Introduction to Operating Systems

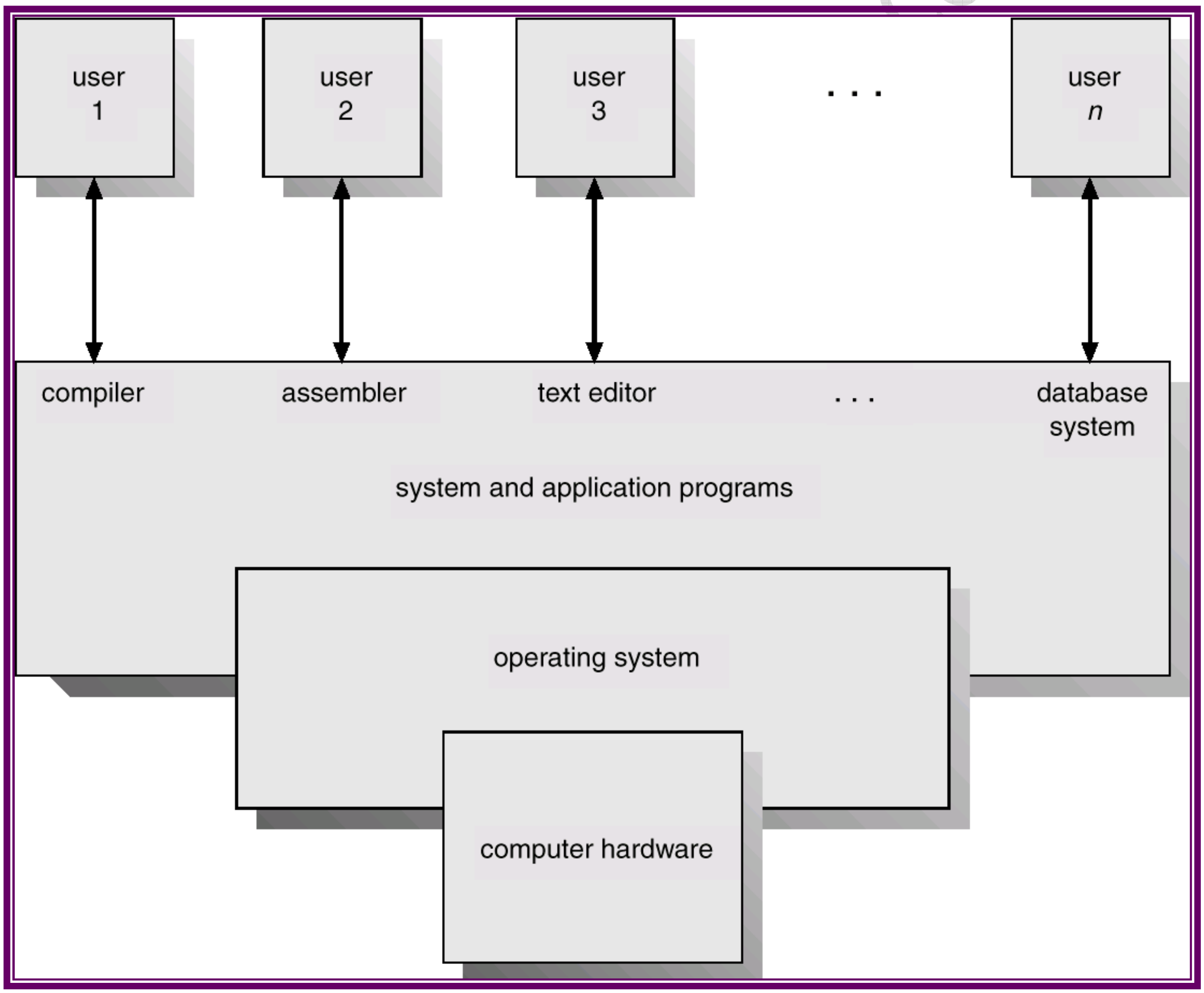
Computer Software is of two types:
(1) System Software (Manages operations of computers)
(2) Application Software (Solves problems of users)

An example of an important system software is **Operating System.**

What is an Operating System?

