PCCE07T

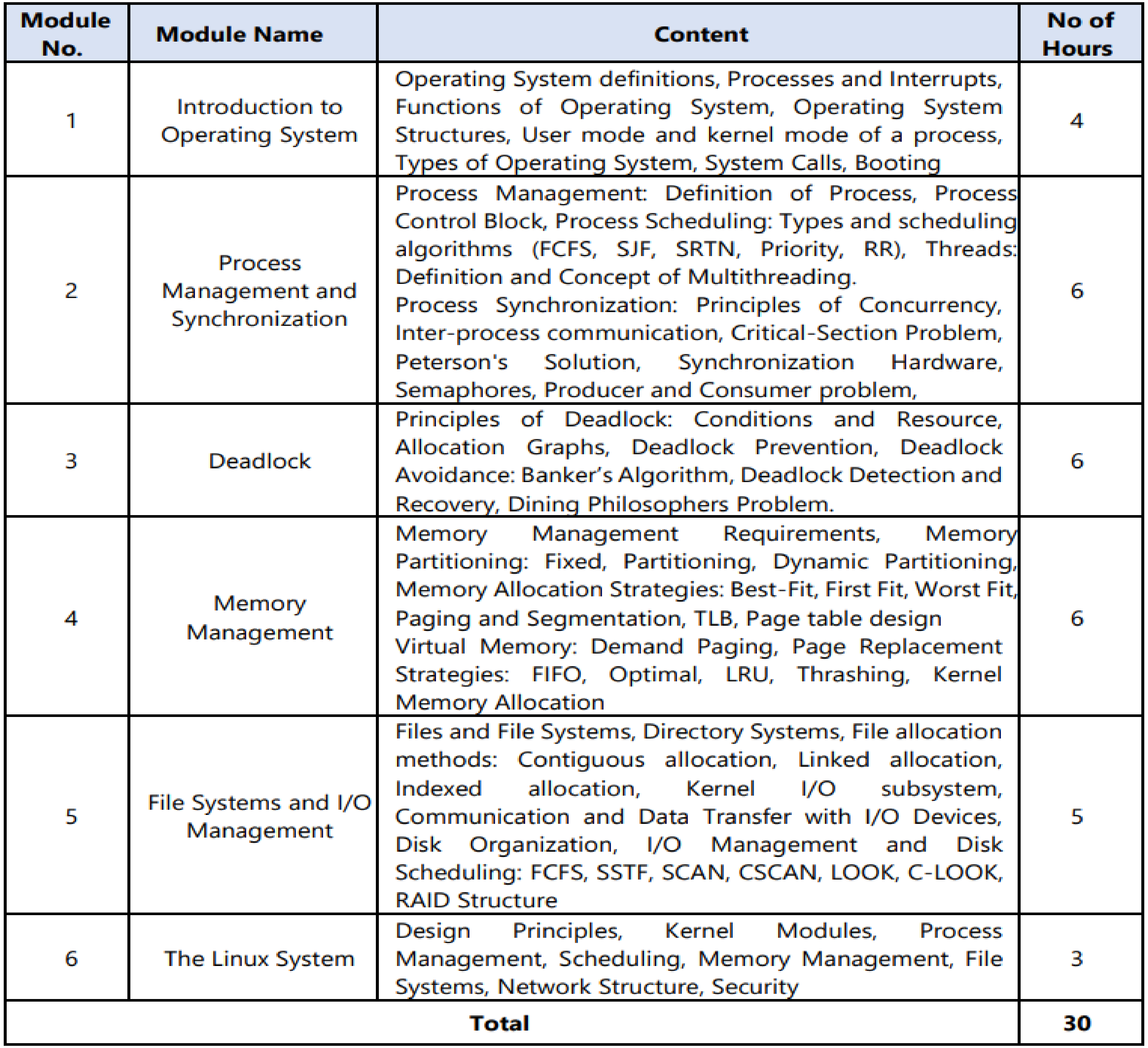
Operating Systems

Course Instructor: Pankaj Vanwari

Credits: 2

Duration: 30 hours contact + 30 hours self study.

Course Contents



Module No.	Module Name	Content	No of Hours
1	Introduction to Operating System	Operating System definitions, Processes and Interrupts, Functions of Operating System, Operating System Structures, User mode and kernel mode of a process, Types of Operating System, System Calls, Booting	4
2	Process Management and Synchronization	Process Management: Definition of Process, Process Control Block, Process Scheduling: Types and scheduling algorithms (FCFS, SJF, SRTN, Priority, RR), Threads: Definition and Concept of Multithreading. Process Synchronization: Principles of Concurrency, Inter-process communication, Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Producer and Consumer problem,	6
3	Deadlock	Principles of Deadlock: Conditions and Resource, Allocation Graphs, Deadlock Prevention, Deadlock Avoidance: Banker's Algorithm, Deadlock Detection and Recovery, Dining Philosophers Problem.	6
4	Memory Management	Memory Management Requirements, Memory Partitioning: Fixed, Partitioning, Dynamic Partitioning, Memory Allocation Strategies: Best-Fit, First Fit, Worst Fit, Paging and Segmentation, TLB, Page table design Virtual Memory: Demand Paging, Page Replacement Strategies: FIFO, Optimal, LRU, Thrashing, Kernel Memory Allocation	6
5	File Systems and I/O Management	Files and File Systems, Directory Systems, File allocation methods: Contiguous allocation, Linked allocation, Indexed allocation, Kernel I/O subsystem, Communication and Data Transfer with I/O Devices, Disk Organization, I/O Management and Disk Scheduling: FCFS, SSTF, SCAN, CSCAN, LOOK, C-LOOK, RAID Structure	5
6	The Linux System	Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management, File Systems, Network Structure, Security	3
Total			30

