Lecture 1

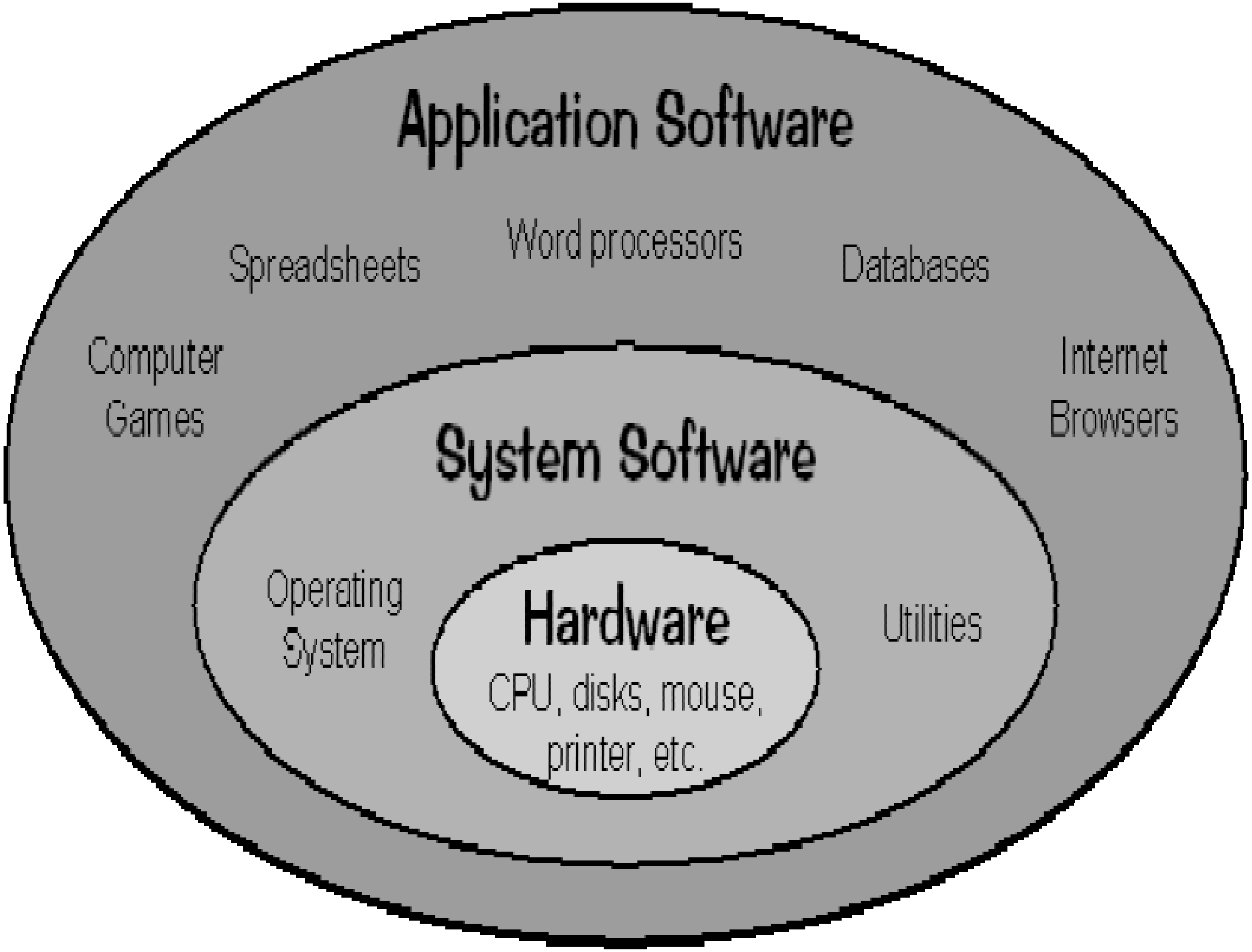
Interaction of OS and hardware, Goals of OS, Basic functions of OS, OS Services, System Calls, Types of system calls.

