CSE3241: Operating System and System Programming

Sangeeta Biswas

Assistant Professor
Dept. of Computer Science and Engineering
Faculty of Engineering
University of Rajshahi
Rajshahi-6205, Bangladesh
E-mail: sangeeta.cse@ru.ac.bd / sangeeta.cse.ru@gmail.com

October 31, 2017

At a Glance

1. Operating System (OS) is a system software which-
   - manages the computer resources (harware, software) and
   - provides an environment where application software can run in order to fullfill users' demands.

2. In this course, students will-
   - learn very basic things of OS.
   - be familiar with the Command Line Interface (CLI) of Linux kernel based OS.

3. Recommended Books: [1] and [2].

Summary of Syllabus

- Overview
  - Introduction
  - OS Structure
- Process Management
  - Process Concept
  - Threads
  - CPU Scheduling
- Process Coordination
  - Synchronization
  - Deadlocks
- Memory Management
  - Memory-Management Strategy
  - Virtual Memory
- Storage Management
  - File System
  - Disk Management
  - I/O Systems
- Protection and Security
  - System Protection
  - System Security

Syllabus I

- Overview
  - ⊛ Introduction
    - ∗ What is OS?
    - ∗ Tasks of OS
    - ∗ Types of OS
    - ∗ Kernel
  - ⊛ OS Structure
    - ∗ User-OS Interface
    - ∗ System Software Vs. Application Software
    - ∗ OS Vs. System Software
    - ∗ System Call
    - ∗ Different Types of System Calls
    - ∗ Virtual Machine
- Process Management
  - ⊛ Process Concept
    - ∗ What is Process?
    - ∗ Operations on Process
    - ∗ Interprocess Communication
  - ⊛ Threads
    - ∗ Overview

Syllabus II

- Benefits of Threads
- User and Kernel Threads

⊛ CPU Scheduling
- Process Scheduling
- Scheduling Criteria
- Scheduling Algorithms
- Algorithm Evaluation
- Multi-Processor Scheduling

• Process Coordination

⊛ Synchronization
- Background
- Critical Region
- Critical Section Problems
- Synchronization Hardware
- Classical Problems of Synchronization
- Semaphores

⊛ Deadlocks
- What is Deadlock?
- Deadlock Characterization

Syllabus III

- Methods for Handling Deadlocks
- Deadlock Prevention
- Deadlock Avoidance
- Deadlock Detection
- Recovery from Deadlock

- Memory Management
  - Memory-Management Strategy
    - Logical Vs. Physical Address Space
    - Swapping
    - Contiguous Memory Allocation
    - Paging
    - Segmentation
    - Segmentation with Paging
  - Virtual Memory
    - Demand Paging
    - Page Replacement
    - Page Replacement Algorithm
    - Allocation of Frames
    - Trashing

Syllabus IV

- Storage Management
  - ⊛ File System
    - ∗ File Concept
    - ∗ Access Methods
    - ∗ Directory Structure
    - ∗ File System Structure
    - ∗ Allocation Methods
    - ∗ Free-Space Management
    - ∗ Directory Implementation
    - ∗ Efficiency and Performance
  - ⊛ Disk Management
    - ∗ Disk Reliability
    - ∗ Disk Formatting
    - ∗ Boot Block
    - ∗ Bad Blocks
    - ∗ Swap-Space Management
  - ⊛ I/O Systems
    - ∗ I/O Hardware
    - ∗ Polling

Syllabus V

- Interrupts
- DMA
- Application I/O Interface
- Kernel I/O Subsystem
- Performance

- Protection and Security
  - System Protection
    - Goals of Protection
    - Domain of Protection
    - Access Matrix
    - Access Control
  - System Security
    - Security Problem
    - One Time Password
    - Program Threats
    - System Threats
    - User Authentication
    - Threat Monitoring
    - Encryption