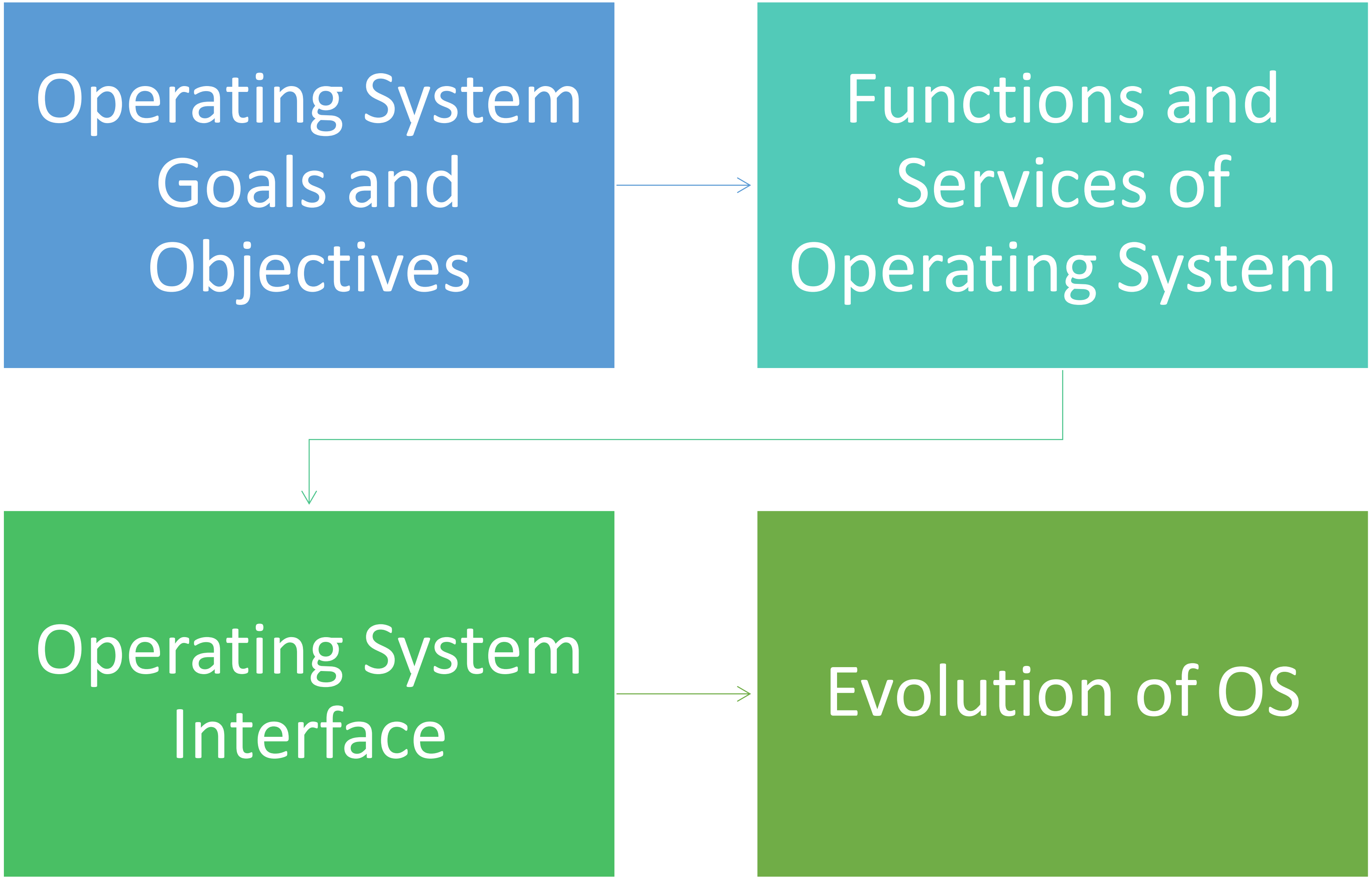
Assessment of 75 Marks

- ISA = 15
  - Top 10 of 11 Assignments for 10 Marks
  - Online Course for 5 Marks
- MSE = 20
  - Average of 2 MSEs
  - MSE1 Module 1 and Partial Module 2
  - MSE2 Remaining Module 2, Module 3 and Module 4
- ESE = 40
  - Regular

Module 1_Part 1

Introduction to Operating System

Contents



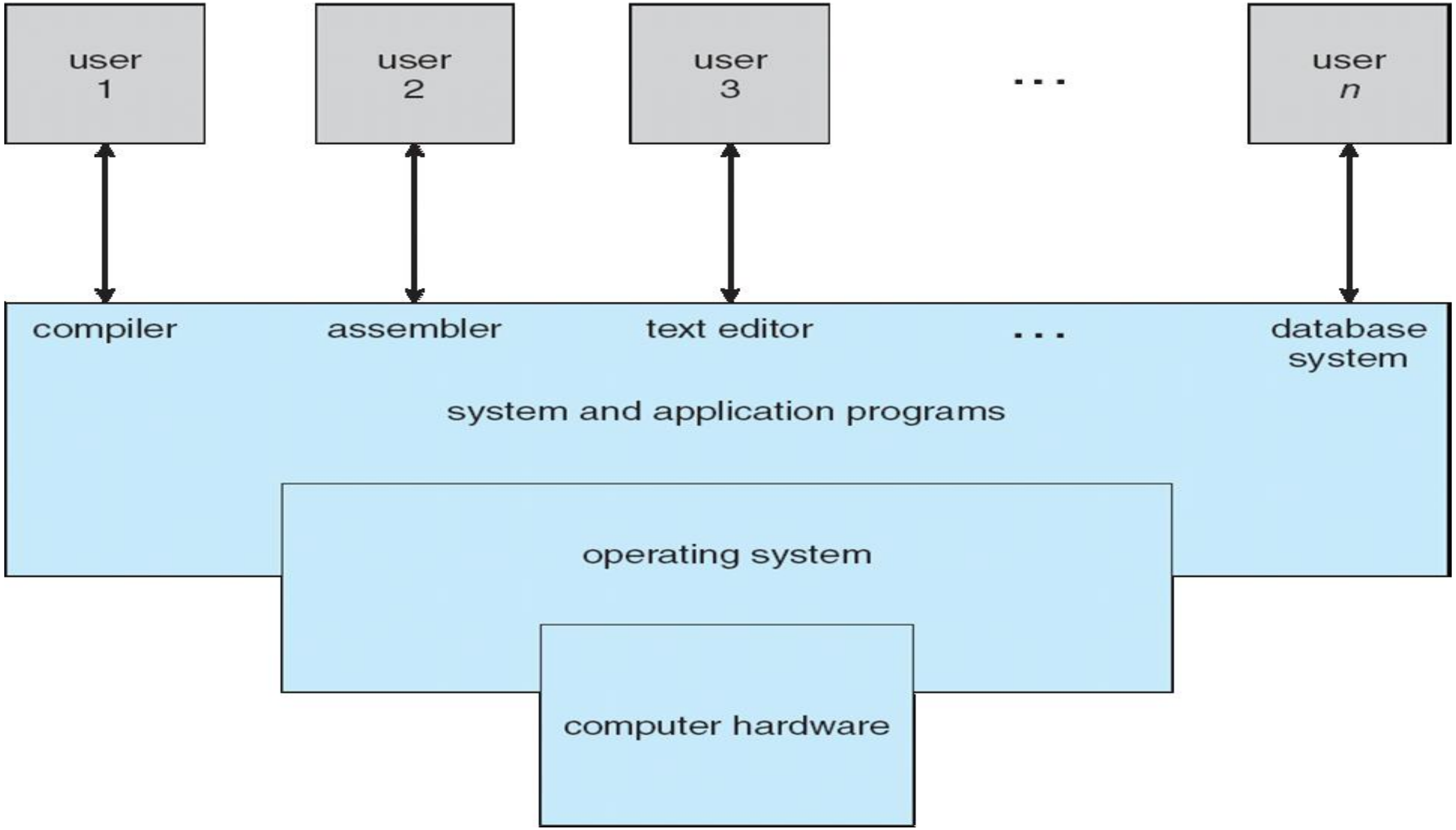
Operating System Goals and Objectives

Functions and Services of Operating System

Operating System Interface

Evolution of OS

4 layers of Computer System



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/ Objectives**:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Operating System Functions and Services

