SCHEME OF VALUATION

(Scoring indicators)

Revision:2015

Course code: 4134

Course title: OPERATING SYSTEM

Qst. No.	Scoring indicators	Split up score	Sub Total	Total
I (1)	**System software**: software on a computer that is designed to control and work with computer hardware. Operating system is an example.	2	2	2
I (2)	DEADLOCK: • A process requests a resource and if the resources are not available at that time, the process enters a waiting state. • This waiting will not end.	2	2	2
I (3)	• **Compile Time:-** know at compile time where the process will reside in memory, • **Load Time:-** known at compile time where the process will reside in memory then the compiler must generate re-locatable code. • **Execution Time:-**If the process can be moved during its execution , from one memory segment to another , then binding is delayed until run time.	2*1	2	2
I (4)	**Process State:** **New:** The process is being created. **Running:** Instructions are being executed. **Waiting:** The process is waiting for some event to occur (such as an I/O completion or reception of a signal). **Ready:** The process is waiting to be assigned to a processor. **Terminated:** The process has finished execution.	2	2	2
I (5)	**Tree directory Structure:** **(2 marks)** The tree has a root directory, and every file in the system has a unique path name.	2	2	2
II(1)	**Multiprogramming** • One or more programs loaded in main memory which are ready to execute. • Only one program at a time is able to get the CPU	2	6	6

	for executing. • Others are waiting. • Maximize the use of CPU time. • No CPU time is wasted by the system waiting for other programs to complete. • The ultimate goal keep the CPU busy Advantages • High and efficient CPU utilization. • User feels that many programs are allotted CPU almost simultaneously. Disadvantages • CPU scheduling is required. • To accommodate many jobs in memory, memory management is required	2 2		
II(2)	**Deadlock Prevention** • Mutual Exclusion: Mutual Exclusion holds for non-sharable resources. Example : A printer can not be shared simultaneously by several processes. Some resources are intrinsically non-sharable. • Hold and Wait : To ensure that the hold and wait condition never occurs in the system, we must guarantee that whenever a process requests a resource it does not hold any other resource. • No Pre-emption: If a process is holding some resources and requests another resource that cannot be immediately allocated to it, then all resources currently being held are pre-empted. • Circular Wait : To ensure that this condition never holds is to impose a total ordering of all resource types and to require that each process request resource in a increasing order of enumeration.	1.5 1.5 1.5 1.5	6	6
II(3)	**Logical Address and Physical Address** • Logical Address: ➢ An address generated by the CPU. ➢ Virtual address • Physical Address: ➢ An address seen by the memory unit i.e, the one loaded into the memory address register is commonly referred to as a physical address. • The compile time and load time binding methods generate identical logical and physical addresses. • The runtime mapping from virtual to physical address is done by a hardware device called the memory management unit (MMU).	3 3	6	6

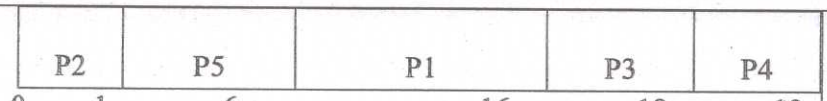
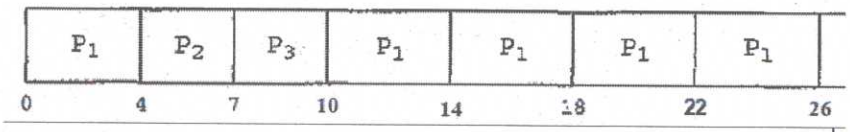
II(4)	**Single-Level Directory** • The simplest directory structure. • All files are contained in the same directory. • All files have unique names. • If two users call their data file *test,* then the unique-name rule is violated. • A single user on a single-level directory may find it difficult to remember the names of all the files as the number of files increases. • Figure **Two-Level Directory** • Create a *separate* directory for each user. • Each user has his own user file directory . • Figure	3 3	6	6
II(5)	**Preemptive and Non Preemptive Scheduling** 1. When a process switches from the running state to the waiting **2.** When a process switches from the running state to the ready state 3. When a process switches from the waiting state to the ready state 4. When a process terminates 1 or 4 : **nonpreemptive** or **cooperative.** 2 or 3: **preemptive.**	3*2	6	6
II(6)	**Fixed Partition Method :-** • It is the simplest method . • Divide the memory into fixed sized partitions. • Each partition may contain exactly one process. • When a partition is free , a process is selected and loaded into the free partition. • When the process terminates, the partition becomes available for another process. **Variable Partition :-** • Memory is partitioned into variable sized blocks. • One of the following strategies is used: • **The First Fit, Best Fit, Worst Fit** • First Fit: Allocate the first free partition that is big enough to hold the process. • Best Fit: Allocate the smallest free partition that is big enough to hold the process. • Worst Fit: Allocate the largest free partition. Disadvantage of contiguous memory allocation strategy is fragmentation. Two types of fragmentation: internal and external.	3 3	6	6
II(7)	**Operating System and its functions** An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.	1	6	6