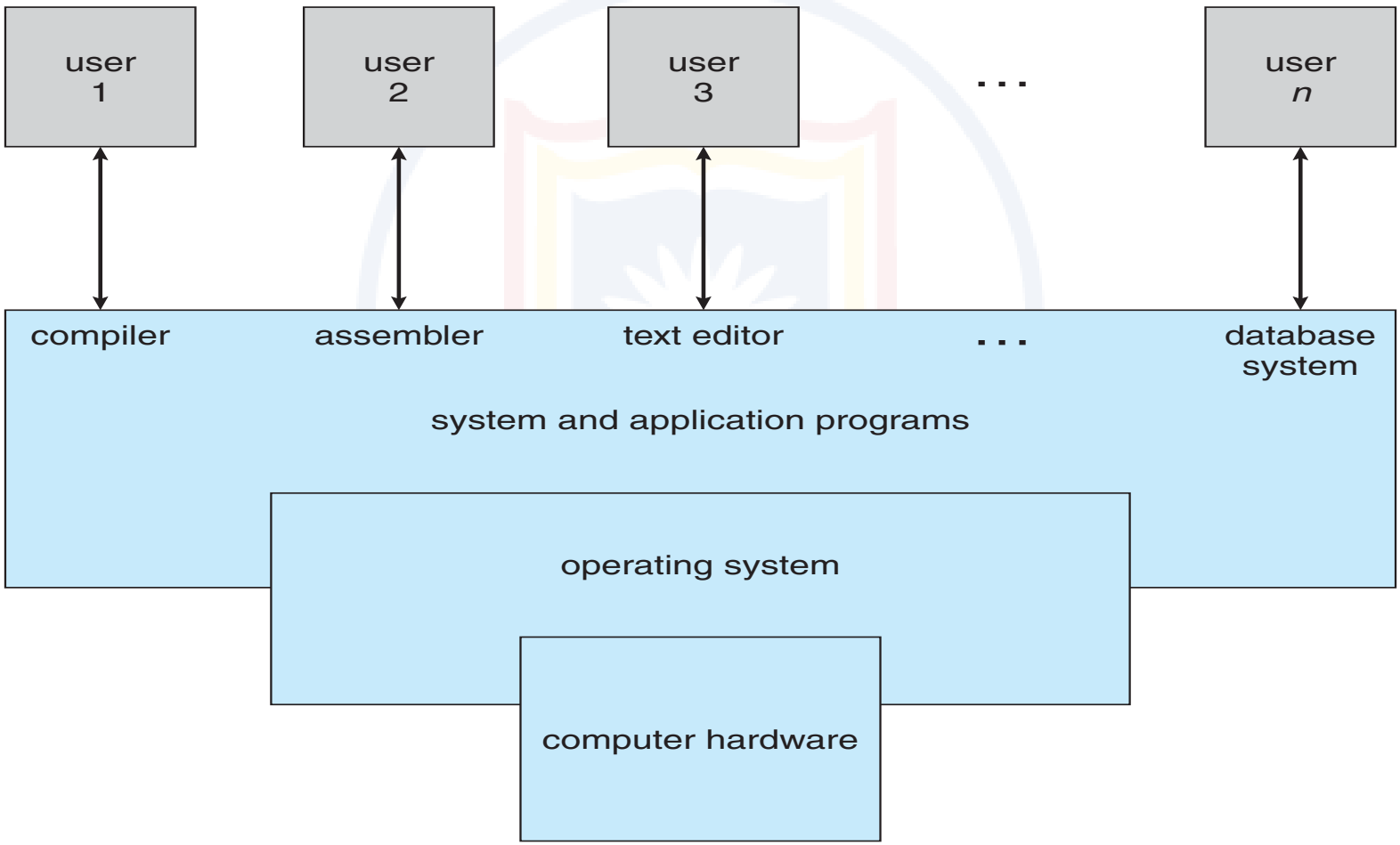
What is OS?

Operating system is a **software** which acts as a:

- **Bridge**:
  - Establishes links between hardware and software (application and system software).
- **Coordinator**:
  - Coordinates all the activities among hardware devices.
- **Abstractor**:
  - Hides details of complicated working procedure of hardware devices from user program.
  - Provides interfaces through which user programs can access hardware for doing their jobs.
- **Controller**:
  - Controls execution of programs to prevent errors and improper use of the computer.
- **Resource Allocator**:
  - Manages all resources (both software and hardware).
  - Decides between conflicting requests for efficient and fair resource use.

