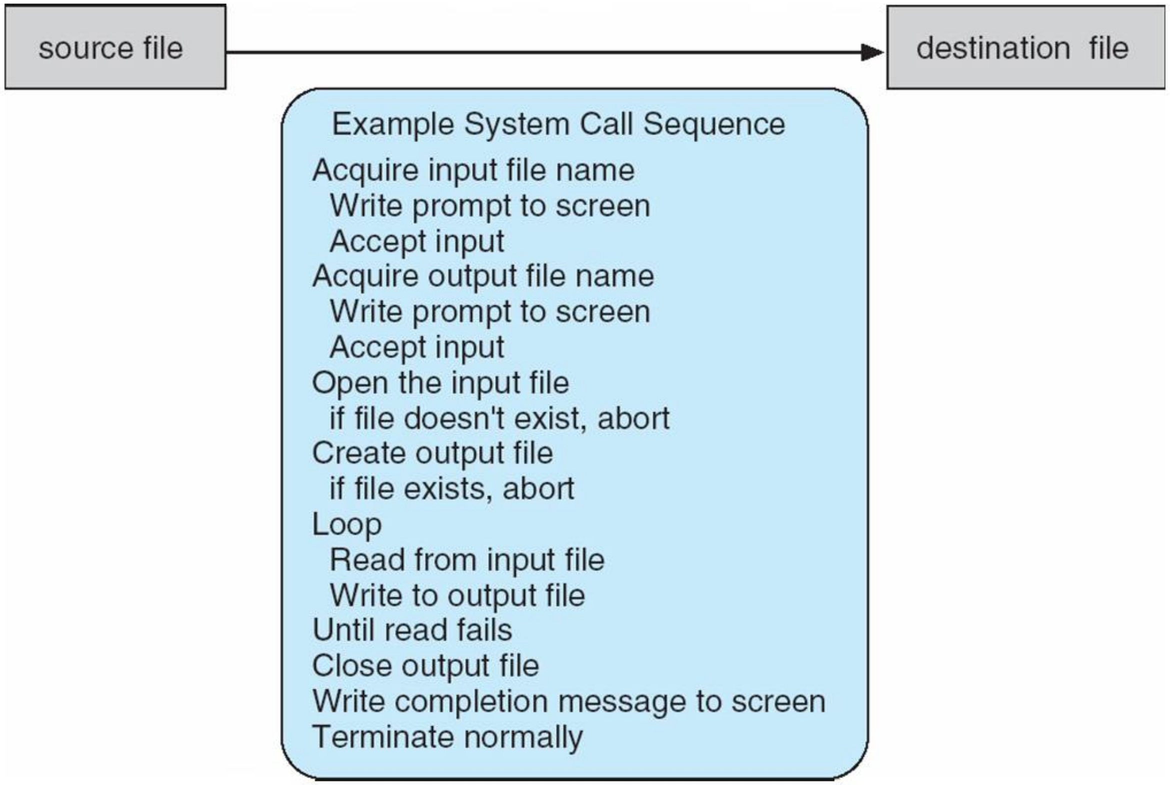
Safe Mode displayed in corners of screen

System Calls

- Programming interface to the services provided by the OS
- Typically written in a high-level language (C or C++)
- Mostly accessed by programs via a high-level Application Program Interface (API) rather than direct system call use
- Three most common APIs are: Win32 API for Windows, POSIX API for POSIX-based systems (including virtually all versions of UNIX, Linux, and Mac OS X), and Java API for the Java virtual machine (JVM)
- Why use APIs rather than system calls?