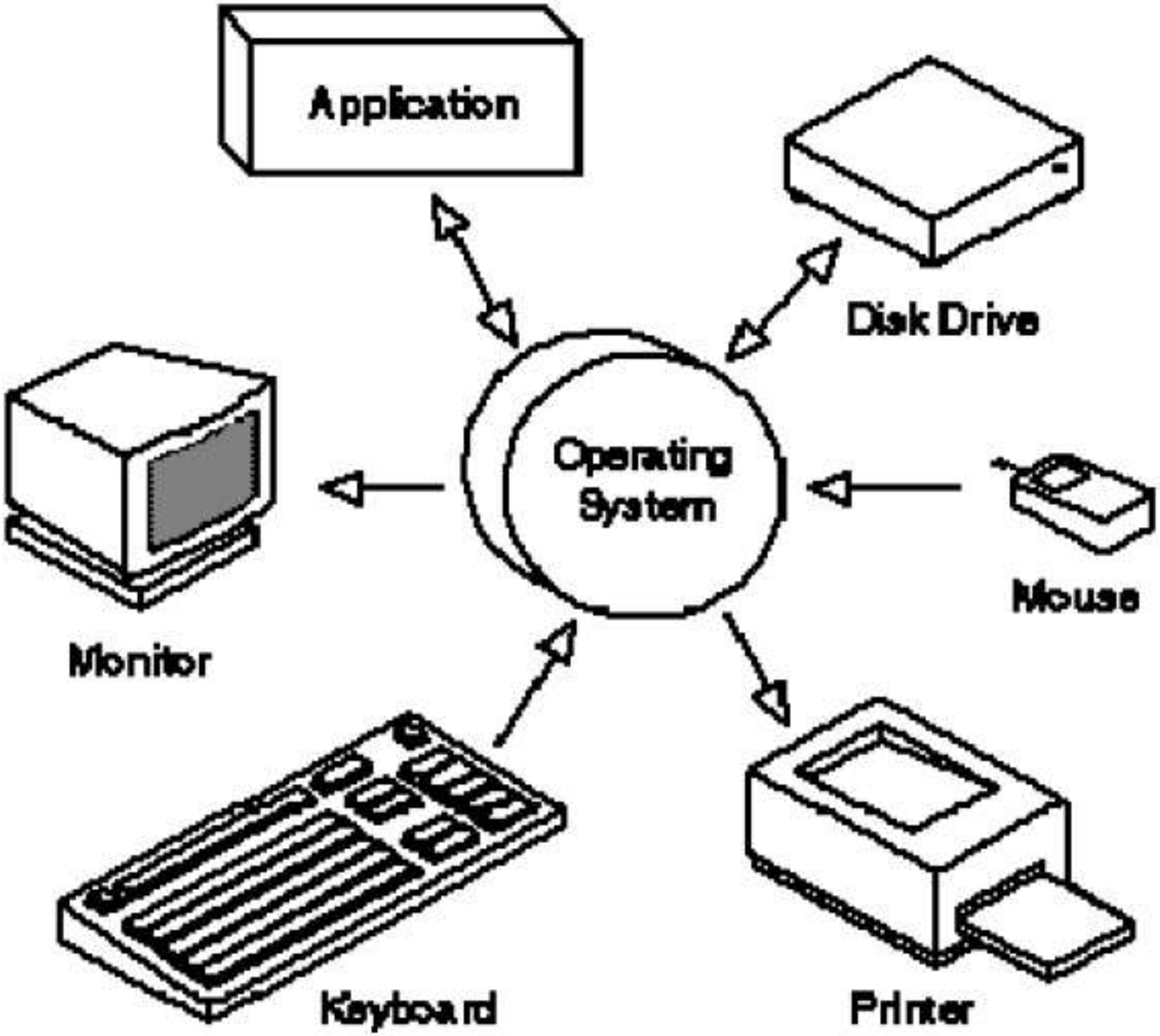
OS definition

- Operating System (OS) is system software, which acts as an interface between a user of the computer and the computer hardware. It is a supervisory program that manages hardware, processes, files, memory, I/O etc. in the system. It acts as a resource manager and
- The main purpose of an Operating System is:
  - to provide an environment in which we can execute programs.
- The main goals of the Operating System are:
  - To make the computer system convenient to use,
  - To make the use of computer hardware in efficient way.