Components of a Computer System

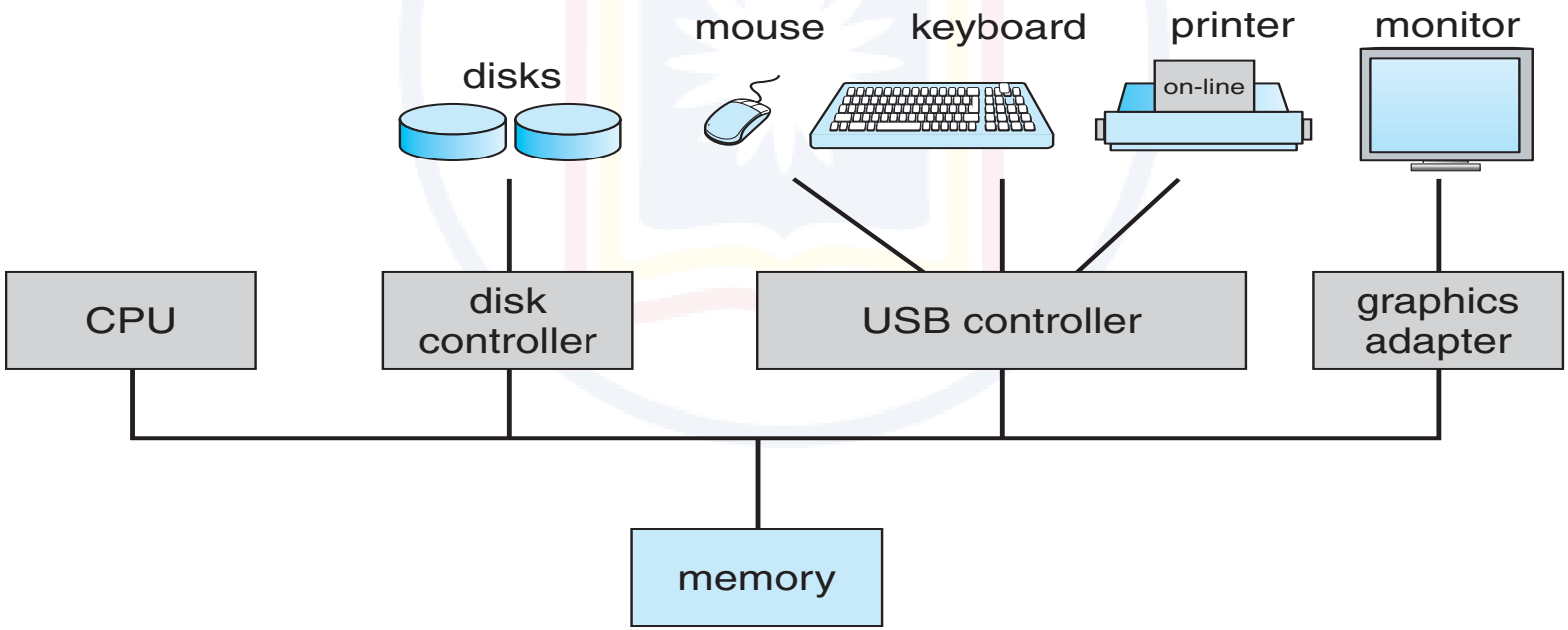
Figure: this and most of other figures are taken from [1]



Device Controller

Each hardware device is connected with a device controller.

- A device controller is the hardware (and/or firmware) that controls the operation of the device.
- It maintains a set of device registers:
  - command registers (write-only)
  - status registers (read-only)
  - data registers (read/write)