Example of System Calls

- System call sequence to copy the contents of one file to another file

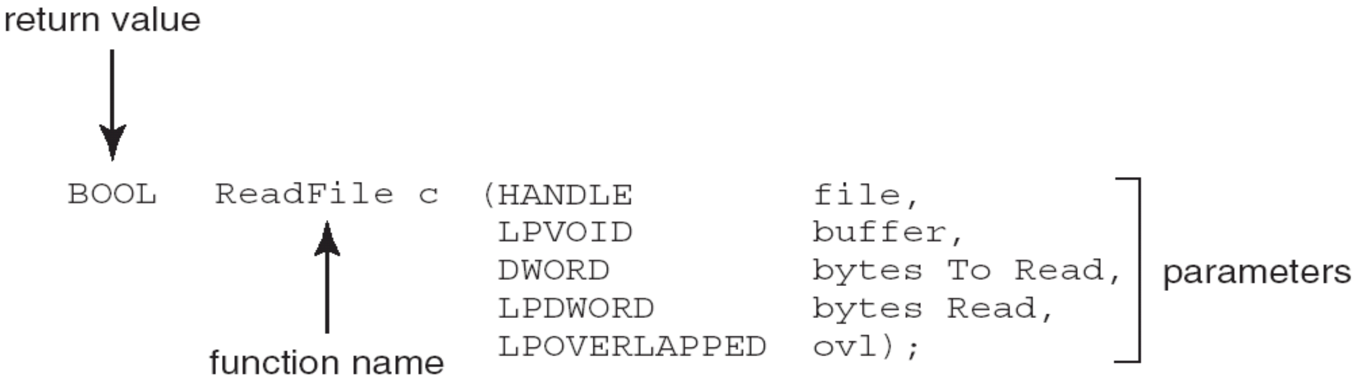


Example System Call Sequence

```
Acquire input file name
  Write prompt to screen
  Accept input
Acquire output file name
  Write prompt to screen
  Accept input
Open the input file
  if file doesn't exist, abort
Create output file
  if file exists, abort
Loop
  Read from input file
  Write to output file
Until read fails
Close output file
Write completion message to screen
Terminate normally
```

Example of Standard API

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



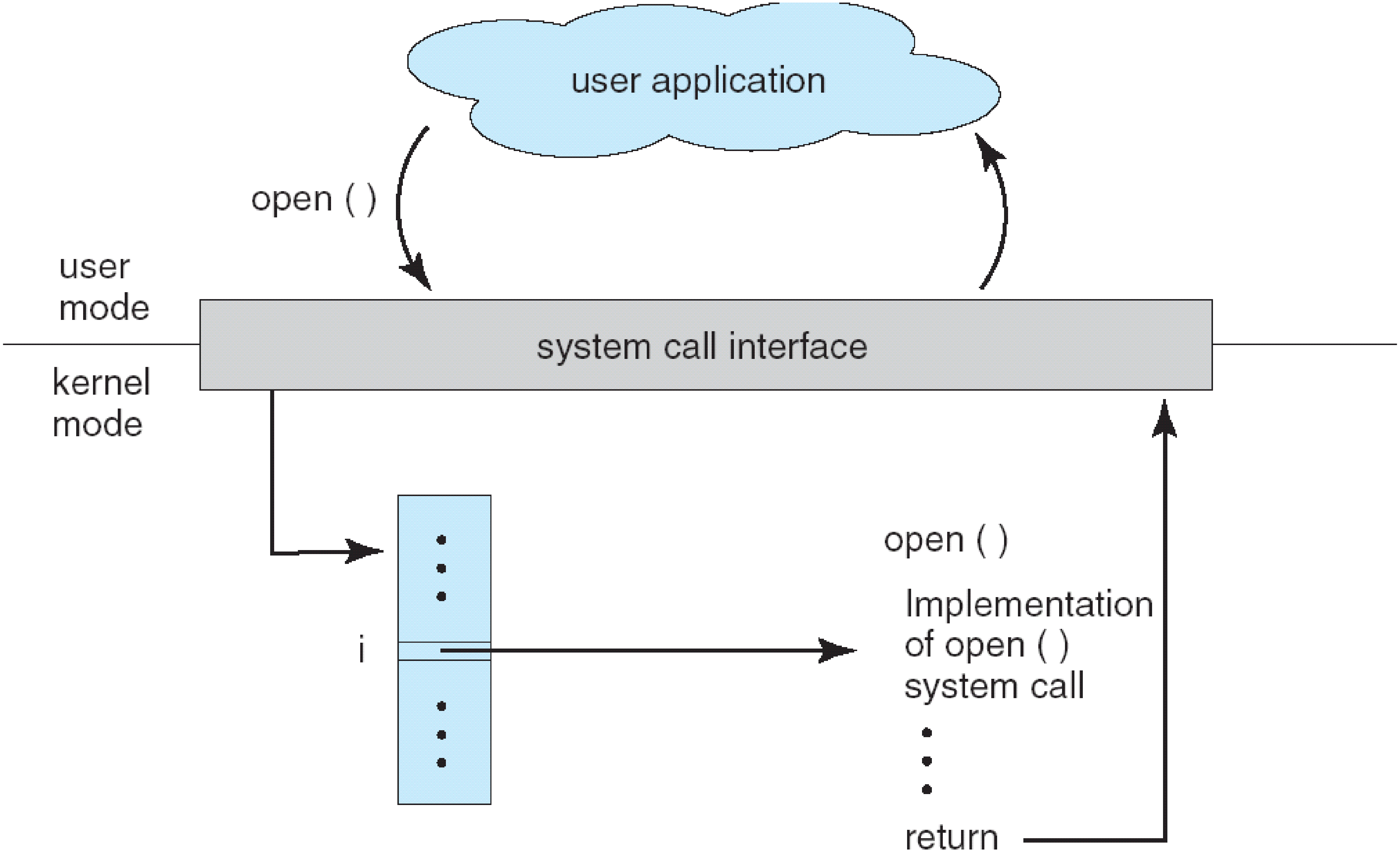
```
BOOL   ReadFile c  (HANDLE          file,
                    LPVOID          buffer,
                    DWORD           bytes To Read,
                    LPDWORD         bytes Read,
                    LPOVERLAPPED    ovl);
```