Interaction of OS and hardware

- Scene 1: Perform basic tasks such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers – OS addresses the need for managing the computer hardware
- Scene 2: Provide a software platform on top of which other programs can run- OS provides an interface between application software and hardware


Application Software
Word processors
Spreadsheets
Databases
Computer Games
Internet Browsers
System Software
Operating System
Hardware
Utilities
CPU, disks, mouse, printer, etc.



As a user, we normally interact with the Operating System through a set of commands. The commands are accepted and executed by a part of the Operating System called the command processor or command line interpreter.

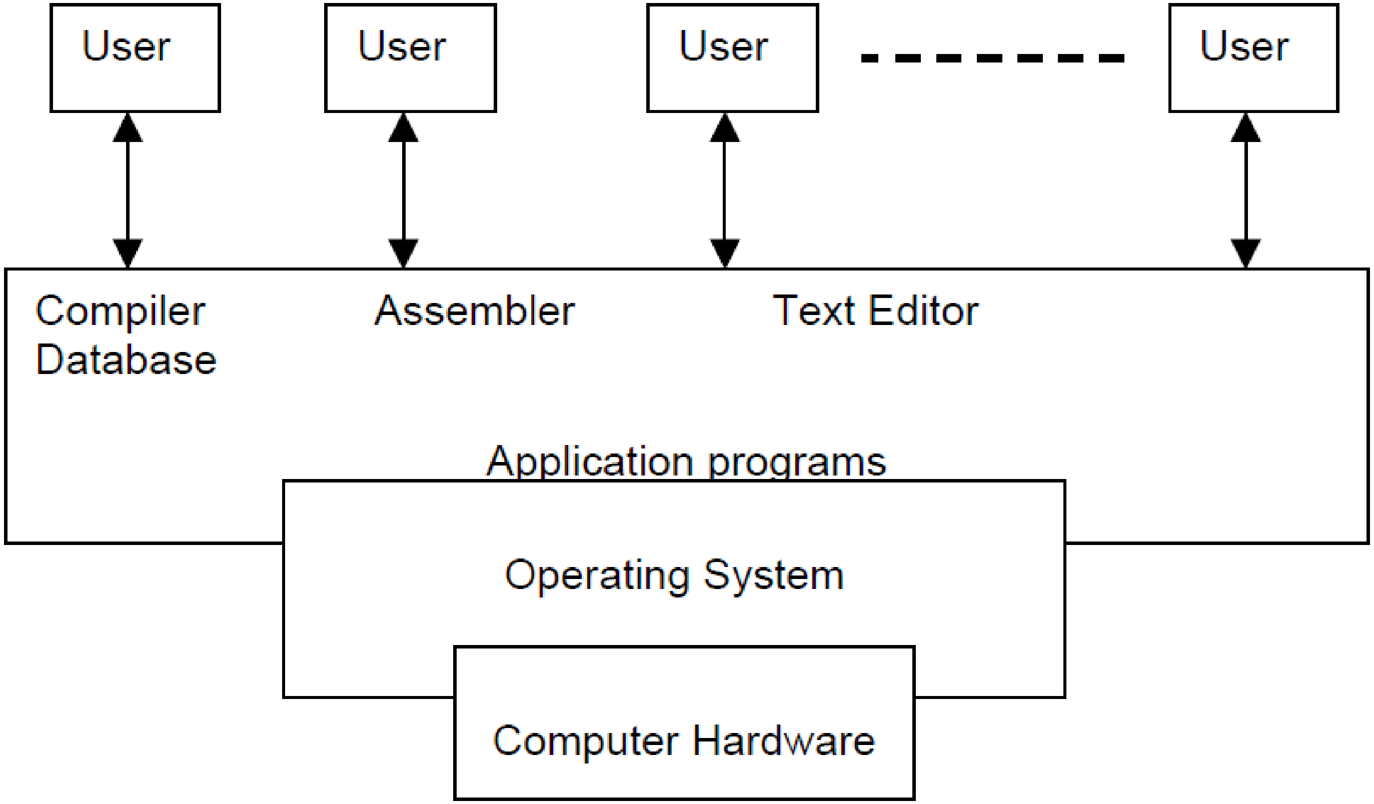
Understanding hardware of the system

- Hardware
- Software
- Main memory
- Input/output devices
- The central processing unit
- The entire computer systems can be divided into four parts or components
  - (1) The hardware
  - (2) The Operating System
  - (3) The application programs and system programs
  - (4) The users (Refer to the slide 1.5 of Galvin)

Cont..

- The Operating System controls and coordinates the use of the hardware among the various users and the application programs.

Basic view of a Computer System



Operating System is a resource allocator

- A computer system has many resources
- These resources are the CPU time, memory space, files storage space, input/output devices and so on.
- The Operating System acts as a manager of all of these resources and allocates them to the specific programs and users as needed by their tasks.
- Proper Allocation of resources

Cont…

- An Operating System can also be viewed as a control program
- control the various I/O devices and the users programs.
- execution of the user programs to prevent errors and improper use of the computer resources.
- The common functions of controlling and allocation of resources between different users and application programs is brought together into one piece of software called operating system.

