Chapter 1

1- Operating system is a resource allocator and control program making efficient use of HW and managing execution of user programs

2- Users of dedicate systems such as workstations have dedicated resources but frequently use shared resources from servers

3- The one program running at all times on the computer" is the kernel, part of the operating system

4- a system program (ships with the operating system, but not part of the kernel)

5- an application program, all programs not associated with the operating system

6- middleware – a set of software frameworks that provide addition services to application developers such as databases, multimedia, graphics

7- One or more CPUs, device controllers connect through common bus providing access to shared memory

8- Each device controller type has an operating system device driver to manage it

9- Device controller informs CPU that it has finished its operation by causing an interrupt

10- Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all the service routines

11- A trap or exception is a software-generated interrupt caused either by an error or a user request

12- An operating system is interrupt driven

13- bootstrap program is loaded at power-up or reboot

14- bootstrap program Typically stored in ROM or EPROM, generally known as firmware

15- System call – request to the OS to allow user to wait for I/O completion

16- Device-status table contains entry for each I/O device indicating its type, address, and state

17- Main memory – only large storage media that the CPU can access directly

18- Secondary storage – extension of main memory that provides large nonvolatile storage capacity

19- Hard Disk Drives (HDD) – rigid metal or glass platters covered with magnetic recording material

20- Disk surface is logically divided into tracks, which are subdivided into sectors

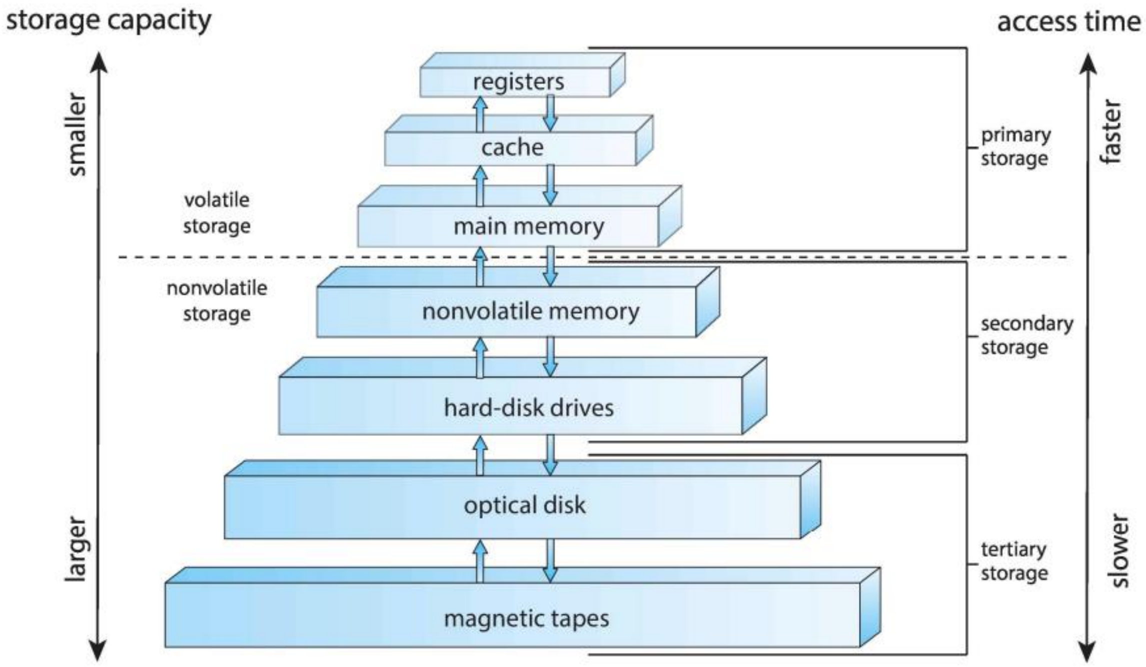
21- The disk controller determines the logical interaction between the device and the computer

22- Non-volatile memory (NVM) devices– faster than hard disks, nonvolatile

23- Storage systems organized in hierarchy Speed, Cost then Volatility

24- Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage

25- Device Driver for each device controller to manage I/O, it provides uniform interface between controller and kernel



26-

27- Multiprocessors systems growing in use and importance, Also known as parallel systems, tightly-coupled systems

28- Multiprocessors Advantages include: 1. Increased throughput 2. Economy of scale 3. Increased reliability – graceful degradation or fault tolerance

29- Asymmetric Multiprocessing – each processor is assigned a specie task

30- Symmetric Multiprocessing – each processor performs all tasks

31- Usually sharing storage via a storage-area network (SAN)

32- Asymmetric clustering has one machine in hot-standby mode

33- Symmetric clustering has multiple nodes running applications, monitoring each other

34- Some clusters are for high-performance computing (HPC), Applications must be written to use parallelization

35- Some have distributed lock manager (DLM) to avoid conflicting operations

36- Multiprogramming (Batch system) needed for efficiency

37- Timesharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing.