- 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*

System Call Implementation

- Typically, a number associated with each system call
  - *System-call interface maintains a table indexed according to these numbers*
- The system call interface invokes intended system call in OS kernel and returns status of the system call and any return values
- The caller need know nothing about how the system call is implemented
  - *Just needs to obey API and understand what OS will do as a result call*
  - *Most details of OS interface hidden from programmer by API*
    - Managed by run-time support library (set of functions built into libraries included with compiler)

API – System Call – OS Relationship



Types of System Calls

System calls can be grouped roughly into six major categories:

- 1- Process control
  - ❑ *create process, terminate process*
  - ❑ *end, abort*
  - ❑ *load, execute*
  - ❑ *get process attributes, set process attributes*
  - ❑ *wait for time*
  - ❑ *wait event, signal event*
  - ❑ *allocate and free memory*
  - ❑ *Dump memory if error*
  - ❑ ***Debugger** for determining **bugs, single step** execution*
  - ❑ ***Locks** for managing access to shared data between processes*

Types of System Calls (Contd.)

- 2- File management
  - *create file, delete file*
  - *open, close file*
  - *read, write, reposition*
  - *get and set file attributes*
- 3- Device management
  - *request device, release device*
  - *read, write, reposition*
  - *get device attributes, set device attributes*
  - *logically attach or detach devices*
- 4- Information maintenance
  - *get time or date, set time or date*
  - *get system data, set system data*
  - *get and set process, file, or device attributes*

Types of System Calls (Contd.)

- 5- Communications
  - *create, delete communication connection*
  - *send, receive messages if **message passing model** to **host name** or **process name***
    - From **client** to **server**
  - ***Shared-memory model** create and gain access to memory regions*
  - *transfer status information*
  - *attach and detach remote devices*
- 6- Protection
  - *Control access to resources*
  - *Get and set permissions*
  - *Allow and deny user access*

System Programs

- System programs provide a convenient environment for program development and execution. They can be divided into:
  - *File manipulation and modification*
  - *Status information sometimes stored in a File modification*
  - *Programming language support*
  - *Program loading and execution*
  - *Communications*
  - *Background services*
  - *Application programs*
- Most users′ view of the operation system is defined by system programs, not the actual system calls

System Programs (Contd.)

- Provide a convenient environment for program development and execution
  - *Some of them are simply user interfaces to system calls; others are considerably more complex*

- **File management** - Create, delete, copy, rename, print, dump, list, and generally manipulate files and directories

- **Status information**
  - *Some ask the system for info - date, time, amount of available memory, disk space, number of users*
  - *Others provide detailed performance, logging, and debugging information*
  - *Typically, these programs format and print the output to the terminal or other output devices*
  - *Some systems implement a **registry** - used to store and retrieve configuration information*

System Programs (Contd.)

- **File modification**
  - *Text editors to create and modify files*
  - *Special commands to search contents of files or perform transformations of the text*
- **Programming-language support** - Compilers, assemblers, debuggers and interpreters sometimes provided
- **Program loading and execution**- Absolute loaders, relocatable loaders, linkage editors, and overlay-loaders, debugging systems for higher-level and machine language
- **Communications** - Provide the mechanism for creating virtual connections among processes, users, and computer systems
  - *Allow users to send messages to one another's screens, browse web pages, send electronic-mail messages, log in remotely, transfer files from one machine to another*

System Programs (Contd.)