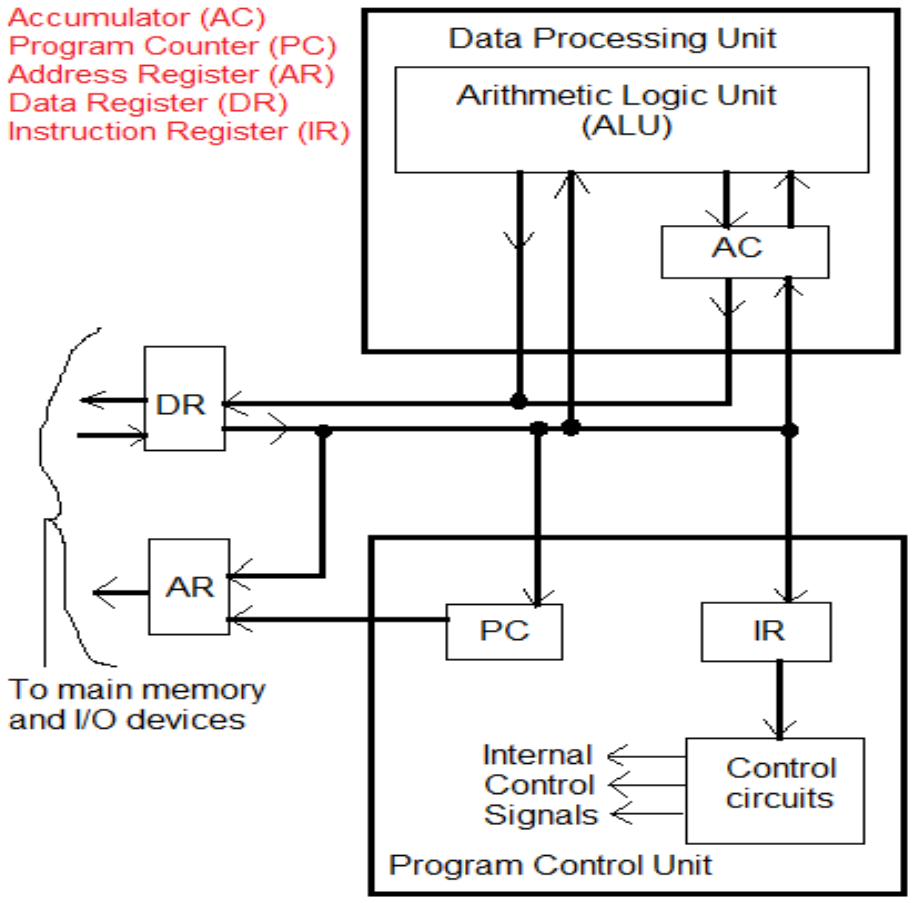
Device Driver

- Device drivers manages interaction with the device controllers.
- Device drivers are part of OS, but not necessarily part of the OS kernel.
- OS provides a simplified view of the device to user applications (e.g., character devices vs. block devices in UNIX).
- In some OS (e.g., Linux), devices are also accessible through the /dev file system.

CPU Registers

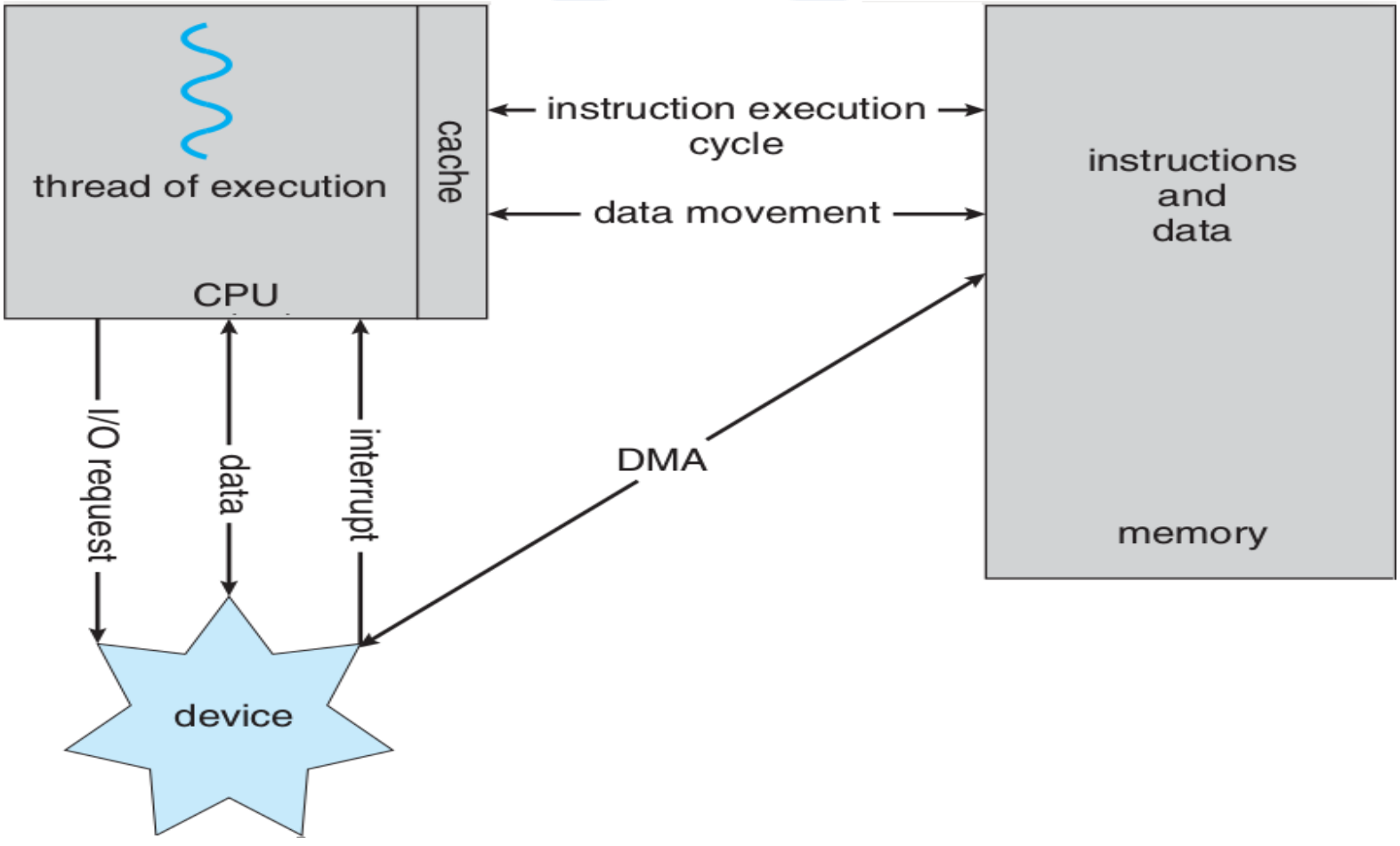
- PC initialization.
- PC $\longrightarrow$ AR
- M(AR) $\longrightarrow$ DR
- DR[opcode] $\longrightarrow$ IR and DR[operand] $\longrightarrow$ AC
- Increment PC
- Decode Instruction
- Excute Instruction