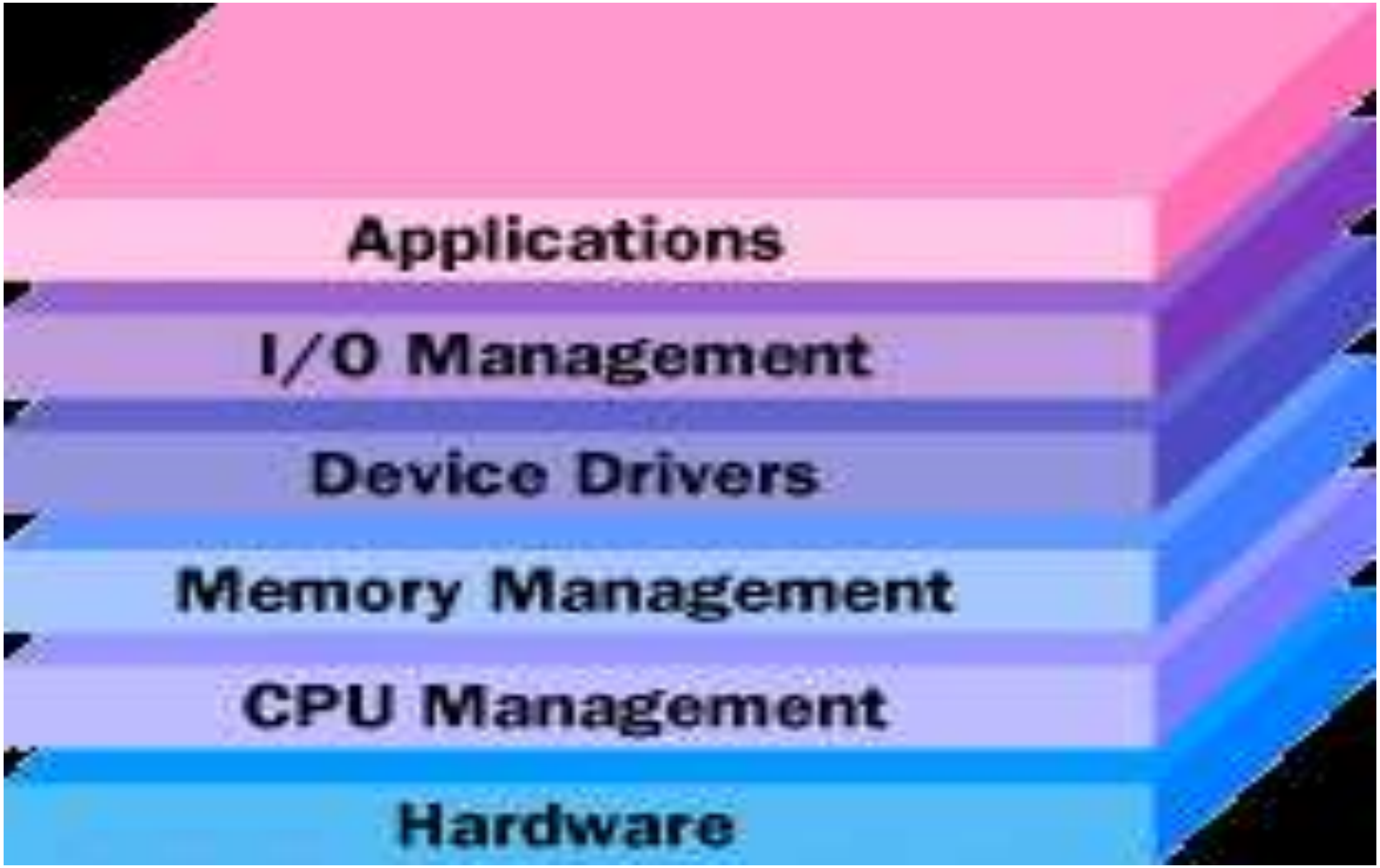
Goals of OS

- The main goals of the Operating System are:
  - To make the computer system convenient to use,
  - To make the use of computer hardware in efficient way/ efficient operation of the computer system .

Operating system objectives (Stallings)

- Convenience
- Efficiency
- Ability to evolve

OS Functions (Stallings)



Early Evolution

- 1945: ENIAC, Moore School of Engineering, University of Pennsylvania.
- 1949: EDSAC and EDVAC
- 1949 BINAC - a successor to the ENIAC
- 1951: UNIVAC by Remington
- 1952: IBM 701
- 1956: The interrupt
- 1954-1957: FORTRAN was developed

Operating Systems by the late 1950s

- By the late 1950s Operating systems were well improved and started supporting following usages :

- It was able to Single stream batch processing
- It could use Common, standardized, input/output routines for device access
- Program transition capabilities to reduce the overhead of starting a new job was added
- Error recovery to clean up after a job terminated abnormally was added.
- Job control languages that allowed users to specify the job definition and resource requirements were made possible

Operating Systems In 1960s

- 1961: The dawn of minicomputers
- 1962 Compatible Time-Sharing System (CTSS) from MIT
- 1963 Burroughs Master Control Program (MCP) for the B5000 system
- 1964: IBM System/360
- 1960s: Disks become mainstream