- **Background Services**
  - *Launch at boot time*
    - Some for system startup, then terminate
    - Some from system boot to shutdown
  - *Provide facilities like disk checking, process scheduling, error logging, printing*
  - *Run in user context not kernel context*
  - *Known as* ***services***, ***subsystems***, ***daemons***

- **Application programs**
  - *Don’t pertain to system*
  - *Run by users*
  - *Not typically considered part of OS*
  - *Launched by command line, mouse click, finger poke*

Process/Task/Application Management

- A process is a program in execution. Program is a *passive entity*, process is an *active entity*.
- Process needs resources to accomplish its task
  - *CPU time*
- Representation of process
  - *Process has one **program counter** specifying location of next instruction to execute*
  - *Data structure (stores information of a process)*
- Many processes may be associated with the same program
- Typically system has many processes
  - *some user processes,*
  - *some operating system processes*
- Life cycle of a process
  - *States*
  - *Arrival, Computation, I/O, I/O completion, termination*

Process/Task/Application Management Activities

The operating system is responsible for the following activities in connection with process management:

- ➢ Creating and deleting both user and system processes
- ➢ Suspending and resuming processes
- ➢ Process scheduling
- ➢ Providing mechanisms for process synchronization
- ➢ Providing mechanisms for process communication
- ➢ Providing mechanisms for deadlock handling

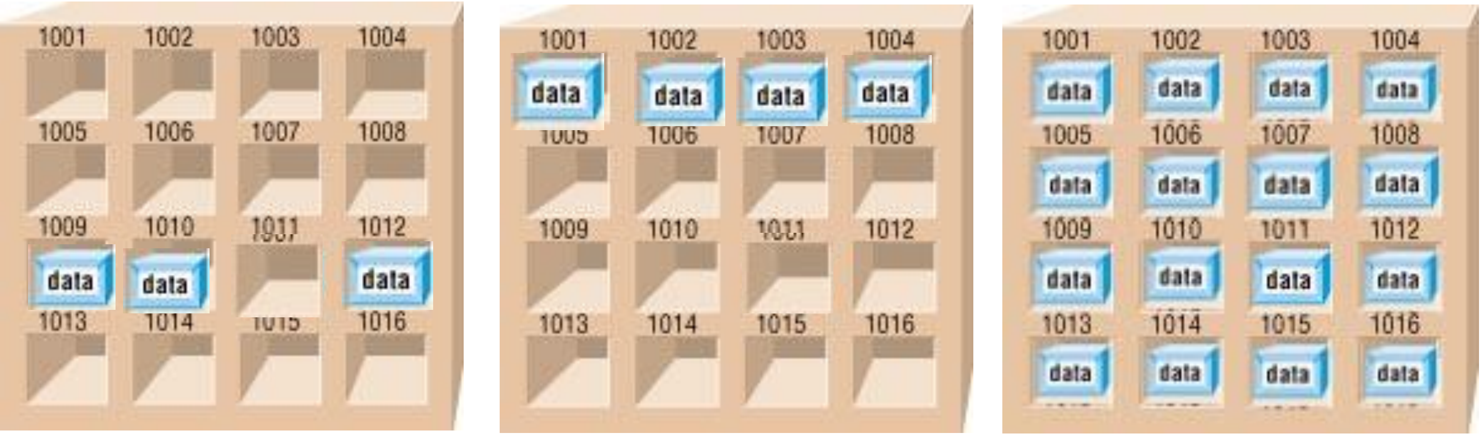
Managing Processor Tasks

- Controls the timing of events the processor works on
  - *Interrupts*
  - *Interrupt handler*
  - *Interrupt table*
  - *Stack*

Managing Memory Tasks

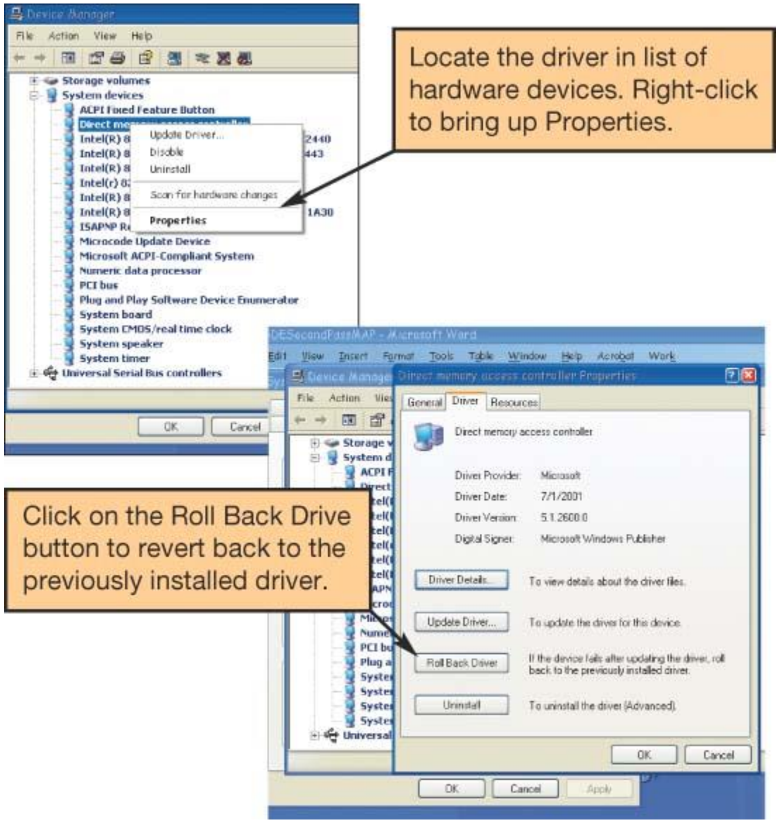
- The operating system allocates space in RAM for instructions and data

