

## **Operating Systems**

Narasimhulu M<sub>M. Tech.</sub> Assistant Professor Department of Computer Science & Engineering

SRU						
S.No.	Course Outcomes	Cognitive Level				
1	Illustrate various types of system calls and find the stages of various process states.	Understand				
2	Implement thread scheduling and process scheduling techniques	Apply				
3	Distinguish among IPC synchronization Techniques	Understand				
4	Implement page replacement algorithms, memory management techniques and deadlock issues.	Apply				
5	Make use of the file systems for applying different allocation and access techniques.	Understand				
6	Illustrate system protection and Security.	Understand				



## Unit-1 Syllabus

**Introduction:** Operating System Operations, Resource Management, Security and Protection, Virtualization, Distributed Systems, Computing Environments.

**Operating-System Structures:** Operating-System Services, User and Operating-System Interface, System Calls, System Services.



## Unit I – Introduction to Operating systems



## **Chapter 1**Introduction

Narasimhulu  $M_{\text{M. Tech.}}$ Assistant Professor Department of Computer Science & Engineering



## Operating System Operations



#### **Operating-System Operations**

- Bootstrap program simple code to initialize the system, load the kernel
- Kernel loads
- Starts system daemons (services provided outside of the kernel)
- Kernel interrupt driven (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (exception or trap):
    - Software error (e.g., division by zero)
    - Request for operating system service system call
    - Other process problems include infinite loop, processes modifying each other or the operating system



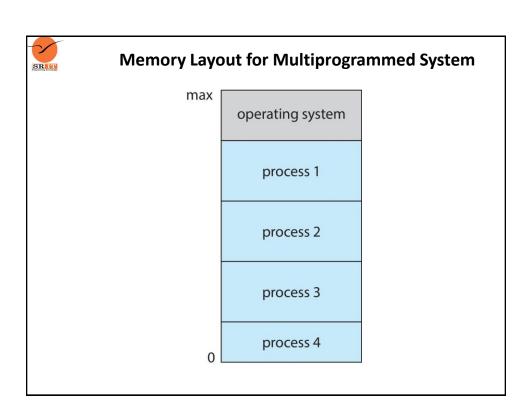
#### Multiprogramming (Batch system)

- Single user cannot always keep CPU and I/O devices busy
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via job scheduling
- When job has to wait (for I/O for example), OS switches to another job



#### Multitasking (Timesharing)

- A logical extension of Batch systems— the CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing
  - Response time should be < 1 second
  - Each user has at least one program executing in memory, which is called process
  - If several jobs ready to run at the same time ⇒ CPU scheduling
  - If processes don't fit in memory, swapping moves them in and out to run
  - Virtual memory allows execution of processes not completely in memory





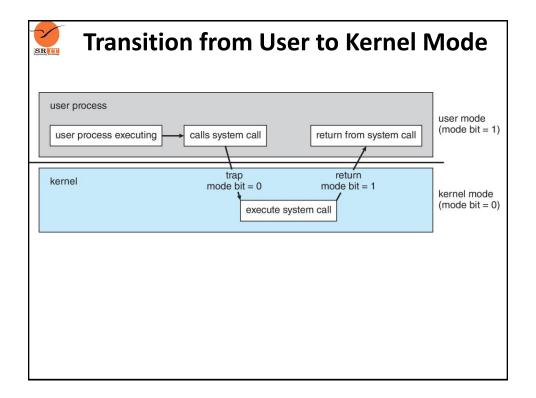
#### **Dual-mode Operation**

- Dual-mode operation allows OS to protect itself and other system components
  - User mode and kernel mode
- Mode bit provided by hardware
  - Provides ability to distinguish when system is running user code or kernel code.
  - When a user is running → mode bit is "user"
  - When kernel code is executing → mode bit is "kernel"
- Some instructions designated as privileged, only executable in kernel mode



#### **Dual-mode Operation (Cont.)**

- How do we guarantee that user does not explicitly set the mode bit to "kernel"?
- When the system starts executing it is in kernel mode
- When control is given to a user program the mode-bit changes to "user mode".
- When a user issues a system call it results in an interrupt, which trap to the operating system. At that time, the mode—bit is set to "kernel mode".