Functions of Operating System

- OS is a **Resource Manager**
    - Allocates/ Deallocates all resources
        - Process Management for Processor(s)
        - Memory Management
        - File Management
        - Storage Management
        - I/O Management
    - Decides between conflicting requests for efficient and fair resource use
- OS is a **Control Program**
    - Controls execution of programs to prevent errors and improper use of the computer

Process Management

- The operating system is responsible for the following activities in connection with process management:
  - Scheduling processes and threads on the CPUs
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication

Memory Management

- The operating system is responsible for the following activities in connection with memory management:
  - Keeping track of which parts of memory are currently being used and who is using them
  - Deciding which processes (or parts of processes) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

File Management

- The operating system is responsible for the following activities in connection with file management:
  - Creating and deleting files
  - Creating and deleting directories to organize files
  - Supporting primitives for manipulating files and directories
  - Mapping files onto secondary storage
  - Backing up files on stable (nonvolatile) storage media

Storage Management

- Also known as Mass-Storage/ Secondary Storage Management.
- The operating system is responsible for the following activities in connection with disk management:
  - Free-space management
  - Storage allocation
  - Disk scheduling

I/O Management

- Also known as I/O subsystem of OS.
- One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. For example, in UNIX, the peculiarities of I/O devices are hidden from the bulk of the operating system itself by the I/O subsystem.
- The I/O subsystem consists of several components:
  - A memory-management component that includes buffering, caching, and spooling
  - A general device-driver interface
  - Drivers for specific hardware devices
  - Only the device driver knows the peculiarities of the specific device to which it is assigned.

Operating System Services

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

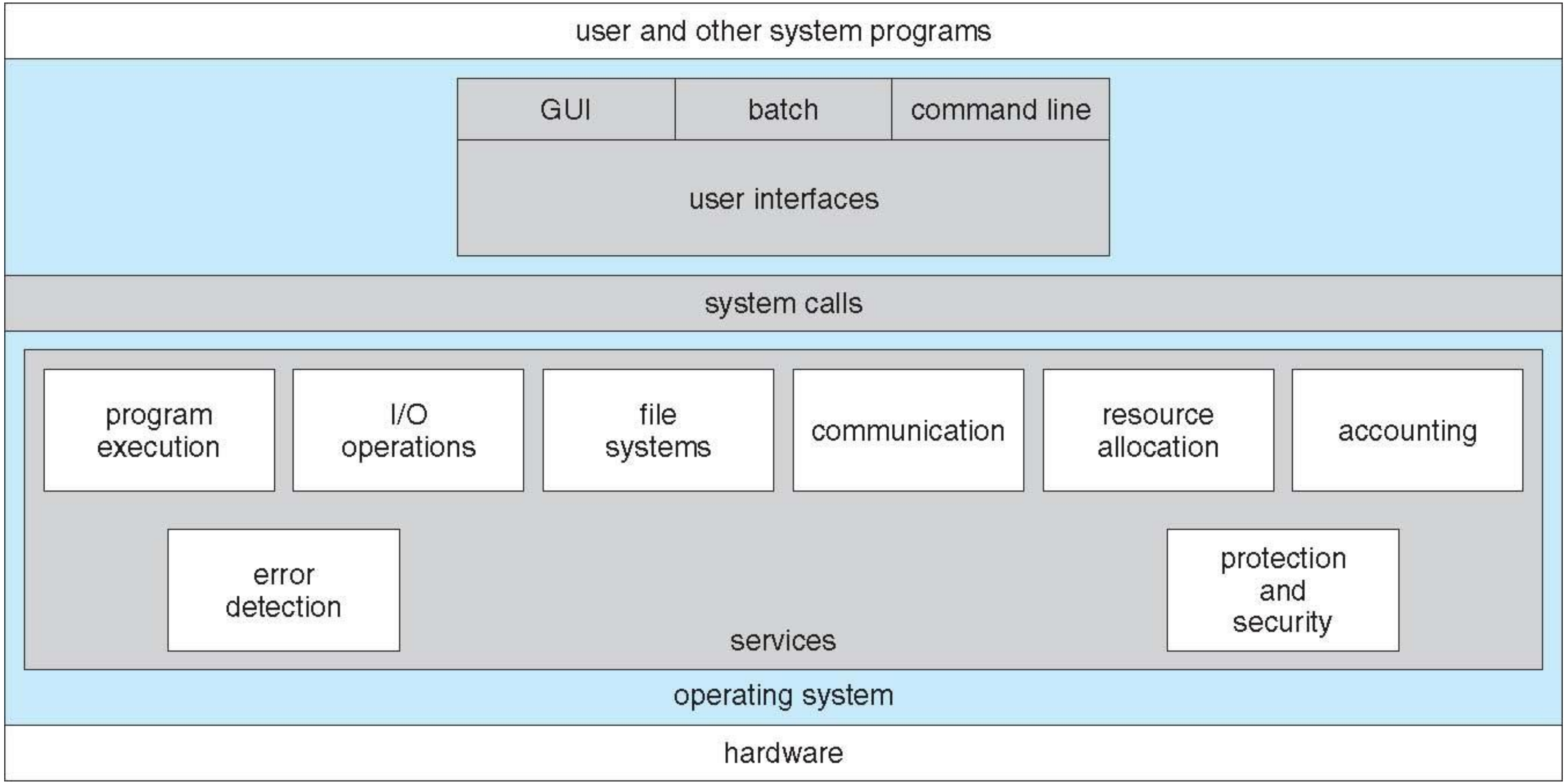
Operating System Services (Cont.)

- One set of operating-system services provides functions that are helpful to the user (Cont.):
  - **File-system manipulation** - The file system is of particular interest. Programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.
  - **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.)