RAM



Managing Software Tasks

- Device drivers:
  - *Programs that enable the operating system to communicate with peripheral devices*
  - *Provided by the manufacturer of the device*
- Plug and Play:
  - *Hardware and software standard*
  - *Facilitates the installation of new hardware*



Direct memory access controller

Driver Provider: Microsoft
Driver Date: 7/1/2001
Driver Version: 5.1.2600.0
Digital Signer: Microsoft Windows Publisher

Driver Details... To view details about the driver files.

Update Driver... To update the driver for this device.

Roll Back Driver If the device fails after updating the driver, roll back to the previously installed driver.

Uninstall To uninstall the driver (Advanced).

Locate the driver in list of hardware devices. Right-click to bring up Properties.

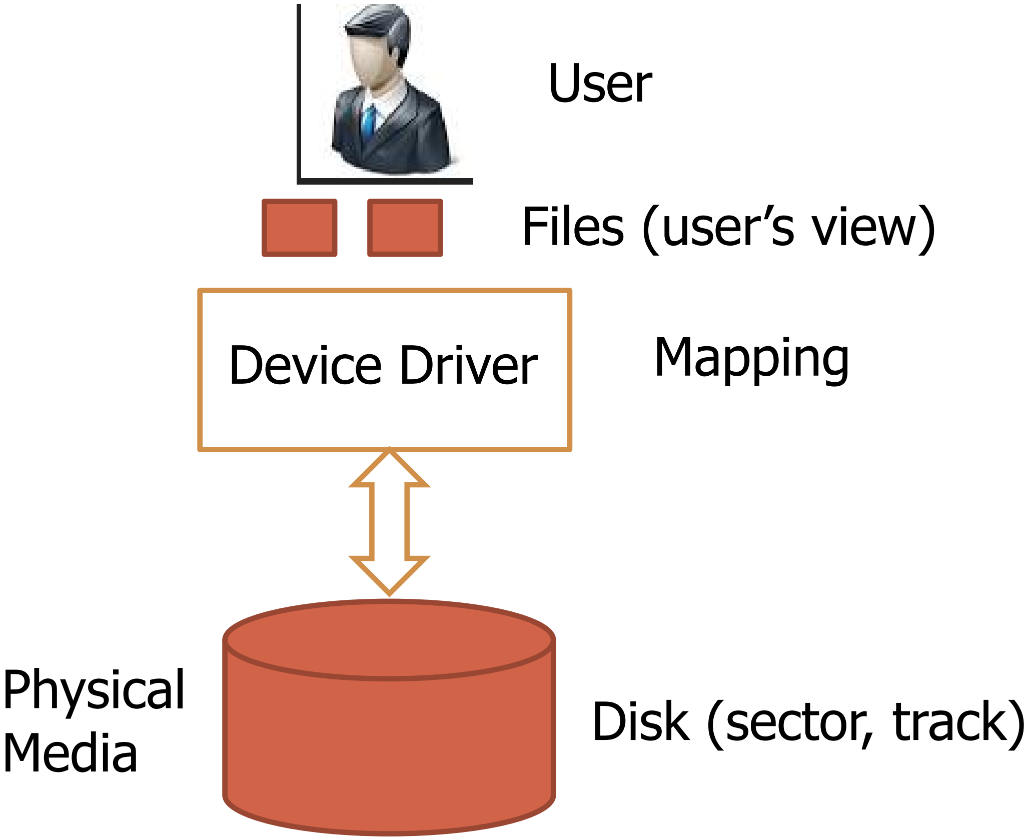
Click on the Roll Back Drive button to revert back to the previously installed driver.