38- Response time should be < 1 second

39- Each user has at least one program executing in memory -> process

40- If several jobs ready to run at the same time -> CPU scheduling

41- If processes don't fit in memory, swapping moves them in and out to run

42- Virtual memory allows execution of processes not completely in memory

43- Dual-mode operation allows OS to protect itself and other system components

44- Single-threaded process has one program counter specifying location of next instruction to execute

45- Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache

46- Protection – any mechanism for controlling access of processes or users to resources defined by the OS
47- Security – defense of the system against internal and external attacks
48- Emulation used when source CPU type different from target type (i.e. PowerPC to Intel x86)
49- Virtualization – OS natively compiled for CPU, running guest OSes also natively compiled
50- 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)
51- Network Operating System provides features between systems across network
52- Hash function can create a hash map
53- Bitmap – string of n binary digits representing the status of n items
54- Portals provide web access to internal systems
55- Network computers (thin clients) are like Web terminals
56- Mobile computers interconnect via wireless networks
57- Networking becoming ubiquitous – even home systems use firewalls to protect home computers from Internet attacks
58- Leaders are Apple iOS and Google Android
59- Compute-server system provides an interface to client to request services (i.e., database)
60- File-server system provides interface for clients to store and retrieve files
61- Public cloud – available via Internet to anyone willing to pay
62- Private cloud – run by a company for the company's own use
63- Hybrid cloud – includes both public and private cloud components
64- Software as a Service (SaaS) – one or more applications available via the Internet (i.e., word processor)
65- Platform as a Service (PaaS) – software stack ready for application use via the Internet (i.e., a database server)
66- Infrastructure as a Service (IaaS) – servers or storage available over Internet (i.e., storage available for backup use)

Chapter 4

1- Process creation is heavy-weight (time consuming and resource intensive) while thread creation is light-weight (because threads share the code, data and OS resources)

2- If the web-server process is multithreaded, the server will create a separate thread that listens for client requests

3- When a request is made, rather than creating another process, the server creates a new thread to service the request and resume listening for additional requests.

4- If the web server ran as a traditional single-threaded process, it would be able to service only one client at a time, and a client might have to wait a very long time for its request to be serviced.

5- Kernels are generally multithreaded

6- Processes can only share resources through techniques such as shared memory and message passing.

7- A single-threaded process can run on only one processor, regardless how many are available.

8- Multicore or multiprocessor systems putting pressure on programmers, (to make better use of the multiple computing cores)

9- Parallelism implies a system can perform more than one task simultaneously

10- Concurrency allows more than one task to make progress

11- Data parallelism – distributes subsets of the same data across multiple cores, same operation on each

12- Task parallelism – distributing threads across cores, each thread performing unique operation

13- Three primary thread libraries: POSIX Pthreads, Windows threads and Java threads.

14- Thread library provides programmer with API for creating and managing threads

15- Invoking a function in the API for the library typically results in a system call to the kernel.

16- Asynchronous threading, once the parent creates a child thread, the parent resumes its execution

17- Synchronous threading occurs when the parent thread creates one or more children and then must wait for all of its children to terminate before it resumes

18- POSIX stands for Portable Operating System Interface, is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems.

19- Any data declared globally—that is, declared outside of any function—are shared among all threads belonging to the same process.

20- Because Java has no notion of global data, access to shared data must be explicitly arranged between threads.

21- There are two techniques for creating threads in a Java program. One approach is to create a new class that is derived from the Thread class and to override its run() method. An alternative—and more commonly used— technique is to define a class that implements the Runnable interface.

22- If two or more threads are to share data in a Java program, the sharing occurs by passing references to the shared object to the appropriate threads.

23- parallel regions – blocks of code that can run in parallel

24- if a thread invokes the exec() system call, the program specified in the parameter to exec() will replace the entire process—including all threads.

25- Signals are used in UNIX systems to notify a process that a particular event has occurred.

26- A signal handler is used to process signals.

27- Every signal has default handler that kernel runs when handling signal

28- User-defined signal handler can override default

29- A signal may be received either synchronously or asynchronously

30- Typically, an asynchronous signal is sent to another process.