An operating system is a set of routines to manage the running of the computer. It controls all computer resources and provides a base upon which application programs can be written.

(Abstract view of the components of a computer system)



Why do we need an operating system?

- Convenience – operating system is an extended machine i.e.
  1st Gen → 2nd Gen → 3rd Gen → 4th Generation computers
- Efficiency – Operating system as a resource allocator

Functions of an Operating System:

- Job scheduling and management
- Scheduling of input and output operations
- Operator communication
- Job-to-job communication
- Interrupt handling
- Process management
- Memory management
- File management

History of Operating Systems

→ Early systems were "Bare machines" {1940's}
- programming done in machine language
- programmer was operator

Drawbacks:

- Idle time
- Common functions programmed by all.

→ Additional hardware and software developed {1950's}
- hardware implied card readers, line printers, paper tape
- software implied compilers, libraries of common functions

Steps included load compiler, run compiler, unload compiler, load object program

Drawbacks:

- Operations became more complex
- Too much setup time and cpu idle

→ Simple Monitor
- Utilization improvement
- Professional operators hired which meant less idle cpu reduced setup time increased efficiency and decreased convenience for the programmer.

→ Batch Systems (First rudimentary operating system) also called Mainframe systems
- automatic job sequencing (resident monitor)
- resident monitor consisted of control card interpreter, loader and job sequences, device drivers, interrupt handling routines and automatic job sequencing.

Performance:

- Automatic job sequencing used to reduce setup time
- Improved I / O devices speed (mechanical to electromechanical)

Offline operation features

- cards copied to magnetic tape for input
- contents on tape and printed later

Multi programming (Parallel processing)

- denotes an operating system to increase cpu utilization (by having more than one program in memory at the same time) by executing programs concurrently.

P1	Idle	P1	Idle	P1	Idle	P2	idle	P2	Idle	P2	Idle

←-------------------------- program 1 ----------------→←-------------------------- program 2 ---------------→

Sequential Execution time $t = t1 + t2$

P1	Idle	P1	Idle	P1	Idle

Program 1

P2	idle	P2	Idle	P2	Idle

Program 2 ←--θ →

Total time, $t = t1 + \theta$ (multiprogrammed execution)

Result: Throughput increases

Degree of Multiprogramming

Number of programs actively competing for resources (number of programs in memory).

Higher the degree of multiprogramming ➔ higher the resources utilization

Multiaccess or Multiuser operating system

An operating system that allows simultaneous access to computer system through two or more terminals.

➔ A multiuser operating system does not imply multiprogramming

Serial Processing

Resources of a computer system may be dedicated to a single program until its completion.

Time-Sharing Systems

Early Batch systems – "batching" of similar jobs (phasing)
(card / tape systems allowed only sequential access to programs and data)

- lack of user interaction during execution
- turn around time large
- inconvenient to user (handling all possible outcome / errors)

Batch systems appropriate for executing large jobs which need little interaction.

Interactive (Hands – on) Systems

- Implies online communication between user and system.
- Smaller jobs
- Response time should be small

Time shared systems provide interactive use of a system to many users at the same time (by providing user with a part of CPU time)

Time sharing ➔ Multiprogramming
🡐 (reverse not true)

Multiuser / Multiaccess systems ➔ Multiprogramming

Multiprogramming ➔ Multiprocessing systems
🡐 (reverse not true)

Real Time Systems

Processing must be done within a fixed time constraint or system will fail.

➔ Is unix a real time system ? (No)

- Telephone switching systems, flight control
- Objective is to provide quick event response time
- User convenience and response utilization

Separate process is charged with handling a specific event. Process is activated upon occurance of the event.

Hands- on Component

Introduction to Unix / Linux and Windows Operating Systems

****