Diagram of Process State [1]



How a Computer System Works

- When an interrupt is fired, the CPU jumps to a predefined position in the kernel's address space and executes an interrupt handler.



Tasks of OS I

The main tasks of an OS are:

- Process Management
- Memory Management
- Storage Management
  - File Managemnet
  - Mass-Storage Management
  - Cache Management
  - I/O Management
- Security and Security Handling

Tasks of OS II

- Process Management
  - Scheduling process and threads.
  - Creating and deleting both user and system pocesses.
  - Suspending and resuming processes.
  - Providing mechanisms for process synchronization.
  - Providing mechanisms for process communication.

- Memory Management
  - Keeping track of which parts of memory are currently being used and by whom.
  - Deciding which processes and data to move into and out of memory.
  - Allocating and deallocating memory space as needed.

Tasks of OS III

- File System Management
  - Creating and deleting files.
  - Organizing files.
  - Backing up files on stable (nonvolatile) storage media.
- Mass-Storage Management
  - Free-space management.
  - Storage allocation.
  - Disk Schedulling.
- I/O Management
  - Handling buffering, caching and spooling.
  - Interacting with device controllers via device drivers.

What is Process?

A process is a program in execution. It is associated with an *address space*.

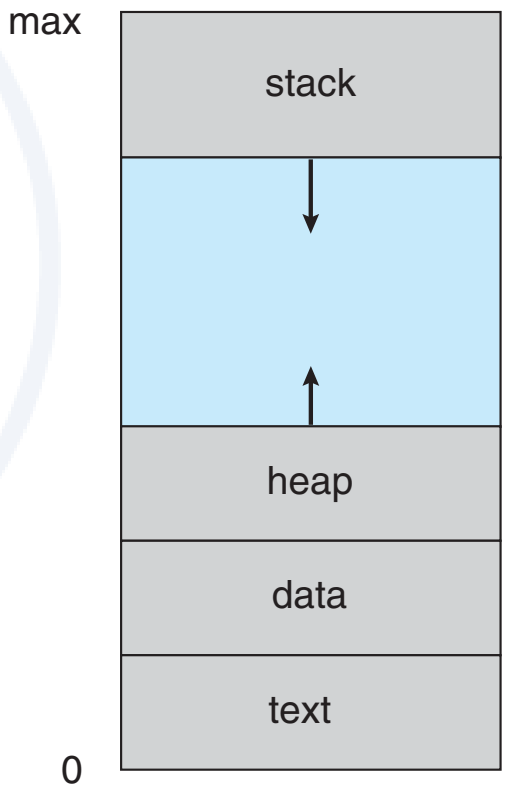
Address Space Contains:

- **Stack**: temporary data such as function parameters, return addresses, local variables.
- **Heap**: dynamically allocated memory locations.
- **Data Section**: global variables.
- **Text Section**: executable program.

Multiple processes associated with the same program:

- have equivalent text sections.
- different data, heap and stack sections.