Managing Software Tasks

- Application programming interfaces (APIs):
  - *Blocks of code contained in the operating system*
  - *Coordinates the operating system with software applications*
    - Similar toolbars and menus
  - *Microsoft Direct X*

File Management

- OS provides uniform, logical view of information storage
  - *Abstracts physical properties to logical storage unit - **File***
  - *File => Collection of related information defined by the creator*
  - *Each medium is controlled by device (i.e., disk drive, tape drive)*
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)



File Management

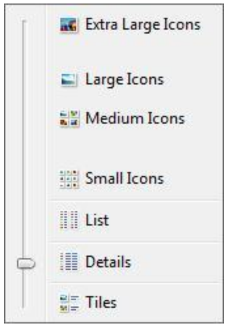
- OS provides uniform, logical view of information storage
  - *Abstracts physical properties to logical storage unit - **file***
  - *Each medium is controlled by device (i.e., disk drive, tape drive)*
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)

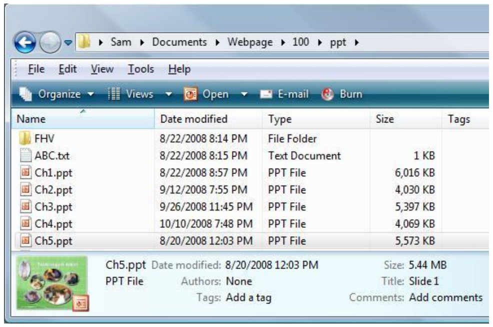
- OS implements the abstract concept of file by managing mass storage media (disk etc) and devices that control them

- Files usually organized into directories
- Access control on most systems to determine who can access what
- File-System management
  - *Creating and deleting files and directories*
  - *Primitives to manipulate files and dirs*
  - *Mapping files onto secondary storage*

File Management

- The operating system provides an organizational structure to the computer’s contents
- Hierarchical structure of directories:
  - *Drives*
    - Folders
      - Subfolders
        - Files




Sam ▸ Documents ▸ Webpage ▸ 100 ▸ ppt ▸
File Edit View Tools Help
Organize Views Open E-mail Burn

Name	Date modified	Type	Size	Tags
FHV	8/22/2008 8:14 PM	File Folder		
ABC.txt	8/22/2008 8:15 PM	Text Document	1 KB	
Ch1.ppt	8/22/2008 8:57 PM	PPT File	6,016 KB	
Ch2.ppt	9/12/2008 7:55 PM	PPT File	4,030 KB	
Ch3.ppt	9/26/2008 11:45 PM	PPT File	5,397 KB	
Ch4.ppt	10/10/2008 7:48 PM	PPT File	4,069 KB	
Ch5.ppt	8/20/2008 12:03 PM	PPT File	5,573 KB	

Ch5.ppt Date modified: 8/20/2008 12:03 PM Size: 5.44 MB
PPT File Authors: None Title: Slide 1
Tags: Add a tag Comments: Add comments

- Viewing and Sorting Files and Folders: Windows Explorer and Views
- Naming Files
  - *Name assigned plus filename extension*
  - *only characters not legal in filenames are:*

    \ / : * ? " < > ¦

  - *all others are allowed*
- File Path: Location of the file → Drive, Primary Folder, Subfolders, and File Name.