31- synchronous signal include: illegal memory access and division by 0

32- Thread to be canceled is target thread.

33- Cancellation only occurs when thread reaches cancellation point

34- Thread-local storage (TLS) allows each thread to have its own copy of data

35- Lightweight process (LWP) Typically use an intermediate data structure between user and kernel threads

36- upcalls - a communication mechanism from the kernel to the upcall handler in the thread library

37- The register set, stacks, and private storage area are known as the context of the thread

Chapter 6

1- Each process has critical section segment of code

2- Each process must ask permission to enter critical section in entry section, may follow critical section with exit section, then remainder section

3- Mutual Exclusion - If process Pi is executing in its critical section, then no other processes can be executing in their critical sections

4- Progress - If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely

5- Bounded Waiting - A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted

6- Preemptive – allows preemption of process when running in kernel mode

7- Non-preemptive – runs until exits kernel mode, blocks, or voluntarily yields CPU

8- Mutex Locks requires busy waiting. This lock called spinlock

9- Counting semaphore – integer value can range over an unrestricted domain

10- Binary semaphore and mutex lock – integer value can range only between 0 and 1

11- block – place the process invoking the operation on the appropriate waiting queue

12- wakeup – remove one of processes in the waiting queue and place it in the ready queue

13- Deadlock – two or more processes are waiting indefinitely for an event that can be caused by only one of the waiting processes

Chapter 9

1- Main memory can take many cycles, causing a stall

2- Cache sits between main memory and CPU registers

3- We can provide protection by using a pair of base and limit registers define the logical address space of a process

4- Compile time: If memory location known a priori, absolute code can be generated; must recompile code if starting location changes

5- Load time: Must generate relocatable code if memory location is not known at compile time

6- The concept of a logical address space that is bound to a separate physical address space is central to proper memory management

7- Logical address – generated by the CPU; also referred to as virtual address

8- Physical address – address seen by the memory unit

9- Logical address space is the set of all logical addresses generated by a program

10- Physical address space is the set of all physical addresses generated by a program

11- Dynamically linked libraries are system libraries (such as language subroutine libraries) that are linked to user programs when the programs are run.

12- Static linking – system libraries and program code combined by the loader into the binary program image

13- Small piece of code, stub, used to locate the appropriate memory-resident library routine

14- System also known as shared libraries

15- A library may be replaced by a new version, and all programs that reference the library will automatically use the new version.

16- Main memory usually into two partitions, OS and user

17- Hole – block of available memory; holes of various size are scattered throughout memory

18- First-fit: Allocate the first hole that is big enough

19- Best-fit: Allocate the smallest hole that is big enough; must search entire list, unless ordered by size

20- Worst-fit: Allocate the largest hole; must also search entire list

21- First-fit and best-fit better than worst-fit in terms of speed and storage utilization

22- External Fragmentation – total memory space exists to satisfy a request, but it is not contiguous

23- Internal Fragmentation – allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used

24- Reduce external fragmentation by compaction

25- Divide physical memory into fixed-sized blocks called frames

26- Divide logical memory into blocks of same size called pages

27- Set up a page table to translate logical to physical addresses

28- Page number (p) – used as an index into a page table which contains base address of each page in physical memory

29- Page offset (d) – combined with base address to define the physical memory address that is sent to the memory unit

30- Page-table base register (PTBR) points to the page table

31- Page-table length register (PTLR) indicates size of the page table

32- The two memory access problem can be solved by the use of a special fast-lookup hardware cache called translation look-aside buffers (TLBs) (also called associative memory)

33- Some TLBs store address-space identifiers (ASIDs) in each TLB entry – uniquely identifies each process to provide address-space protection for that process

34- Some entries can be wired down for permanent fast access

35- Valid-invalid bit attached to each entry in the page table: "valid" indicates that the associated page is in the process ' logical address space, and is thus a legal page "invalid" indicates that the page is not in the process' logical address space

36- Variation for 64-bit addresses is clustered page tables

37- A process can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution

38- Backing store – fast disk large enough to accommodate copies of all memory images for all users; must provide direct access to these memory images

39- Roll out, roll in – swapping variant used for priority-based scheduling algorithms; lower-priority process is swapped out so higher-priority process can be loaded and executed

40- System maintains a ready queue of ready-to-run processes which have memory images on disk

41- Android terminates apps if low free memory, but first writes application state to flash for fast restart

42- 32-bit address limits led Intel to create page address extension (PAE), allowing 32-bit apps access to more than 4GB of memory space, Top two bits refer to a page directory pointer table

43-