Process in memory [1]

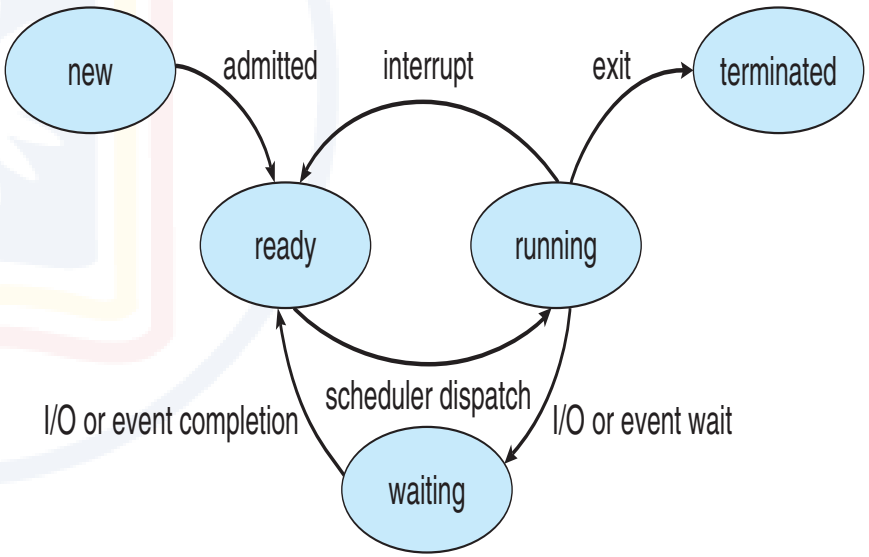


Process State

As a process (P) executes, it changes state. P may be at one of the following 5 states in a system:

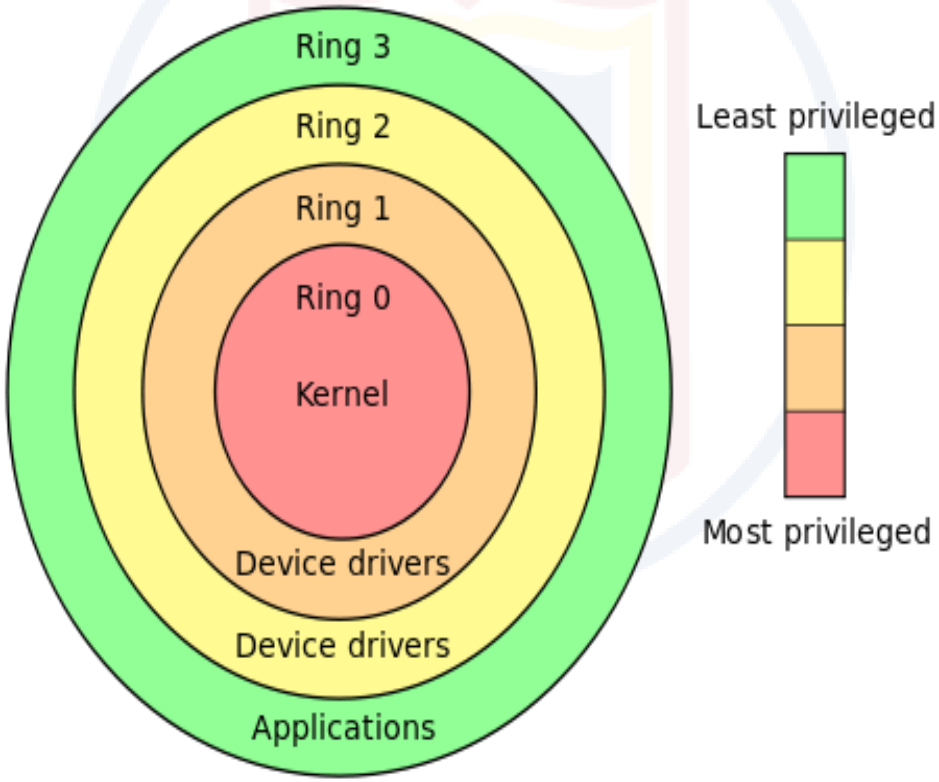
- **New**: P is being created.
- **Running**: P's instructions are being executed.
- **Waiting**: P is waiting for some event to occur.
- **Ready**: P is waiting to be assigned to a processor
- **Terminated**: P has finished execution.

Diagram of Process State [1]



Protection Ring

- OS provides different levels of access to resources.
- A protection ring is one of two or more hierarchical levels or layers of privilege within the architecture of a computer system.