C:\My Documents\Tech in Action\TIA Pics\dotmatrix.gif

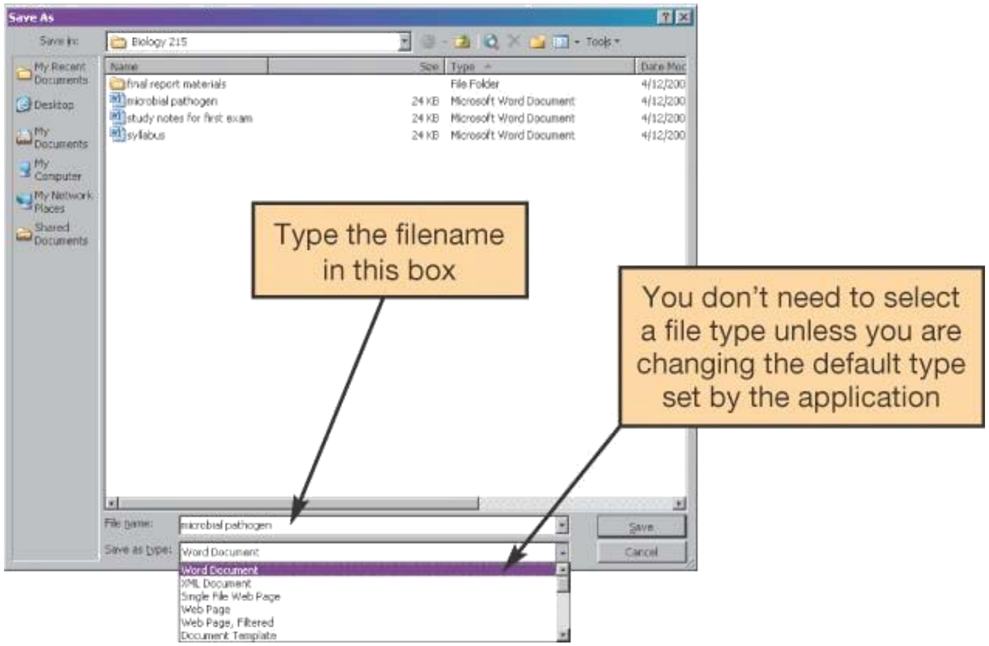
Filename Extensions

- Filename extensions:
  - *Used by programs*

Extension	Type of Document	Application
.doc	Word processing document	Microsoft Word; Corel WordPerfect
.xls	Workbook	Microsoft Excel
.ppt	PowerPoint presentation	Microsoft PowerPoint
.mdb	Database	Microsoft Access
.bmp	Bitmap image	Windows
.zip	Compressed file	WinZip
.pdf	Portable Document Format	Adobe Acrobat
.htm or .html	Web page	Hypertext Markup Language

Working with Files

- ➢ File management actions:
  - ❑ *Open*
  - ❑ *Copy*
  - ❑ *Move*
  - ❑ *Rename*
  - ❑ *Delete*
- ➢ Recycle bin



Save in: Biology 215

Name	Size	Type	Date Mod
final report materials		File Folder	4/12/200
microbial pathogen	24 KB	Microsoft Word Document	4/12/200
study notes for first exam	24 KB	Microsoft Word Document	4/12/200
syllabus	24 KB	Microsoft Word Document	4/12/200

File name: microbial pathogen

Save as type: Word Document

Word Document
XML Document
Single File Web Page
Web Page
Web Page, Filtered
Document Template

Type the filename in this box

You don't need to select a file type unless you are changing the default type set by the application

Saving files

File Commands

- Caution: the above commands do not prompt for confirmation
  - *easy to overwrite/delete a file;   this setting can be overridden (how?)*

- *Exercise* : Given several albums of `.mp3` files all in one folder, move them into separate folders by artist.
- *Exercise* : Modify `a .java file` based on a certain date and time.

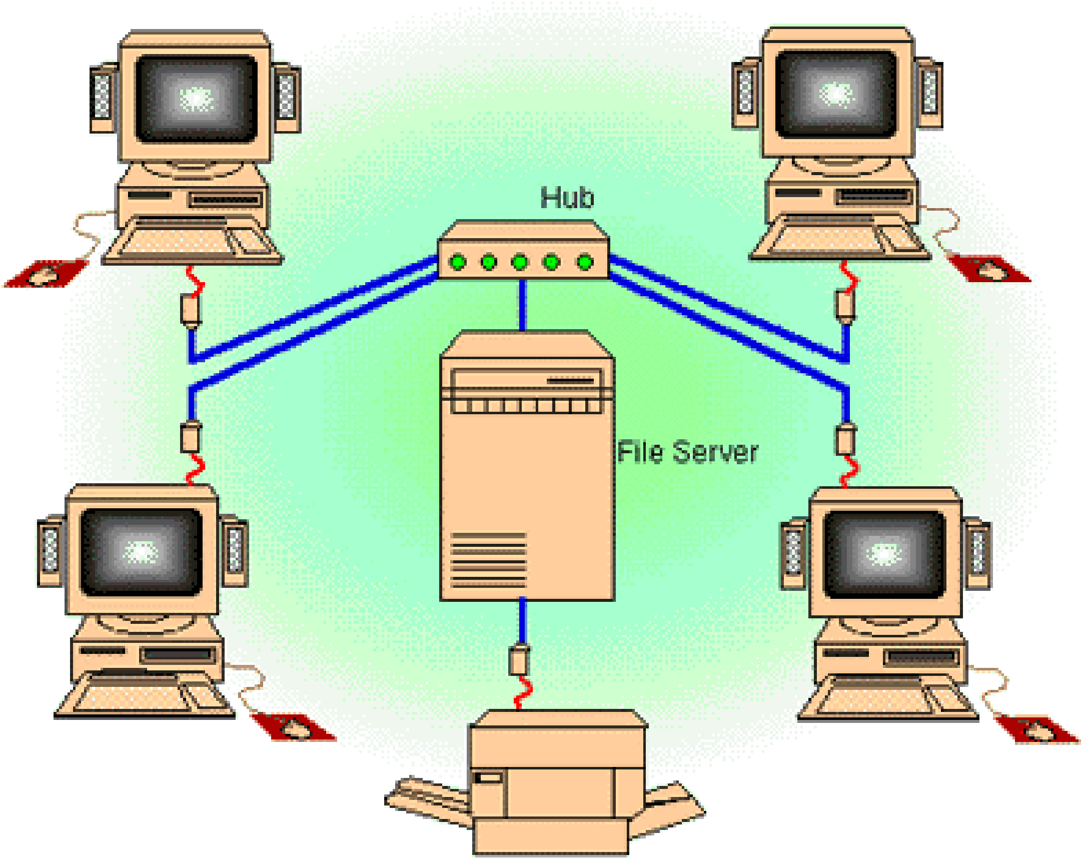
command	description
`cp`	copy a file
`mv`	move or rename a file
`rm`	delete a file
`touch`	create a new empty file, or update its last-modified time stamp