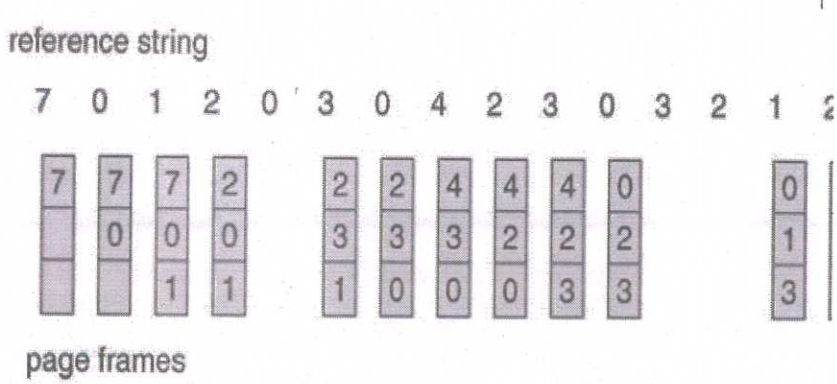
	FUNCTIONS: • Memory Management • Processor Management • Device Management • File Management • Security • Coordination between other software and users	5		
III(a)	**Types of Operating Systems** **(a) Batch Operating System** ➢ The users do not interact with the computer directly. ➢ Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. ➢ To speed up processing, jobs with similar needs are batched together and run as a group. ➢ The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches. ➢ The problems with Batch Systems are as follows: • Lack of interaction between the user and the job. • CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU. • Difficult to provide the desired priority. **(b) Time-sharing Operating Systems** ➢ Time-sharing is a technique which enables many people, located at various terminals, to ➢ use a particular computer system at the same time. ➢ Time-sharing or multitasking is a logical extension of multiprogramming. ➢ Processor's time which is shared among multiple users simultaneously is termed as time-sharing. Advantages of Timesharing operating systems are as follows: • Provides the advantage of quick response • Avoids duplication of software • Reduces CPU idle time Disadvantages of Time-sharing operating systems are as follows: • Problem of reliability • Question of security and integrity of user programs and data • Problem of data communication **(c) Real-Time Operating System** ➢ Data processing system ➢ The time taken by the system to respond to an input and display of required updated information is termed as the response time ➢ The response time is very less . ➢ Well-defined, fixed time constraints, ➢ **Hard real-time systems** ➢ **Soft real-time systems.**	3*3	9	15

III(b)	**Multiprocessor Systems** Have two or more processors in close communication, sharing the computer bus and sometimes the clock, memory, and peripheral devices. Advantages: **1. Increased throughput.** **2. Economy of scale.** **3. Increased reliability.** Types. **1 Asymmetric multiprocessing** **2 Symmetric multiprocessing (SMP**	1.5 1.5 3		
IV (a)	**Assembler:** ➢ A program which translates an assembly language program into a machine language program is called an assembler. ➢ If an assembler which runs on a computer and produces the machine codes for the same computer then it is called self assembler or resident assembler. ➢ If an assembler that runs on a computer and produces the machine codes for other computer then it is called Cross Assembler. **Assembler's functions** ➢ Convert mnemonic operation codes to their machine language equivalents ➢ Convert symbolic operands to their equivalent machine addresses. ➢ Build the machine instructions in the proper format. ➢ Convert the data constants to internal machine representations ➢ Write the object program and the assembly listing **Compiler:** ➢ Program which translates a high level language program into a machine language program. ➢ More intelligent than an assembler. ➢ Occupies a larger part of the memory. ➢ Slow speed. ➢ If a compiler runs on a computer and produces the machine codes for the same computer then it is known as a self compiler or resident compiler. ➢ If a compiler runs on a computer and produces the machine codes for other computer then it is known as a cross compiler. **Interpreter:** ➢ Program which translates statements of a program into machine code. ➢ It translates only one statement of the program at a time. ➢ It reads only one statement of program, translates it and executes it.	3*3	9	15