- 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 -   CPU cycles, main memory, file storage, I/O devices.
  - **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

A View of Operating System Services



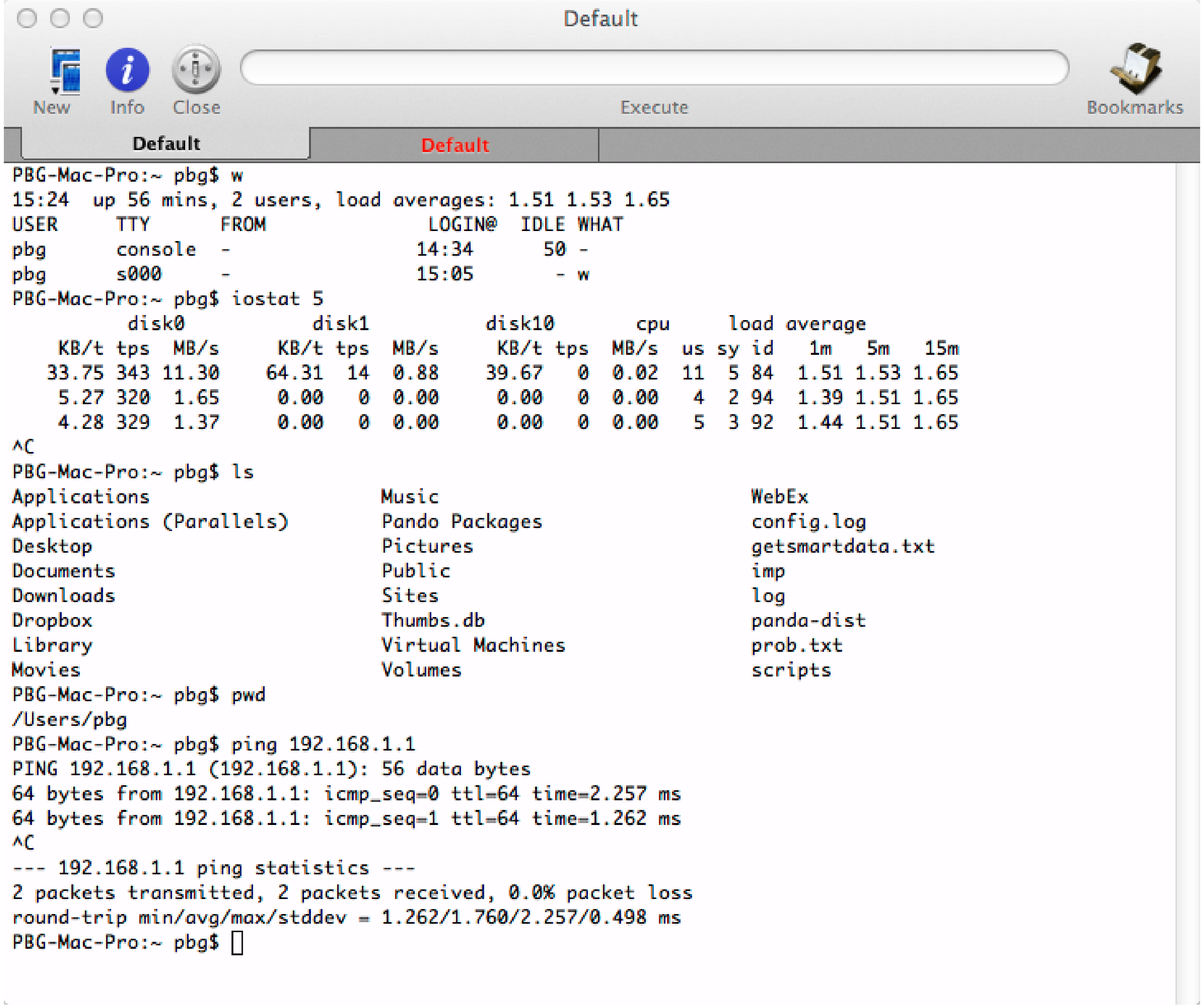
Operating System Interface

User Operating System Interface - CLI

CLI or **command interpreter** allows direct command entry

- Sometimes implemented in kernel, sometimes by systems program
- Sometimes multiple flavors implemented – **shells**
- Primarily fetches a command from user and executes it
- Sometimes commands built-in, sometimes just names of programs
  - If the latter, adding new features doesn’t require shell modification

Bourne Shell Command Interpreter



```
PBG-Mac-Pro:~ pbg$ w
15:24  up 56 mins, 2 users, load averages: 1.51 1.53 1.65
USER     TTY      FROM              LOGIN@  IDLE WHAT
pbg      console  -                14:34      50 -
pbg      s000     -                15:05       - w
PBG-Mac-Pro:~ pbg$ iostat 5
          disk0             disk1             disk10       cpu     load average
    KB/t tps  MB/s     KB/t tps  MB/s     KB/t tps  MB/s  us sy id   1m   5m   15m
   33.75 343 11.30    64.31  14  0.88    39.67   0  0.02  11  5 84  1.51 1.53 1.65
    5.27 320  1.65     0.00   0  0.00     0.00   0  0.00   4  2 94  1.39 1.51 1.65
    4.28 329  1.37     0.00   0  0.00     0.00   0  0.00   5  3 92  1.44 1.51 1.65
^C
PBG-Mac-Pro:~ pbg$ ls
Applications                    Music                           WebEx
Applications (Parallels)        Pando Packages                  config.log
Desktop                         Pictures                        getsmartdata.txt
Documents                       Public                          imp
Downloads                       Sites                           log
Dropbox                         Thumbs.db                       panda-dist
Library                         Virtual Machines                prob.txt
Movies                          Volumes                         scripts
PBG-Mac-Pro:~ pbg$ pwd
/Users/pbg
PBG-Mac-Pro:~ pbg$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=2.257 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.262 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.262/1.760/2.257/0.498 ms
PBG-Mac-Pro:~ pbg$ 
```