- 1966: Minicomputers get cheaper, more powerful, and really useful
- 1967-1968: The mouse
- 1964 and onward: Multics
- 1969: The UNIX Time-Sharing System from Bell Telephone Laboratories

Supported OS Features by 1970s

- Multi User and Multi tasking was introduced.
- Dynamic address translation hardware and Virtual machines came into picture.
- Modular architectures came into existence.
- Personal, interactive systems came into existence.

Accomplishments after 1970

- 1971: Intel announces the microprocessor
- 1972: IBM comes out with VM: the Virtual Machine Operating System
- 1973: UNIX 4th Edition is published
- 1973: Ethernet
- 1974 The Personal Computer Age begins
- 1974: Gates and Allen wrote BASIC for the Altair
- 1976: Apple II
- August 12, 1981: IBM introduces the IBM PC
- 1983 Microsoft begins work on MS-Windows
- 1984 Apple Macintosh comes out
- 1990 Microsoft Windows 3.0 comes out
- 1991 GNU/Linux
- 1992 The first Windows virus comes out
- 1993 Windows NT
- 2007: iOS
- 2008: Android OS

- External links to page 13

Terms

- **Multicore processor**
  - A multi-core processor is an integrated circuit to which two or more processors have been attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks.
  - A multi-core processor is a single computing component with two or more independent actual processing units (called "cores"), which are the units that read and execute program instructions.

