# Module 12

## Operating System

# Module Twelve

- This week, we are going to talk about :

- Operating Systems
  - Basic Function : Make the Computer easier to Use
  - Components
- Hardware Dependent Tasks
  - Interrupt Processing
  - Process Scheduling
  - I/O Supervision
  - Memory Management

- Material adapted from L. Beck, System Software,1997
- Selected information and material adapted from[2] Dr. H. Malcom, 2008

# What is an Operating System?

- A **program** that acts as an intermediary between the user of a computer and the computer hardware.

- Operating system goals:

  Execute user programs and make solving user problems easier.

  Make the computer system convenient for people to use.

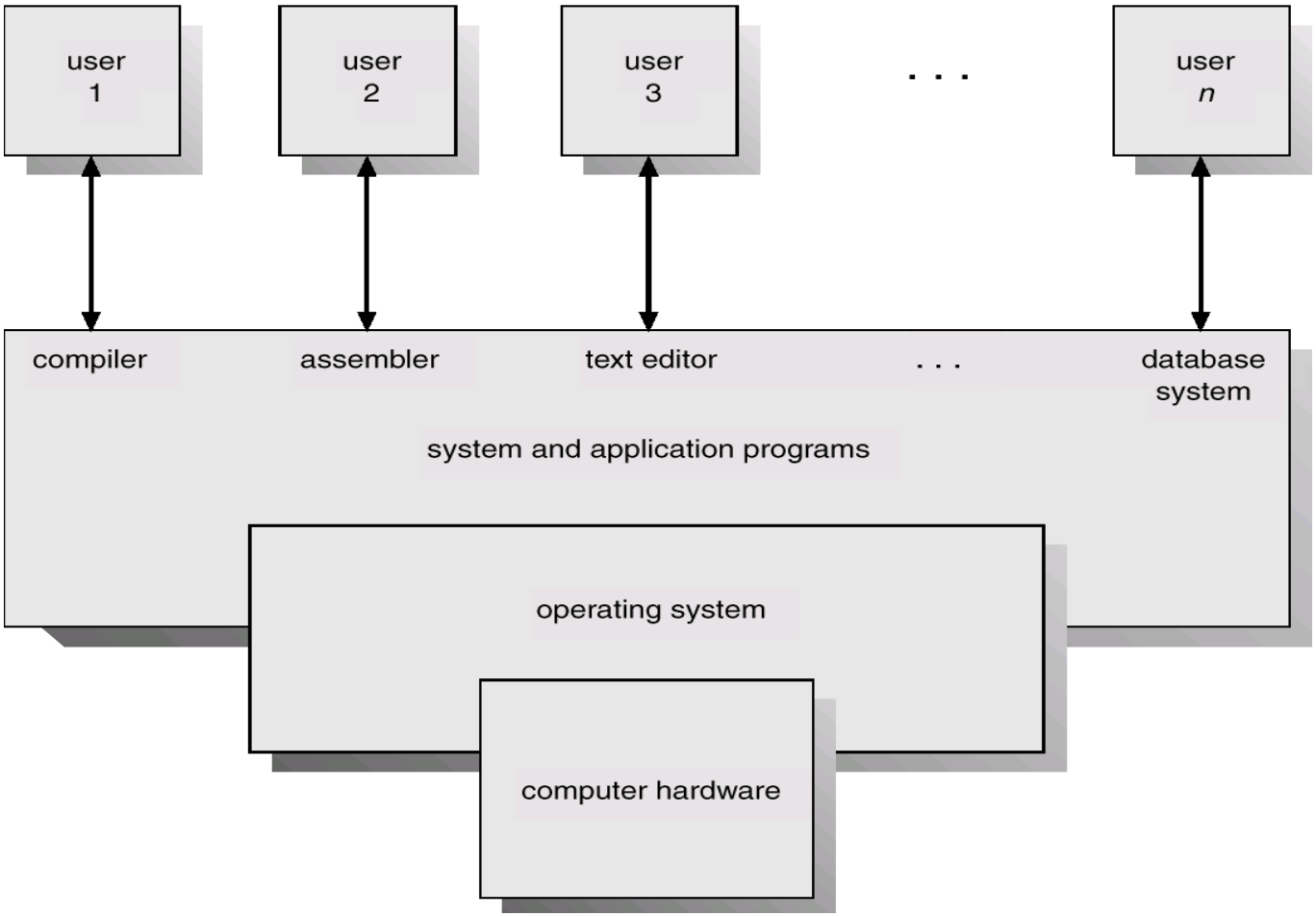
  Be efficient in the use of the computer hardware.

- Operating systems are system software, the glue that holds everything together.

# Why Operating Systems

- The biggest and fastest computer in the world is of no use if it cannot efficiently provide beneficial services to its users.
- Users see the computer through their application programs. These programs are ultimately executed by computer hardware.

# Abstract View of System Components



# User Interface

- First Task
- More friendly than the hardware.
- Used by :
  - Programmers
  - Machine Operators
- Management of computer resources

# Run Time Environment

- Second Task
- Service routines
  - Input / Output
  - Memory management
- Illusion of separate extended machine
- Privileged instructions

# Crucial components

- Two operating system components are crucial:
  - The ***kernel*** and the ***system programs***.
- As the core of the operating system, the ***kernel*** performs scheduling, synchronization, memory management, interrupt handling and it provides security and protection.
- ***Microkernel systems*** provide minimal functionality, with most services carried out by the system programs.
- ***Monolithic systems*** provide most of their services within a single operating system program.

# More about the components

- ***Microkernel systems*** provide better security, easier maintenance, and portability at the expense of execution speed.
  - Examples: Windows 2000, Mach, and QNX.
  - Symmetric multiprocessor computers are ideal platforms for microkernel operating systems.
- ***Monolithic systems*** give faster execution speed, but are difficult to port from one architecture to another.
  - Examples are Unix, Linux, and DOS.

# BIOS

- ***Personal computer*** operating systems are designed for ease of use rather than high performance.
- The idea that revolutionized small computer operating systems was the BIOS (basic input-output operating system) chip that permitted a single operating system to function on different types of small systems.
- The BIOS takes care of the hardware details involved in addressing divergent peripheral device designs and protocols.

# Personal-Computer Systems

- *Personal computers* – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system.
- Often individuals have sole use of computer and do not need advanced CPU utilization of protection features.

# Operating System Features Needed for Multiprogramming

- I/O routine supplied by the system.
- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling – the system must choose among several jobs ready to run.
- Allocation of devices.

# Summary

- Operating Systems
  - Basic Function : Make the Computer easier to Use
  - Components

- Next: Hardware Dependent Tasks

  Interrupt Processing