User Operating System Interface - GUI

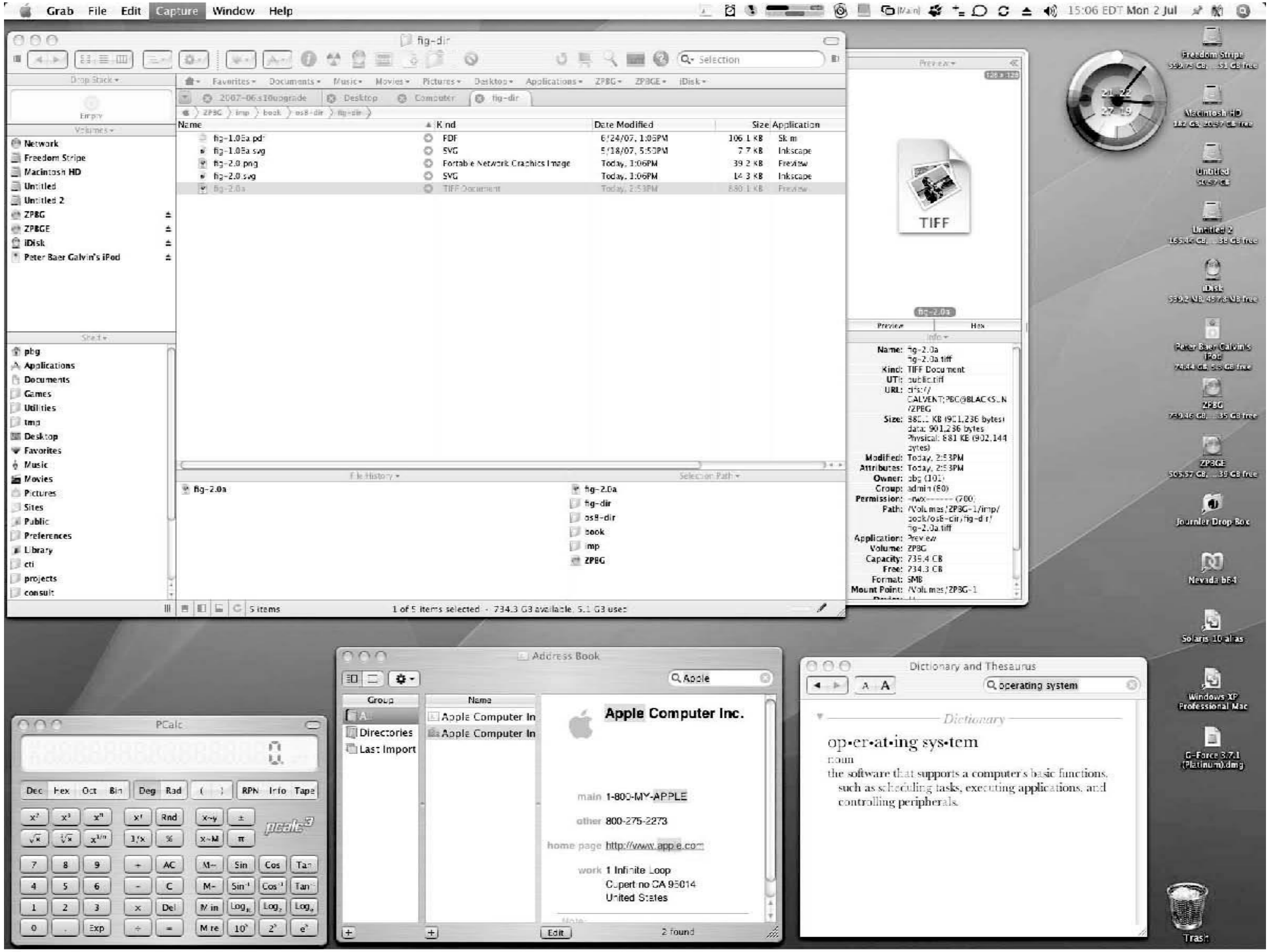
User-friendly **desktop** metaphor interface

- Usually mouse, keyboard, and monitor
- **Icons** represent files, programs, actions, etc
- Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory (known as a **folder**)
- Invented at Xerox PARC

Many systems now include both CLI and GUI interfaces

- Microsoft Windows is GUI with CLI “command” shell
- Apple Mac OS X is “Aqua” GUI interface with UNIX kernel underneath and shells available
- Unix and Linux have CLI with optional GUI interfaces (CDE, KDE, GNOME)

The Mac OS X GUI



Name	Kind	Date Modified	Size	Application
fig-1.05a.pdf	PDF	6/24/07, 1:05PM	106.1 KB	Skim
fig-1.05a.svg	SVG	5/18/07, 5:53PM	7.7 KB	Inkscape
fig-2.0.png	Portable Network Graphics Image	Today, 1:06PM	39.2 KB	Preview
fig-2.0.svg	SVG	Today, 1:06PM	14.3 KB	Inkscape
fig-2.0a	TIFF Document	Today, 2:53PM	880.1 KB	Preview

5 items   1 of 5 items selected · 734.3 GB available, 5.1 GB used

Name: fig-2.0a
fig-2.0a.tiff
Kind: TIFF Document
UTI: public.tiff
URL: cifs://CALVENT;PBG@BLACKSUN/ZPBG
Size: 880.1 KB (901,236 bytes)
data: 901,236 bytes
Physical: 881 KB (902,144 bytes)
Modified: Today, 2:53PM
Attributes: Today, 2:53PM
Owner: pbg (101)
Group: admin (80)
Permission: -rwx------ (700)
Path: /Volumes/ZPBG-1/imp/book/os8-dir/fig-dir/fig-2.0a.tiff
Application: Preview
Volume: ZPBG
Capacity: 735.4 GB
Free: 734.3 GB
Format: SMB
Mount Point: /Volumes/ZPBG-1

Apple Computer Inc.

main 1-800-MY-APPLE
other 800-275-2273
home page http://www.apple.com
work 1 Infinite Loop
Cupertino CA 95014
United States

2 found

Dictionary

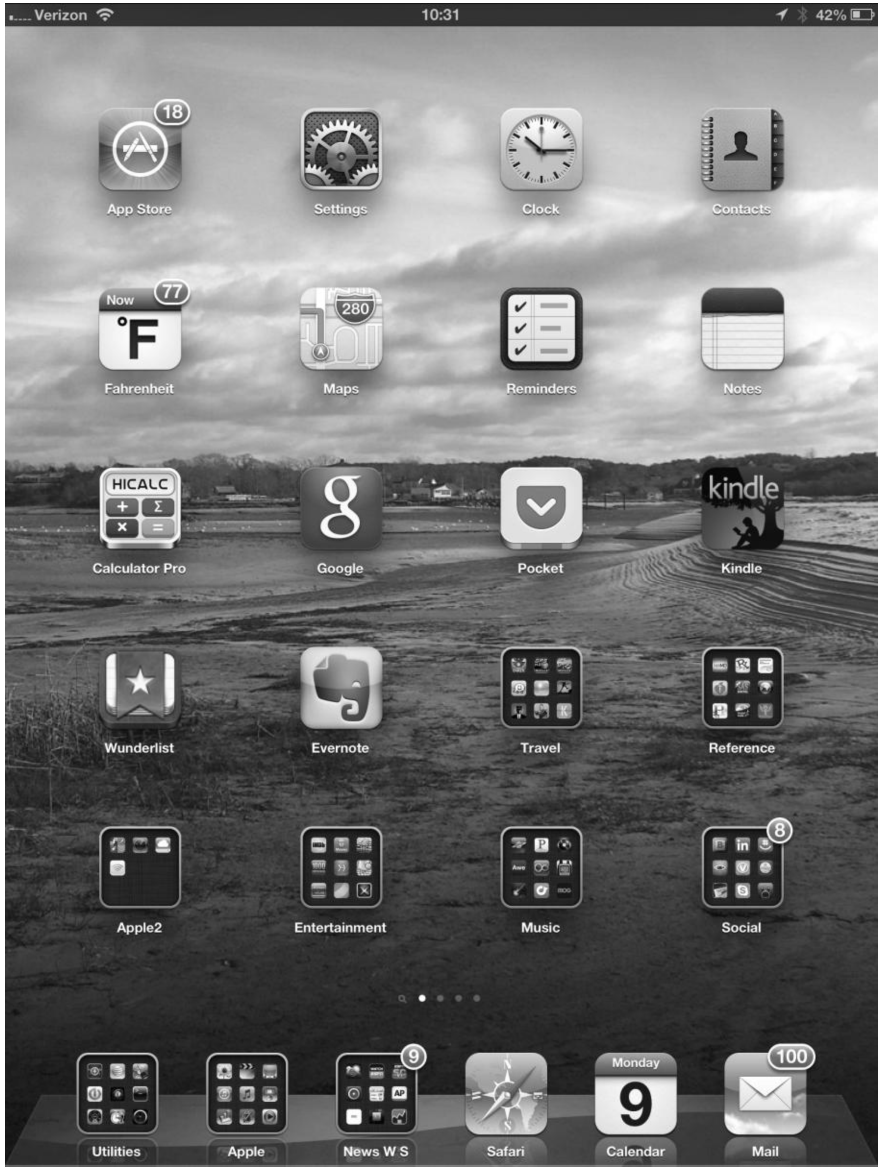
op•er•at•ing sys•tem
noun
the software that supports a computer's basic functions, such as scheduling tasks, executing applications, and controlling peripherals.

