

SCHEME OF EVALUATION

OPERATING SYSTEM(5132)

PART A

1	The software used for an application is known as Application software. Eg. MS-Word	1	1
2	Compiler-Translates the whole high level language program into object code. Interpreter – Translates the high level language program line by line into object code.	1	1
3	Short term scheduler, Long term scheduler and medium term scheduler	1	1
4	F/CFS and SJF	1	1
5	Demand Paging ,Segmented paging, <i>demand segmentation</i>	1	1
6	Compile time binding -If the address of memory location where the process will reside in memory is know at compile time, then absolute code can be generated.	1	1
7	LRU stands for Least Recently Used	1	1
8	A file is a collection of data that can be read from or written to. A directory is a collection of files	1	1
9	File operations are Creating a file, Writing a file, Reading a file, Deleting a file and repositioning a file	1	1

(PART-B)

1	<p>A multiprocessor system consists of several processors that share a common physical memory. Multiprocessor system provides higher computing power and speed. Increases the performance and throughput of the system</p>	3	3														
2	<p>Assembler is a translating program which translates the assembly language program(Source Program) to machine language program(Object program) Loader is a program which loads the program into memory for execution Linker is a program which combines all modules into the main program during execution</p>	1 1 1	3														
3	<table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 50%;">pointer</td> <td style="width: 50%;">process state</td> </tr> <tr> <td colspan="2">process number</td> </tr> <tr> <td colspan="2">program counter</td> </tr> <tr> <td colspan="2">registers</td> </tr> <tr> <td colspan="2">memory limits</td> </tr> <tr> <td colspan="2">list of open files</td> </tr> <tr> <td colspan="2">• • •</td> </tr> </table>	pointer	process state	process number		program counter		registers		memory limits		list of open files		• • •		3	3
pointer	process state																
process number																	
program counter																	
registers																	
memory limits																	
list of open files																	
• • •																	
4	<p>Resource Allocation Graph shows the relation between process and resource. The graph consists of a set of vertices and edges. The vertices are processes and resources. The edges are of two types. Request Edge – Process → Resource and assignment edge From Resource → Process Or Resource Allocation graph example</p>	3 x 1	3														
5	<p>Priority Scheduling: A priority number (integer) is associated with each process. The CPU is allocated to the process with the highest priority. It is either</p> <ul style="list-style-type: none"> - Pre-emptive - Non-pre-emptive 	3 x 1	3														

6	<p>Logical address – An address generated by the CPU is referred to as a logical address.</p> <p>Physical address – An address seen by the memory unit—<i>that is, the one loaded into the memory-address register of the memory</i>—is referred to as a physical address.</p> <p>The set of all logical addresses generated by a program is a logical address space. The set of all physical addresses corresponding to these logical addresses is a physical address space.</p>	1 1 1	3
7	<p>If the process tries to access a page that was not brought into memory, a page-fault trap occurs.</p> <p>The procedure for handling page fault is as follows:</p> <ol style="list-style-type: none"> 1. An internal table is maintained for this process, to determine whether the reference was a valid or invalid memory access. 2. If the reference was invalid, the process is terminated. If it was valid, but yet not brought in that page, that page is brought in. 3. A free frame is found. 	3	3
8	<p>Fragmentation is the free memory space(hole) results after memory allocation. There are two types of fragmentation.</p> <ol style="list-style-type: none"> 1. Internal Fragmentation – fragmentation within a page. 2. External Fragmentation 	3 x 1	3

9	<p>First Come First Serve (FCFS)</p> <p>Shortest Seek Time First(SSTF)</p> <p>Scan Scheduling</p> <p>C-Scan scheduling</p> <p>Look Scheduling</p>	3 x 1	3
10	<p>Indexed Allocation</p> <ul style="list-style-type: none"> • Each file has its own index block, which is an array of disk-block addresses. • The i^{th} entry in the index block points to the j^{th} block of the file. • The directory contains the address of the index block. • To find and read the i^{th} block, we use the pointer in the j^{th} index-block entry. <p>Indexed allocation supports direct access.</p>	3	3