	Then it reads the next statement of the program again translates it and executes it.			
IV(b)	• **Introduction:** **Linux** is an illustration of open source programming operating system (OS). **Unix** is a system that is extremely main_stream in colleges, organizations, big enterprises and so on. Microsoft **Windows** is said as progression of graphical interface system created, showcased, and traded by Microsoft. • **Use:** **Linux** can usually install on a broad range of PC software and hardware, going from cellular phones, tablets PCs. The **Unix** system is utilized as a part of web servers, workstations and PCs. **Windows** is a system used on computer's desktops, portable workstations, servers and a few phones. • **Users:** **Linux** is an operating system which is used by everybody; from home clients to engineers and PC lovers alike. **Unix** system were produced fundamentally for servers, mainframes and workstations. **Windows** is an operating system used by everybody. • **Price:** **Linux** can consider as free operating system, free downloaded, distributed with the help of books, magazines and so on. There are valued versions for Linux additionally; however they are typically less expensive as compare to windows. **Unix** have different flavors and also have different price structures as indicated by sellers. **Windows** is not freely available.	3*2	6	
V(a)	**PRIORITY SCHEDULING ALGORITHM** ➢ A priority is allocated with each process ➢ CPU selects the process with highest priority. ➢ Equal priority processes are scheduled in FCFS order. Example: Process / Burst Time / Priority P1 / 10 / 3 P2 / 1 / 1 P3 / 2 / 4 P4 / 1 / 5 P5 / 5 / 2	2*4	8	15

	called a deadlock. ➢ A process may utilize the resource in the following sequence 1. Request : A process makes a request for a resource. 2. Use : Process is operating on the resource 3. Release : The process releases the resource Necessary conditions for deadlocks A deadlock can arise if the following four conditions hold simultaneously in a system. 1. Mutual Exclusion 2. Hold and Wait 3. No Preemption 4. Circular Wait.			
VI(b)	**Threads** • A thread is a flow of execution through the process code, with its own program counter that keeps track of which instruction to execute next, system registers which hold its current working variables, and a stack which contains the execution history. • A thread is also called a lightweight process. **Advantages of Thread** • Threads minimize the context switching time. • Use of threads provides concurrency within a process. • Efficient communication. • It is more economical to create and context switch threads. • Threads allow utilization of multiprocessor architectures to a greater scale and efficiency. **Types of Thread** • User Level Threads -- User managed threads • Kernel Level Threads -- Operating System managed threads acting on kernel, an operating system core.	8	8	
VII(a)	**Page Replacement Algorithms** The page replacement algorithms are , 1. FIFO 2. Optimal 3. LRU 1. **FIFO Page Replacement** • simplest page replacement algorithm. • works in a first in first out fashion.	3*3	9	15

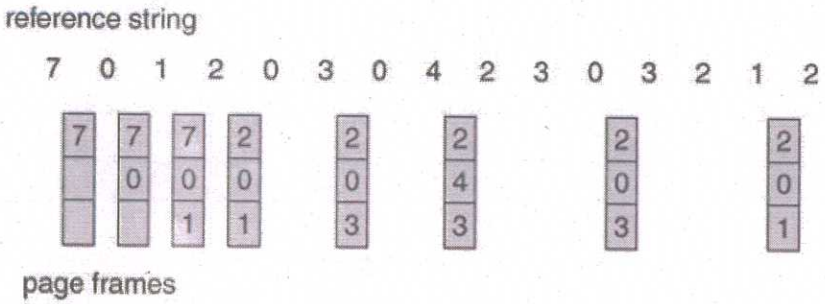
	 ➢ Can be either preemptive or non- preemptive . **ROUND ROBIN SCHEDULING** ➢ Designed for timesharing systems. ➢ It is similar to FCFS scheduling, but preemption is added to switch between processes. ➢ A small unit of time, called a **time quantum** or time slice is used. ➢ A time quantum is generally from 10 to 100 milliseconds. ➢ The ready queue is treated as a circular queue. Process Burst time P1 24 P2 3 P3 3  The average waiting time is 17/3 = 5.66 milliseconds.			
V(b)	**Process Control Block** Each process is represented in the operating system by a process control block (PCB) Also called a task control block. • **Process state.** The state may be new, ready, running, waiting, halted, and so on. • **Program counter.** The counter indicates the address of the next instruction to be executed for this process. • **CPU registers.** The registers vary in number and type, depending on the computer architecture. • **CPU-scheduling information.** This information includes a process priority, pointers to scheduling queues, and any other scheduling parameters. • **Memory-management information.** This information may include such information as the value of the base and limit registers, the page tables. • **Accounting information.** This information includes the amount of CPU and real time used, time limits, account mimbers, job or process numbers, and so on. • **I/O status information.** This information includes the list of I/O devices allocated to the process, a list of open files, and so on.	7	7	
VI(a)	. DEADLOCK AND ITS CAUSES ➢ A process requests a resources and if the resources are not available at that time, the process enters a waiting state. Some times this waiting will not end. This situation is	7	7	15