Touchscreen Interfaces

Touchscreen devices require new interfaces

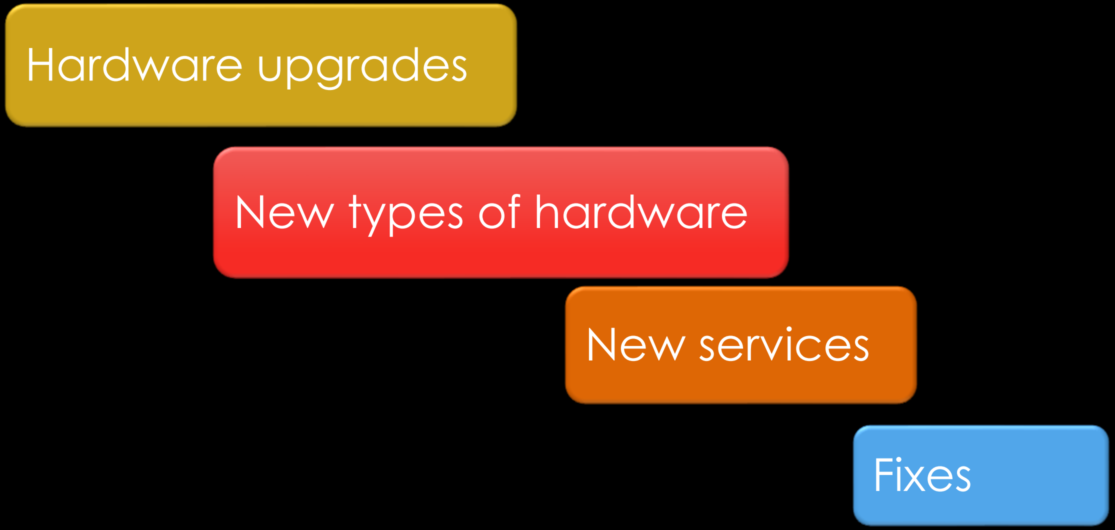
- Mouse not possible or not desired
- Actions and selection based on gestures
- Virtual keyboard for text entry

Voice commands.



EVOLUTION OF OPERATING SYSTEMS

- A major OS will evolve over time for a number of reasons:



Hardware upgrades

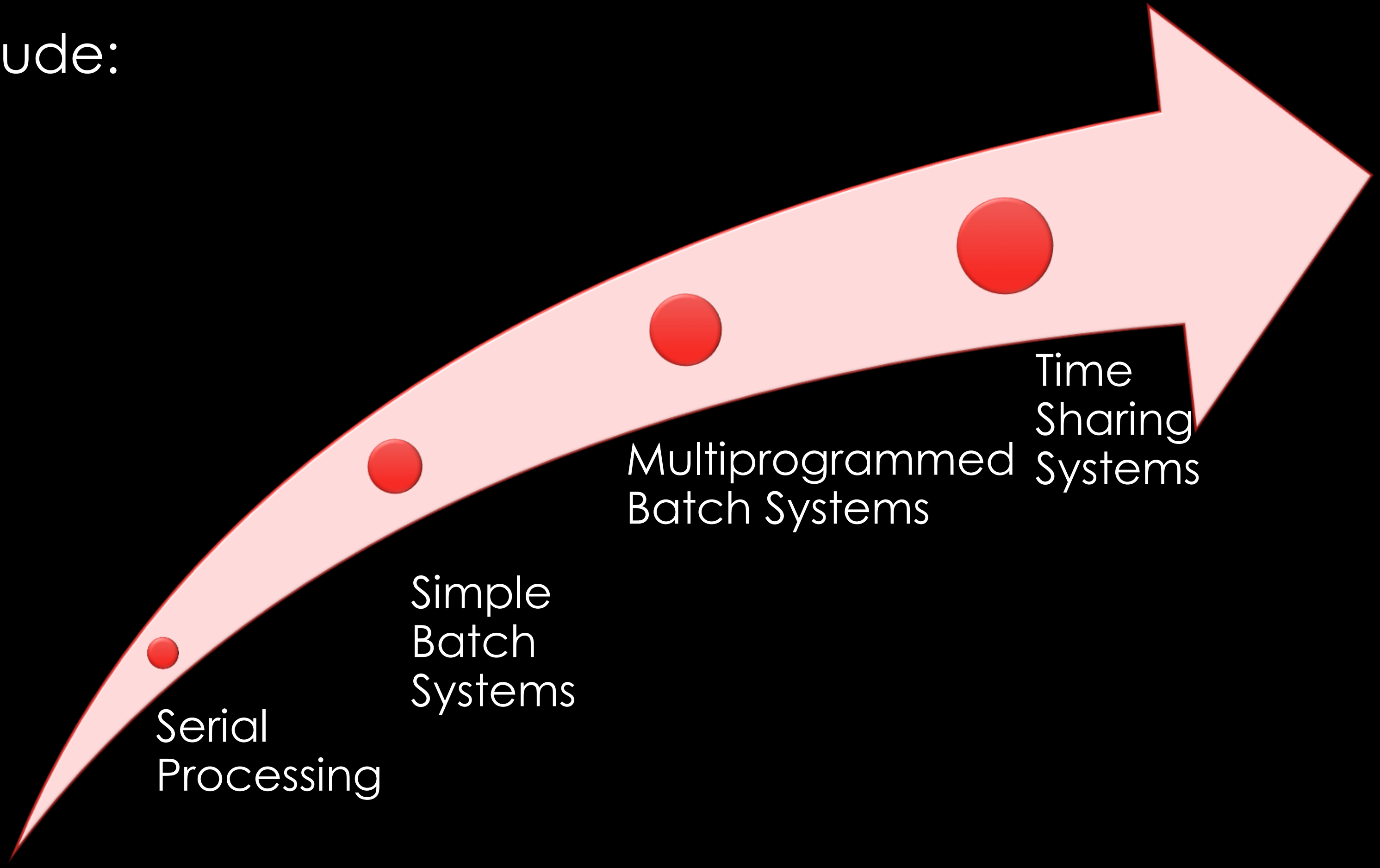
New types of hardware

New services

Fixes

EVOLUTION/TYPES OF OPERATING SYSTEMS

- Stages include:

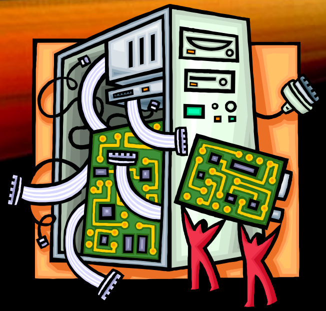


Serial Processing

Simple Batch Systems

Multiprogrammed Batch Systems

Time Sharing Systems



SERIAL PROCESSING

Earliest Computers:

- No operating system
  - programmers interacted directly with the computer hardware
- Computers ran from a console with display lights, toggle switches, some form of input device, and a printer
- Users have access to the computer in "series"

Problems:

- Scheduling:
  - most installations used a hardcopy sign-up sheet to reserve computer time
    - time allocations could run short or long, resulting in wasted computer time
- Setup time
  - a considerable amount of time was spent just on setting up the program to run

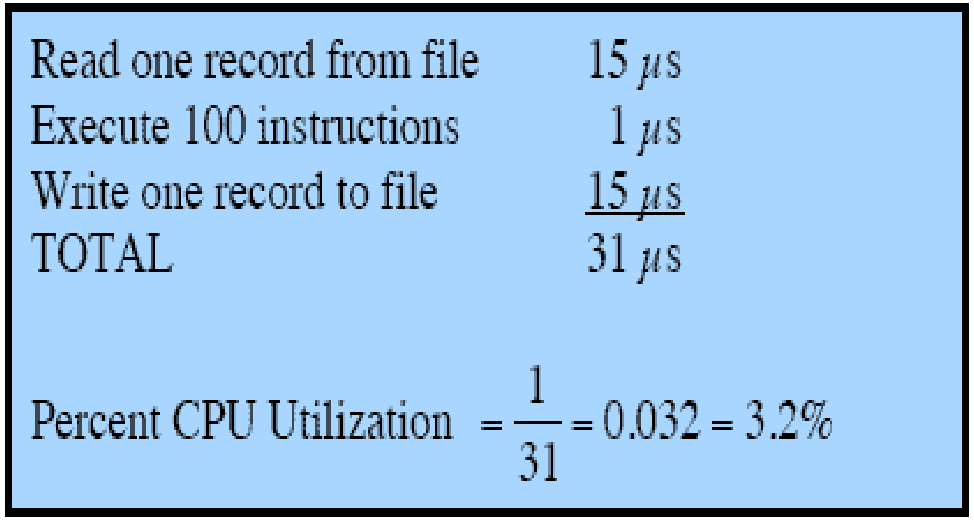
SIMPLE BATCH SYSTEMS

- Early computers were very expensive
  - important to maximize processor utilization
- Monitor
  - user no longer has direct access to processor
  - job is submitted to computer operator who batches them together and places them on an input device
  - program branches back to the monitor when finished

SIMPLE BATCH SYSTEM OVERHEAD

- Processor time alternates between execution of user programs and execution of the monitor
- Sacrifices:
  - some main memory is now given over to the monitor
  - some processor time is consumed by the monitor
- Despite overhead, the simple batch system improves utilization of the computer.

MULTIPROGRAMMED BATCH SYSTEMS



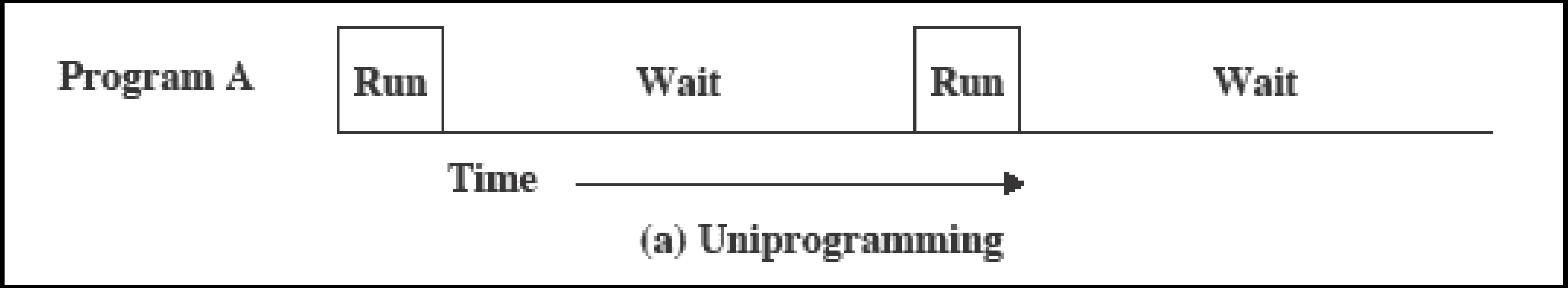
Read one record from file	15 µs
Execute 100 instructions	1 µs
Write one record to file	15 µs
TOTAL	31 µs

$$\text{Percent CPU Utilization} = \frac{1}{31} = 0.032 = 3.2\%$$

Figure 2.4 System Utilization Example

- Processor is often idle
  - even with automatic job sequencing
  - I/O devices are slow compared to processor