PART –C

III	<p>Functions of Operating system</p> <p>1. Booting: Booting is a process of starting the computer operating system starts the computer to work. It checks the computer and makes it ready to work.</p> <p>2. Memory Management</p> <p>Allocation and de allocation of memory to programs are handled by OS</p> <p>3. Loading and Execution</p> <p>Operating system provides the facility to load programs in memory easily and then execute it.</p> <p>4. Data security</p> <p>The operating system protects the data stored on the computer from illegal use, modification or deletion.</p> <p>5. Disk Management</p> <p>Operating system manages the disk space. It manages the stored files and folders in a proper way.</p> <p>6. Process Management</p> <p>CPU can perform one task at one time. if there are many tasks, operating system decides which task should get the CPU.</p> <p>7. Device Controlling</p> <p>operating system also controls all devices attached to computer. The hardware devices are controlled with the help of small software called device drivers.</p> <p>8. Printing controlling</p>	7	7
-----	--	---	---

IV	<p>Operating system also controls printing function. If a user issues two print commands at a time, it does not mix data of these files and prints them separately.</p> <p>9. Providing interface : Provides user interface .</p> <p>Different operating systems are Batch OS, Multiprogramming, Multiprocessing, Time sharing and Real time OS</p> <p>Batch System :</p> <ul style="list-style-type: none"> • The user has to submit a job (written on cards or tape) to a computer operator. • Then computer operator places a batch of several jobs on an input device. • Jobs are batched together by type of languages and requirement. • Then a special program, the monitor, manages the execution of each program in the batch. <p>Multiprogramming System:</p> <ul style="list-style-type: none"> • In this the operating system, picks and begins to execute one job from memory. • Once this job needs an I/O operation operating system switches to another job (CPU and OS always busy). • In Multiprogramming system, CPU will never be idle and keeps on processing. <p><u>Time-Sharing Systems</u></p> <p>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. In time sharing systems the prime focus is on minimizing the response time.</p> <p><u>REAL-TIME OPERATING SYSTEM</u></p> <ul style="list-style-type: none"> • It is defined as an operating system known to give maximum time for each of the critical operations that it performs, like OS calls and interrupt handling. There are two types of real time operating system ; Hard Real time operating system and soft Real time operating system. 	1	3 x2 7
----	--	---	--------------

V Each process may be in one of the following states:

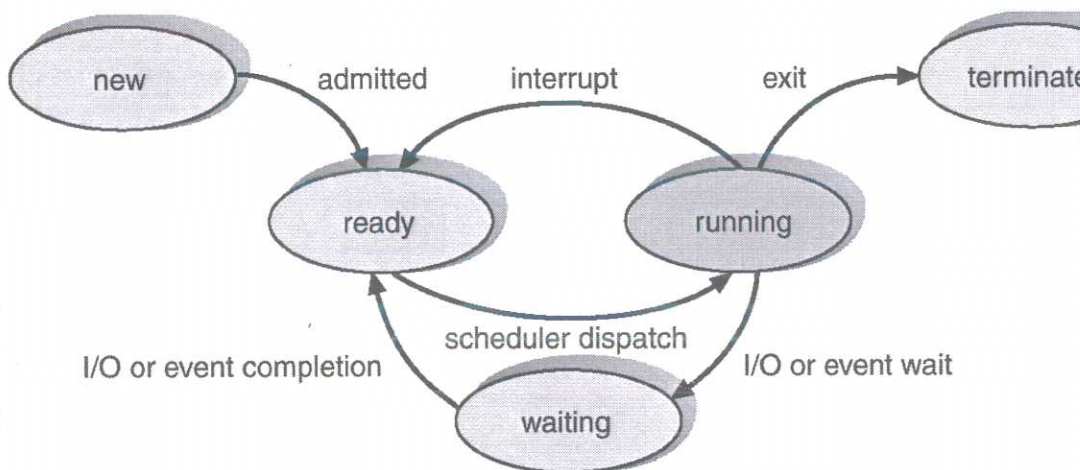
New: The process is being created.

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.