#### **Timer**

- Timer to prevent infinite loop (or process hogging resources)
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock
  - Operating system set the counter (privileged instruction)
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



## Resource Management

Narasimhulu  $M_{M. Tech.}$ Assistant Professor Department of Computer Science & Engineering



#### **Process Management**

- A process is a program in execution. It is a unit of work within the system. Program is a passive entity; process is an active entity.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one program counter specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter



#### **Process Management Activities**

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling



#### Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed



#### File-system Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit file
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - · Creating and deleting files and directories
    - · Primitives to manipulate files and directories
    - · Mapping files onto secondary storage
    - · Backup files onto stable (non-volatile) storage media



#### Mass-Storage Management

- Usually, disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
  - Partitioning
  - Protection



### Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy



#### Characteristics of Various Types of Storage

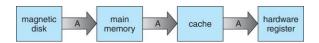
Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit



#### Migration of data "A" from Disk to Register

 Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache
- Distributed environment situation even more complex
  - Several copies of a datum can exist
  - Various solutions covered in Chapter 19



### I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices



## Security and Protection

Narasimhulu  $M_{\text{M. Tech.}}$ Assistant Professor Department of Computer Science & Engineering



#### **Protection and Security**

- Protection mechanism for controlling access of processes or users to resources defined by the OS
- **Security** defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service



#### **Protection**

- Systems generally first distinguish among users, to determine who can do what
  - User identities (user IDs, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file
  - Privilege escalation allows user to change to effective ID with more rights



#### **Virtualization**



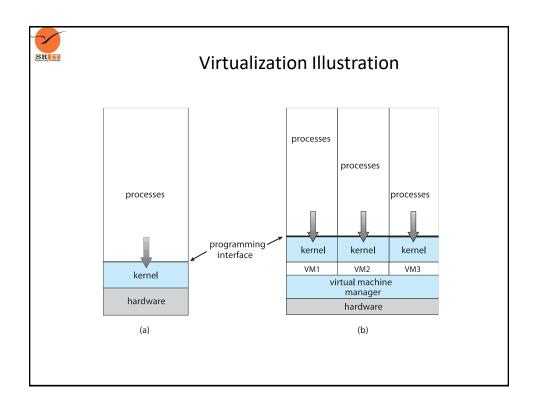
#### Virtualization

- Allows operating systems to run applications within other OSes
  - Vast and growing industry
- Emulation used when source CPU type different from target type (i.e., PowerPC to Intel x86)
  - Generally slowest method
  - When computer language not compiled to native code – Interpretation
- Virtualization OS natively compiled for CPU, running guest OSes also natively



#### Virtualization (cont.)

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - Apple laptop running Mac OS X host, Windows as a guest
  - Developing apps for multiple OSes without having multiple systems
  - Quality assurance testing applications without having multiple systems
  - Executing and managing compute environments within data centers
- VMM can run natively in which case they





## Distributed Systems



## **Distributed Systems**

- Collection of separate, possibly heterogeneous, systems networked together
  - Network is a communications path, TCP/IP most common
    - Local Area Network (LAN)
    - Wide Area Network (WAN)
    - Metropolitan Area Network (MAN)
    - Personal Area Network (PAN)
- Network Operating System provides features between systems across network
  - Communication scheme allows systems to



## Computing Environments



## **Computing Environments**

- Traditional
- Mobile
- Client Server
- Pear-to-Pear
- Cloud computing
- Real-time Embedded



#### **Traditional**

- Stand-alone general-purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- Portals provide web access to internal systems
- Network computers (thin clients) are like Web terminals
- Mobile computers interconnect via wireless networks
- Networking becoming ubiquitous even home systems use firewalls to protect home computers from Internet attacks