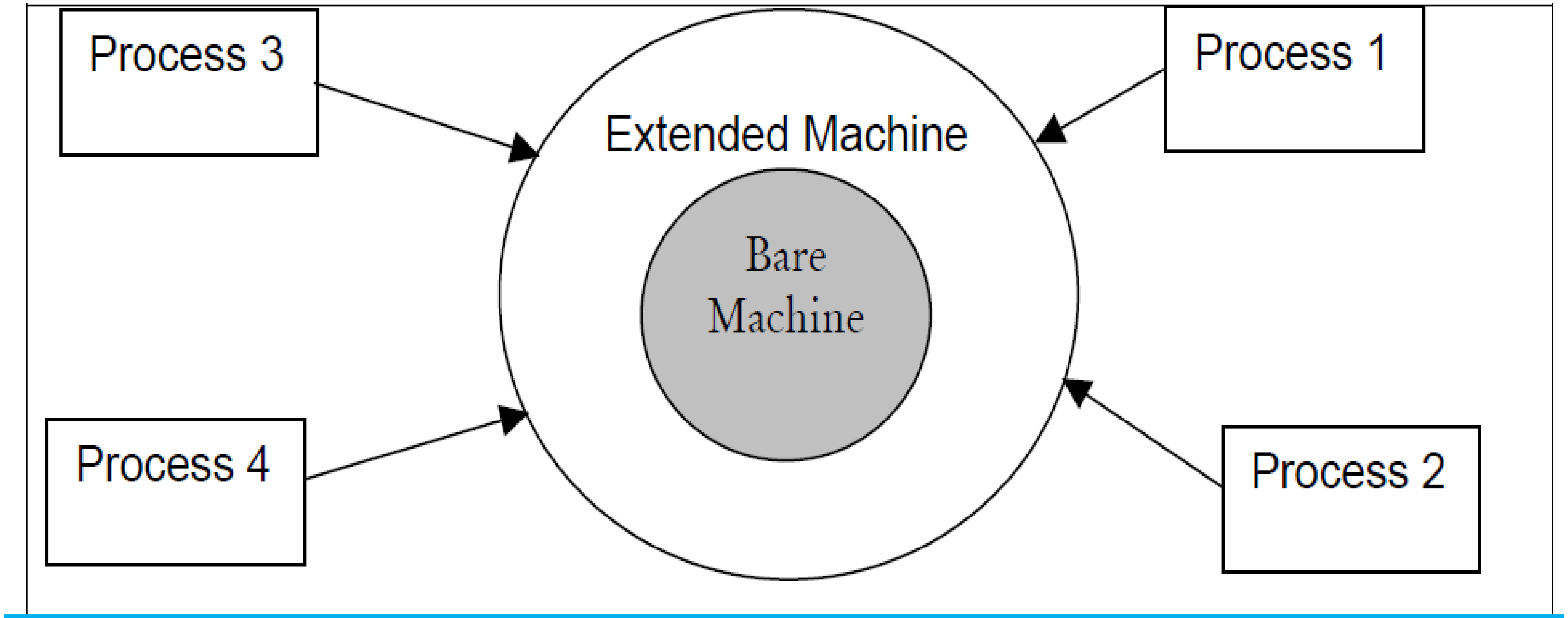
OS Services

- The range and extent of services provided by an Operating System depends on a number of factors.
  - The target environment that the Operating System is intended to support.
    - It is largely determined by user- visible functions of an operating system.

OS Services Cont..