FIFO page-replacement algorithm.
No of page fault = 15

2. **Optimal Page Replacement**

- Replace the page that will not be used for the longest period of time.

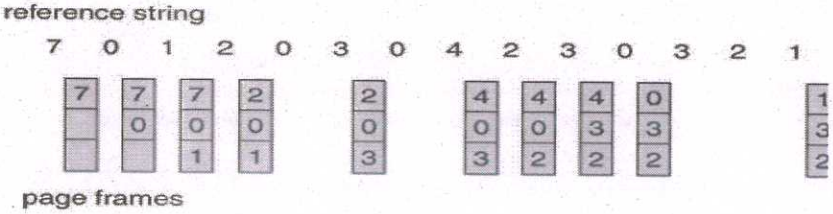


Optimal page-replacement algorithm

No of page fault = 9

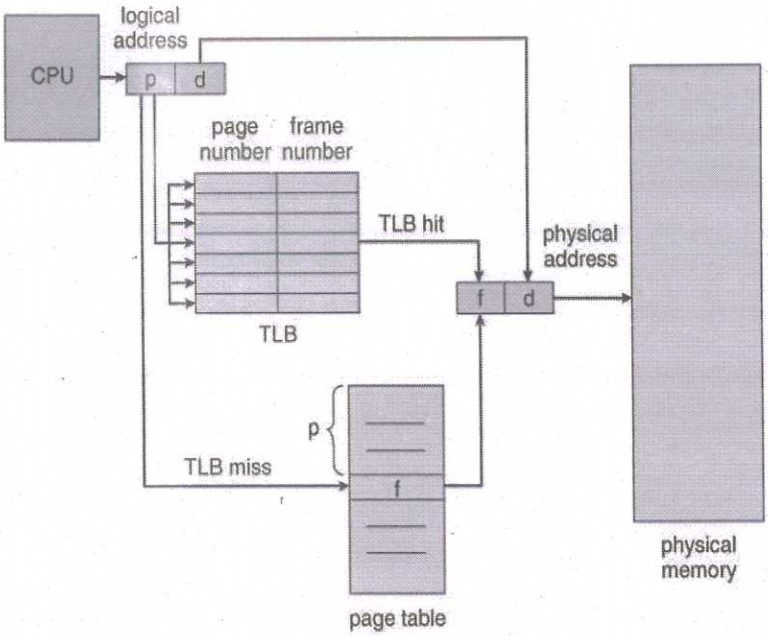
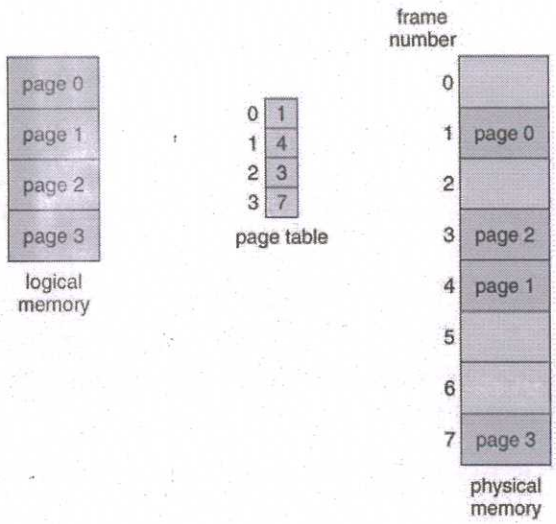
3. **LRU Page Replacement**