User Mode Vs. Kernel Mode

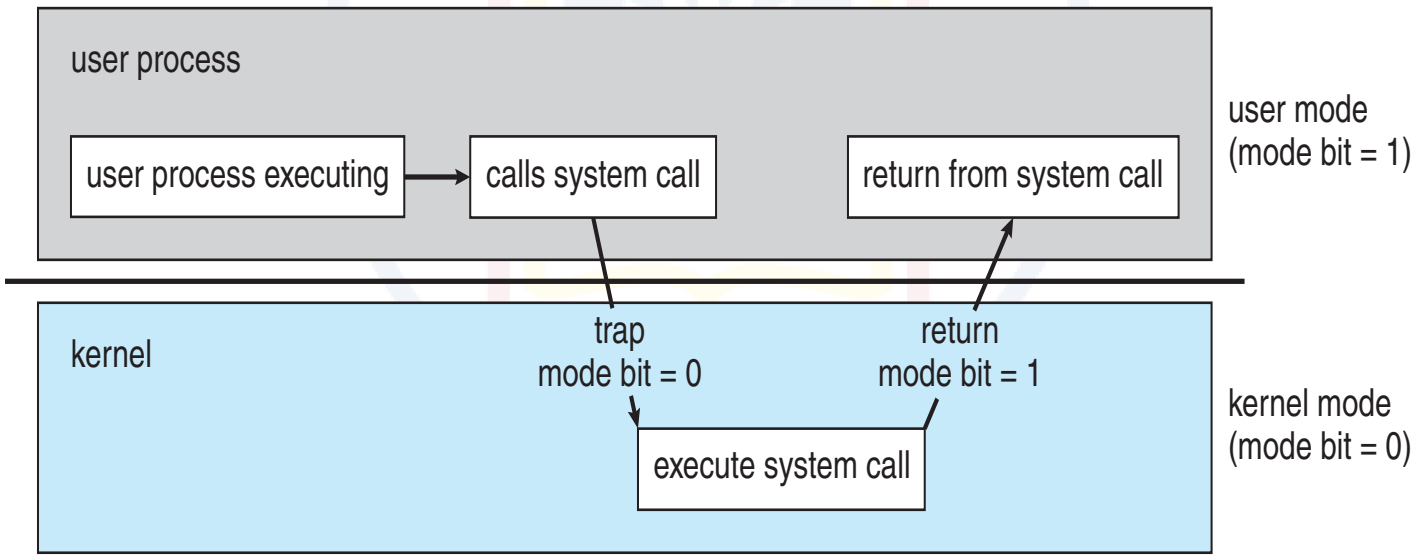
A process can run in different CPU modes, two of them are:

- **Kernel Mode:**
  - It is also known as *supervisor mode*, *system mode* or *privileged mode*.
  - In this mode, a process has complete and unrestricted access to the underlying hardware.
  - In this mode, A process can execute any CPU instruction and reference any memory address.
  - Any process running in this mode is part of the OS, for sure.
  - No process belongs to users' application software can run in this mode.
- **User Mode:**
  - It is also known as *restricted mode*.
  - In this mode, no process can directly access hardware or reference memory.
  - Processes running in this mode must take help from OS process to access hardware.
  - All processes belong to application software run in this mode.

Transition from User to Kernel Mode

- **Mode bit**: is set to the harware to indicate the current mode: kernel (0) or user (1).

- When a user application process needs to access hardware, it requests service from OS via system call.



Segment Register

- In Intel microprocessor, there are 6 segment registers named: cs, ss, ds, es, fs, gs.

- **cs** register points to a segment containing program instructions.

- **cs** register includes a 2-bit field that specifies the Current Privilege Level (CPL) of the CPU.
  - The value 0 denotes the highest privilege level, while the value 3 denotes the lowest one.
  - Linux uses only levels 0 and 3, which are respectively called Kernel Mode and User Mode.

Command: top

'top' can display:

- system summary information.
- a list of processes or threads currently being managed by kernel.

```
sangeeta@sangeeta-Aspire-one-1-131:~$ top

top - 22:49:11 up 14:23,  1 user,  load average: 2.88, 2.56, 1.68
Tasks: 221 total,   3 running, 218 sleeping,   0 stopped,   0 zombie
%Cpu(s): 33.6 us, 36.2 sy,  0.0 ni, 25.4 id,  4.1 wa,  0.0 hi,  0.7 si,  0.0 st
KiB Mem :  1962888 total,   191128 free,  1240904 used,   530856 buff/cache
KiB Swap:  2011132 total,  1391588 free,   619544 used.   284676 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1705 sangeeta  20   0  678876  37516  18784 R  60.9  1.9   7:11.12 gnome-terminal-
 8186 sangeeta  20   0    4356    636    560 R  27.2  0.0   3:00.56 infinteLoop.o
 1277 sangeeta  20   0 1474484  72216  22372 S  11.6  3.7  17:33.30 compiz
 8208 root      20   0       0      0      0 S  10.6  0.0   0:42.83 kworker/u8:4
  886 root      20   0  452468  70160  37576 S   7.0  3.6  16:54.52 Xorg
 8092 root      20   0       0      0      0 S   7.0  0.0   0:40.80 kworker/u8:2
 8121 root      20   0       0      0      0 S   7.0  0.0   0:49.50 kworker/u8:1
 2390 sangeeta  20   0 2723132 537352  57248 S   3.3 27.4 105:40.80 Web Content
 4671 sangeeta  20   0  688476  63288  13308 S   2.0  3.2   8:03.33 skype
 1159 sangeeta  20   0  347720   8208   2460 S   1.3  0.4   1:35.91 ibus-daemon
 8373 root      20   0       0      0      0 S   1.3  0.0   0:09.14 kworker/u8:0
 8412 sangeeta  20   0   41800   3680   3060 R   0.7  0.2   0:00.14 top
  163 root      20   0       0      0      0 S   0.3  0.0   0:32.30 mmcqd/0
 7252 sangeeta  20   0 1343392  62108  18268 S   0.3  3.2   0:21.41 soffice.bin
 8311 sangeeta  20   0  574004  21872  14456 S   0.3  1.1   0:00.39 unity-scope-loa
```

Fields of 'top'

- 1st Row: program name; current time; up time; total number of users; system load avg over the last 1, 5 and 15 minutes.
- 2nd Row: total tasks or threads at different states.
- 3rd Row: CPU usage for {un-niced user processes (us), kernel processes (sy), niced user processes (ni), idle handler (id), I/O wait (wa), hardware interrupts (hi), software interrupts (si), stolen (st)}.
- 4th Row: memory usage.
- In Columns:
  - PID – Process Id
  - USER – User Name
  - PR – Priority
  - NI – Nice Value
  - {VIRT, RES, SHR} – {Virtual, Resident, Shared} Memory Size (KiB).
  - S – Process Status
  - {%CPU, %MEM} – CPU, Memory} Usage
  - TIME – CPU Time
  - COMMAND – Command Name or Command Line

Command: pstree

'pstree' shows running processes as a tree.

- The tree is rooted at either 'systemd' or 'init'.
- 'pstree' visually merges identical branches by putting them in square brackets and prefixing them with the repetition count.
- Child threads of a process are found under the parent process and are shown with the process name in curly braces.

```
sangeeta@sangeeta-Aspire-one-1-131:~$ pstree
systemd─┬─ModemManager─┬─{gdbus}
        │              └─{gmain}
        ├─NetworkManager─┬─dhclient
        │                ├─dnsmasq
        │                ├─{gdbus}
        │                └─{gmain}
        ├─accounts-daemon─┬─{gdbus}
        │                 └─{gmain}
        ├─acpid
        ├─agetty
        ├─auditd───{auditd}
        ├─avahi-daemon───avahi-daemon
        ├─bluetoothd
        ├─colord─┬─{gdbus}
        │        └─{gmain}
        ├─cron
        ├─cups-browsed─┬─{gdbus}
        │              └─{gmain}
        ├─cupsd
        ├─dbus-daemon
        ├─fwupd─┬─3*[{GUsbEventThread}]
        │       ├─{fwupd}
        │       └─{gdbus}
```

Variation of 'pstree'

- Show PIDs with processes' names.
  $ pstree -p
- Show parent processes of the specified process.
  $ pstree -s <PID>

```
sangeeta@sangeeta-Aspire-one-1-131:~$ pstree -p
systemd(1)─┬─ModemManager(685)─┬─{gdbus}(744)
           │                   └─{gmain}(724)
           ├─NetworkManager(708)─┬─dhclient(15873)
```

```
sangeeta@sangeeta-Aspire-one-1-131:~$ pstree -s 4671
systemd───lightdm───lightdm───upstart───skype─┬─{QFileInfoGather}
                                              ├─{QInotifyFileSys}
                                              ├─2*[{QThread}]
                                              ├─18*[{skype}]
                                              └─{threaded-ml}
```

References I

P. B. Galvin A. Silbeschatz and G. Gagne.
Operating System Concepts.
John Wiley & Sons, 9 edition, 2012.

A. S. Tanenbaum and A. S. Woodhull.
Operating Systems– Design and Implementation.
Pearson Prentice Hall, 3 edition, 2006.