- An Operating System provides services to both the users and to the programs:
  - Providing environment to the programs for their execution
  - Users can execute their programs in an efficient way
- For eg: Program execution, I/O operations, File System manipulation, Communication, Error Detection, Resource Allocation, Protection

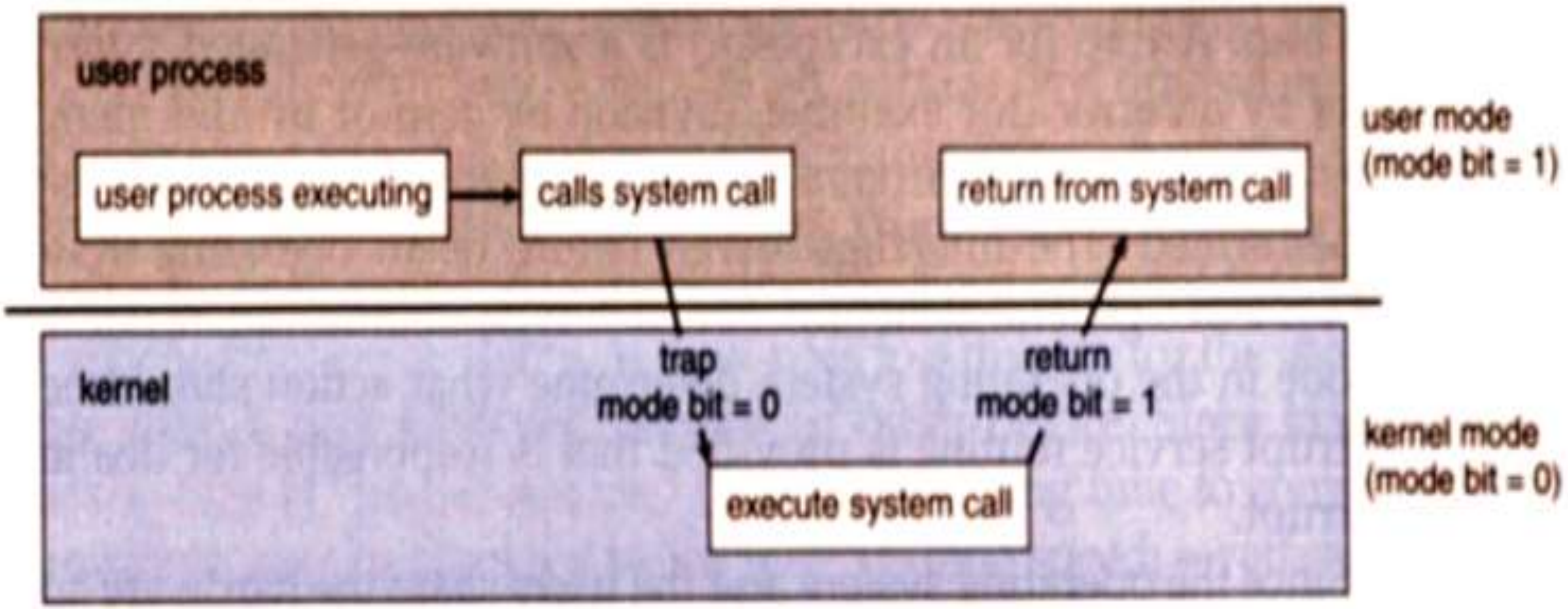
System Call



Operating System extended view

System Call

(Please refer to this link: http://faculty.salina.k-state.edu/tim/ossg/Introduction/sys_calls.html)



System Call Cont..

- Application developers often do not have direct access to the system calls, but can access them through an application programming interface (API).
- The functions that are included in the API invoke the actual system calls.
- By using the API, certain benefits can be gained:
  - Portability: as long a system supports an API, any program using that API can compile and run.
  - Ease of Use: using the API can be significantly easier than using the actual system call.

  - (Please refer to this link: http://faculty.salina.k-state.edu/tim/ossg/Introduction/sys_calls.html)

System Call Parameters

(Please refer to this link: http://faculty.salina.k-state.edu/tim/ossg/Introduction/sys_calls.html)

Types of System Calls