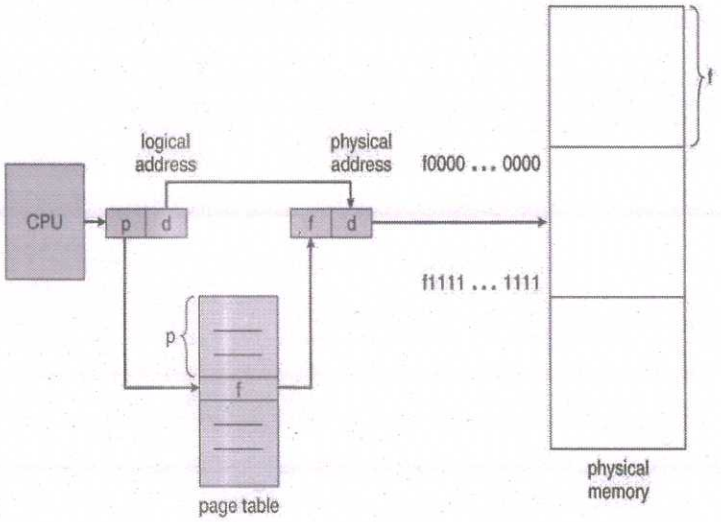
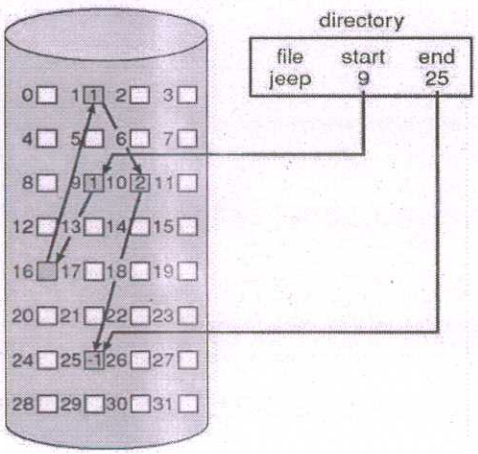
- Approximation of optimal page replacement algorithm.
- Replace the page which is not used for the longest period of time.
- Least recently used (LRU) algorithm.



LRU page-replacement algorithm.

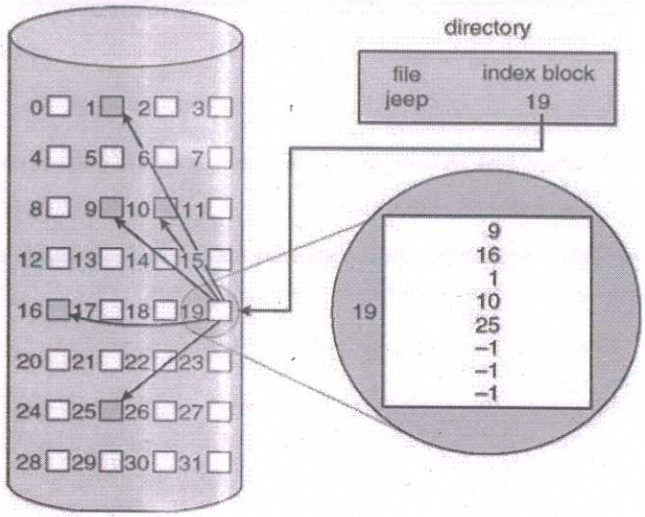
No of page fault = 12

VII(b)	TRASLATION LOOKASIDE BUFFER • Associative high speed memory. • Contains only few of the page table entries, that are accessed frequently. CPU first searches the TLB to get the frame number. • If the page number is found it is known as a **TLB hit**. • If it is not found it is known as a **TLB miss**.  **Paging hardware with TLB**	6	6	
VIII(a)	**Paging** ➢ Paging is a memory management scheme that permits the physical address space of a process to be non-contiguous. ➢ Pages: fixed blocks of logical memory. ➢ Frames: fixed blocks in physical memory.  **Paging model of logical and physical memory** ➢ Pagetable. ➢ Page Fault: if the needed page is not found in page table is	6	6	15

	called page fault.  **Paging hardware**			
VIII(b)	**Thrashing** • high paging activity is called thrashing. • spend more time for paging than executing. **Cause of thrashing** • Multiprogramming **Solutions:** • Working set model • Page Fault Frequency	9	9	
IX(a)	Allocation Methods **(a) Linked allocation** ➢ Solves all problems of contiguous allocation. ➢ Each file is a linked list of disk blocks; the disk blocks may be scattered anywhere on the disk. ➢ The directory contains a pointer to the first and last blocks of the file. ➢ The major problem is that it can be used effectively only for sequential-access files. ➢ Another disadvantage is the space required for the pointers.  	2*5	10	15