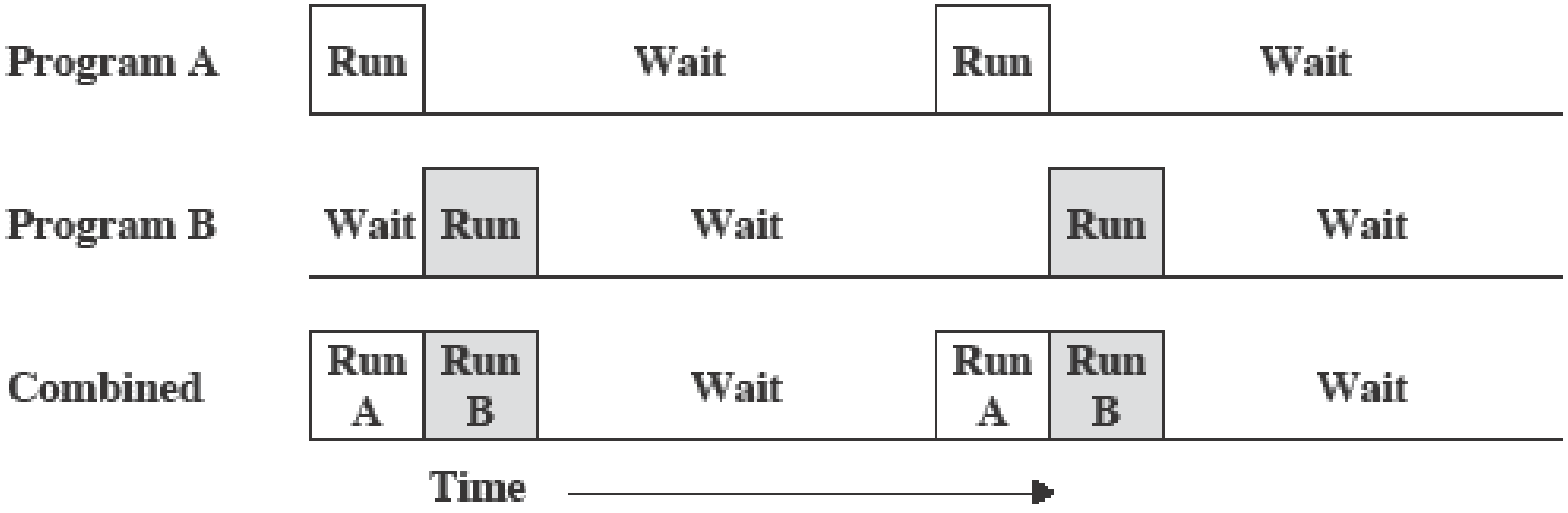
UNIPROGRAMMING



(a) Uniprogramming

- The processor spends a certain amount of time executing, until it reaches an I/O instruction; it must then wait until that I/O instruction concludes before proceeding

MULTIPROGRAMMING



Program A: Run | Wait | Run | Wait

Program B: Wait | Run | Wait | Run | Wait

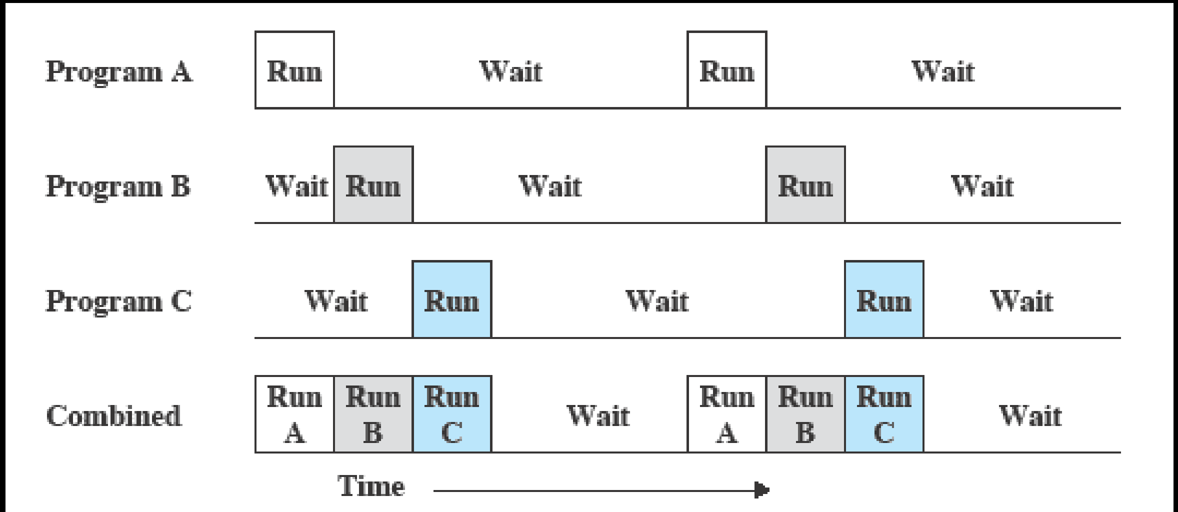
Combined: Run A | Run B | Wait | Run A | Run B | Wait

Time

(b) Multiprogramming with two programs

- There must be enough memory to hold the OS (resident monitor) and one user program
- When one job needs to wait for I/O, the processor can switch to the other job, which is likely not waiting for I/O

MULTIPROGRAMMING



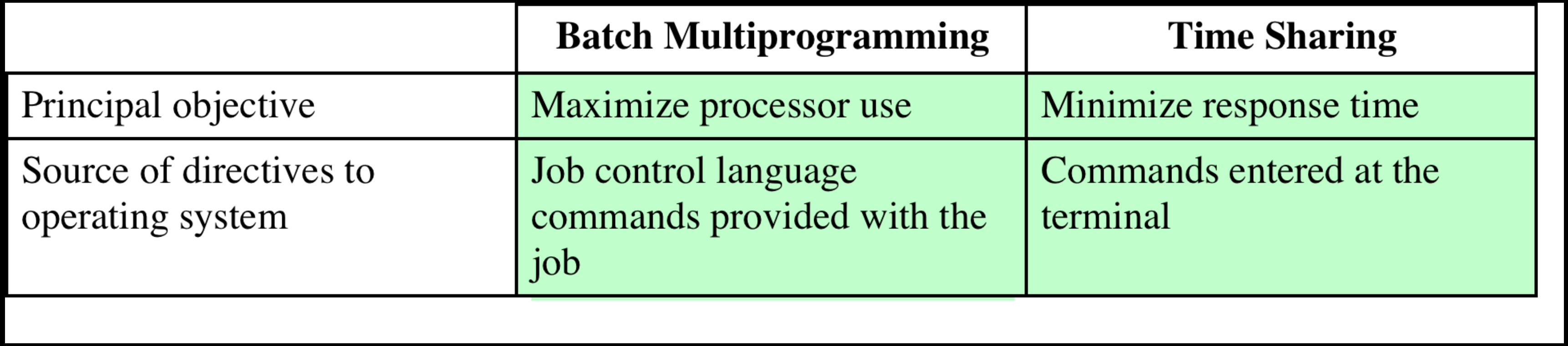
(c) Multiprogramming with three programs

- Multiprogramming
  - also known as multitasking
  - memory is expanded to hold three, four, or more programs and switch among all of them

TIME-SHARING SYSTEMS

- Can be used to handle multiple interactive jobs
- Processor time is shared among multiple users
- Multiple users simultaneously access the system through terminals, with the OS interleaving the execution of each user program in a short burst or quantum of computation

BATCH MULTIPROGRAMMING VS. TIME SHARING



	Batch Multiprogramming	Time Sharing
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

Table 2.3 Batch Multiprogramming versus Time Sharing

Summary

- 4 Layers of Computer System
- OS acts as an intermediate layer between hardware and other programs thereby providing a layer of abstraction.
- It acts as a resource allocator and control program.
- OS Functions
- OS Services
- Types of Interface
- Evolution of OS

Thank You

Assignment 1 to be submitted by next Thursday.