Disk Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
  - *Most of the programs are stored on disk*
- Proper management is of central importance
- Entire speed of computer operation depends on disk subsystem and its algorithms
- OS activities
  - *Storage allocation (logical blocks)*
  - *Free-space management*
  - *Disk scheduling*

Multiuser Operating Systems

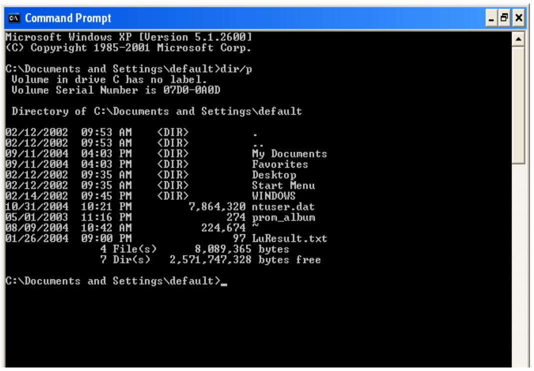
- Known as network operating systems
- Allow access to the computer system by more than one user
- Manage user requests
- Systems include:
  - *UNIX*
  - *Linux*
  - *Novell Netware*
  - *Windows Server 2003*



The User Interface

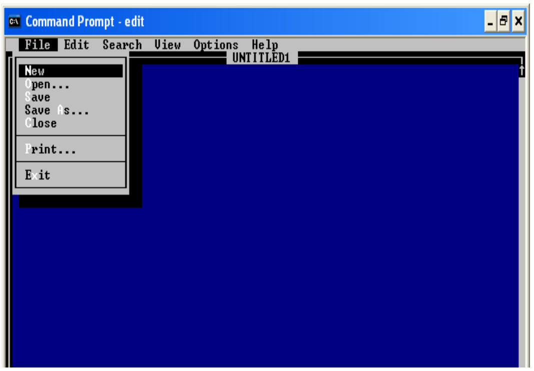
- Enables you to interact with the computer
- Types of interfaces:
  - ❑ *Command-driven interface*
  - ❑ *Menu-driven interface*
  - ❑ *Graphical user interface*

Command-driven

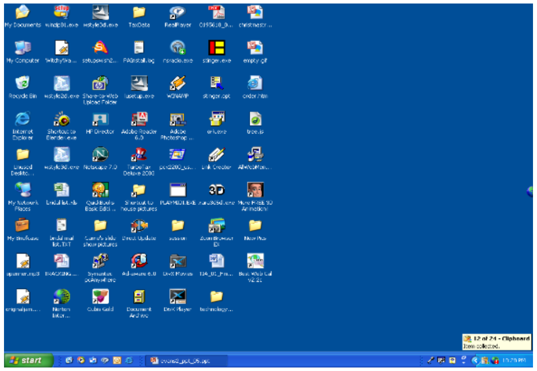


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\default>dir/p
 Volume in drive C has no label.
 Volume Serial Number is 07D0-0A0D

 Directory of C:\Documents and Settings\default

02/12/2002  09:53 AM    <DIR>          .
02/12/2002  09:53 AM    <DIR>          ..
09/11/2004  04:03 PM    <DIR>          My Documents
09/11/2004  04:03 PM    <DIR>          Favorites
02/12/2002  09:35 AM    <DIR>          Desktop
02/12/2002  09:35 AM    <DIR>          Start Menu
02/14/2002  09:45 PM    <DIR>          WINDOWS
10/31/2004  10:21 PM         7,864,320 ntuser.dat
05/01/2003  11:16 PM               274 prom_album
08/09/2004  10:42 AM           224,674 ~
01/26/2004  09:00 PM                97 LuResult.txt
               4 File(s)      8,089,365 bytes
               7 Dir(s)   2,571,747,328 bytes free

C:\Documents and Settings\default>_
```

Menu-driven



Graphical



Questions?