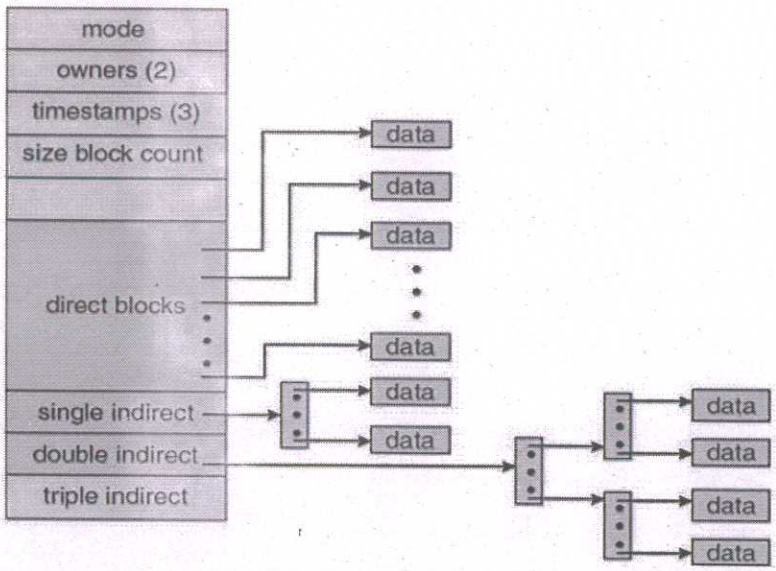
(b) Indexed Allocation

- Linked allocation solves the external-fragmentation and size-declaration problems of contiguous allocation
- **Indexed allocation** solves this problem by bringing all the pointers together into one location: the **index block.**



Mechanisms for this purpose include the following:

- **Linked scheme.**
- **Multilevel index**
- **Combined scheme.**



IX(b)	**Thin Client** ➢ A thin client is a lightweight computer that is purpose-built for remoting into a server. ➢ It depends heavily on another computer (its *server*) to fulfill its computational roles. ➢ This is different from a conventional desktop PC ➢ computer designed to take on these roles by itself. ➢ virtualized applications, a shared desktop stack or virtual	5	5	

	desktop, to data processing and file storage on the client's or user's behalf.			
X (a)	**File Operations** ➢ A file is an **abstract data type.** • **Creating a file.** ➢ Two steps are necessary to create a file. ➢ First, space in the file system must be found for the file. ➢ Second, an entry for the new file must be made in the directory. • **Writing a file.** ➢ make a system call specifying both the name of the file and the information to be written to the file. ➢ Given the name of the file, the system searches the directory to find the file's location. ➢ The system must keep a *write* pointer to the location in the file where the next write is to take place. ➢ The write pointer must be updated whenever a write occurs. • **Reading a file.** ➢ use a system call that specifies the name of the file and where (in memory) the next block of the file should be put. ➢ the directory is searched for the associated entry, and then system needs to keep a *read* pointer to the location in the file where the next read is to take place. ➢ the read pointer is updated. ➢ **position pointer.** Both the read and write operations use this same pointer, saving space and reducing system complexity. • **Repositioning within a file.** ➢ The directory is searched for the appropriate entry, and the current-file-position pointer is repositioned to a given value. ➢ Repositioning within a file need not involve any actual I/O. ➢ This file operation is also known as a file *seek.* • **Deleting a file.** ➢ To delete a file, we search the directory for the named file. ➢ release all file space, so that it can be reused by other files, and erase the directory entry.	2 2 2 2 2	10	15
X (b)	**File Attributes** • **Name:** symbolic file name -human readable form,. • **Identifier:** unique tag - usually a number - identifies the file within the file system - non-human-readable name for the file. • **Type:** types of files. • **Location:** pointer to a device and to the location of the file on that device. • **Size**: current size of the file (in bytes, words, or blocks. • **Protection**. Access-control information. • **Time, date, and user identification**. creation, last modification, and last use.	5	5	