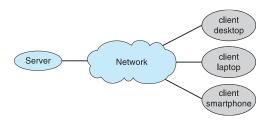
#### **Mobile Computing**

- Handheld smartphones, tablets, etc.
- What is the functional difference between them and a "traditional" laptop?
- Extra feature more OS features (GPS, gyroscope)
- Allows new types of apps like augmented reality
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders are Apple iOS and Google Android



#### **Client Server Computing**

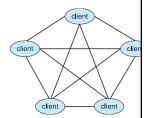
- Dumb terminals supplanted by smart PCs
- Many systems now servers, responding to requests generated by clients
  - Compute-server system provides an interface to client to request services (i.e., database)
  - File-server system provides interface for clients to store and retrieve files





#### **Peer-to-Peer**

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead, all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via discovery protocol
  - Examples include Napster and Gnutella,
    Voice over IP (VoIP) such as Skype





#### **Cloud Computing**

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for its functionality.
  - Amazon EC2 has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage



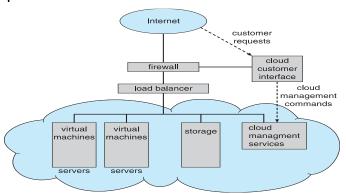
#### **Cloud Computing – Many Types**

- Public cloud available via Internet to anyone willing to pay
- Private cloud run by a company for the company's own use
- Hybrid cloud includes both public and private cloud components
- Software as a Service (SaaS) one or more applications available via the Internet (i.e., word processor)
- Platform as a Service (PaaS) software stack ready for application use via the Internet (i.e., a



#### **Cloud Computing (cont.)**

- Cloud computing environments composed of traditional Os plus cloud management tools
  - Internet connectivity requires security like firewalls
  - Load balancers spread traffic across multiple applications





#### Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
  - Vary considerable, special purpose, limited purpose OS, real-time OS
  - Use expanding
- Many other special computing environments as well
  - Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints



#### Free and Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary closed-source and proprietary
- Counter to the copy protection and Digital Rights Management (DRM) movement
- Started by Free Software Foundation (FSF), which has "copyleft" GNU Public License (GPL)
  - Free software and open-source software are two different ideas championed by different groups of people
    - http://gnu.org/philosophy/open-source-misses-the-point.html/
- Examples include GNU/Linux and BSD



#### The Study of Operating Systems

There has never been a more interesting time to study operating systems, and it has never been easier. The open-source movement has overtaken operating systems, causing many of them to be made available in both source and binary (executable) format. The list of operating systems available in both formats includes Linux, BUSD UNIX, Solaris, and part of macOS. The availability of source code allows us to study operating systems from the inside out. Questions that we could once answer only by looking at documentation or the behavior of an operating system we can now answer by examining the code itself.

Operating systems that are no longer commercially viable have been open-sourced as well, enabling us to study how systems operated in a time of fewer CPU, memory, and storage resources. An extensive but incomplete list of open-source operating-system projects is available from https://curlie.org/Computers/Software/Operating\_Systems/Open\_Source/

In addition, the rise of virtualization as a mainstream (and frequently free) computer function makes it possible to run many operating systems on top of one core system. For example, VMware (http://www.vmware.com) providesa free "player" for Windows on which hundreds of free virtual appliances" can run. Virtualbox (http://www.virtualbox.com) provides a free, open-source virtual machine manager on many operating systems. Using such tools, students can try out hundreds of operating systems without dedicated hardware.

The advent of open-source operating systems has also made it easier to make the move from student to operating-system developer. With some knowledge, some effort, and an Internet connection, a student can even create a new operating-system distribution. Just a few years ago, it was difficult or impossible to get access to source code. Now, such access is limited only by how much interest, time, and disk space a student has.



## Computing Environments



## **END of Chapter - 1**



## **Chapter 2**Operating-System Structures



## Operating-System Services

Narasimhulu  $M_{M. Tech.}$ Assistant Professor Department of Computer Science & Engineering



# User and Operating-System Interface



## System Calls

Narasimhulu  $M_{\text{M. Tech.}}$ Assistant Professor Department of Computer Science & Engineering



## System Services