VI Process Synchronization is the coordination of execution of multiple processes in a multi-process system to ensure that they access shared resources in a controlled and predictable manner. The main objective of process synchronization is to ensure that multiple processes access shared resources without interfering with each other, and to prevent the possibility of inconsistent data due to concurrent access. To achieve this, various synchronization techniques such as semaphores, monitors, and critical sections are used.

In a multi-process system, synchronization is necessary to ensure data consistency and integrity, and to avoid the risk of deadlocks and other synchronization problems.

Synchronization problems are race condition and deadlock.

VII A deadlock in OS is a situation in which more than one process is blocked because it is holding a resource and also requires some resource that is acquired by some other process.

Necessary Conditions for a deadlock

A deadlock situation can arise if the following four conditions hold simultaneously in a system

VIII

1. **Mutual exclusion.** At least one resource must be held in a non sharable mode; that is, only one process at a time can use the resource. If another process requests that resource, the requesting process must be delayed until the resource has been released.
2. **Hold and wait.** A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.
3. **No preemption.** Resources cannot be preempted; that is, a resource can be released only voluntarily by the process holding it, after that process has completed its task.
4. **Circular wait.** A set {P0, P1, ..., Pn} of waiting processes must exist such that P0 is waiting for a resource held by P1, P1 is waiting for a resource held by P2, ..., Pn-1 is waiting for a resource held by Pn, and Pn is waiting for a resource held by P0.

The Gantt chart is

P2	P4	P3	P1
0	3	8	18
			38

The waiting time for P2= 0 ms
 The waiting time for P4= 3 ms
 The waiting time for P3= 8 ms
 The waiting time for P1 =18 ms
 Average waiting time =(0+3+8+18)/4 =7.25 ms

IX

FIFO Page Replacement

- The simplest page-replacement algorithm is a **first-in, first-out (FIFO)** algorithm.
- A FIFO replacement algorithm associates with each page the time when that page was brought into memory.
- When a page must be replaced, the oldest page is chosen.

Disadvantage

Belady's Anamoly

Optimal Page Replacement

- Replace page that will not be used for longest period of time
- An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms and will never suffer from Belady's anomaly.
- Unfortunately, the optimal page-replacement algorithm is difficult to implement, because it requires future knowledge of the reference string.

LRU Page Replacement

- LRU replacement associates with each page the time of that page's last use.

4
x
1.
5

7

4

3

7

7

7

X	<ul style="list-style-type: none"> When a page must be replaced, LRU chooses the page that has not been used for the longest period of time. <p>The allocation strategies are</p> <p>First-fit: Allocate the <i>first</i> hole that is big enough. Searching can start either at the beginning of the set of holes or at the location where the previous first-fit search ended. The searching stops as soon as a free hole is found that is large enough.</p> <p>Best-fit: Allocate the <i>smallest</i> hole that is big enough; must search entire list, unless ordered by size. Produces the smallest leftover hole.</p> <p>Worst-fit: Allocate the <i>largest</i> hole; must also search entire list. Produces the largest leftover hole.</p>	7	7
XI	<p>First-fit and best-fit better than worst-fit in terms of speed and storage utilization.</p> <p>The different address binding schemes are</p> <p>Compile time ,load time and Execution time</p> <ul style="list-style-type: none"> Compile time: <ul style="list-style-type: none"> → If the address of memory location where the process will reside in memory is know at compile time, then absolute code can be generated. For example, if you know that a user process will reside starting at location R, then the generated compiler code will start at that location and extend up from there. Load time: <ul style="list-style-type: none"> → Must generate <i>relocatable</i> code if memory location is not known at compile time. → If memory location is not known at compile time where the process will reside in memory, then the compiler must generate relocatable code. → In this case, final binding is delayed until load time. Execution time: <ul style="list-style-type: none"> → If the process can be moved during its execution from one memory segment to another, then binding must be delayed until run time. → Need hardware support for address maps 	1 3 x 2	7
XII	<p>Advantages of segmentation over paging</p> <ul style="list-style-type: none"> No internal fragmentation Improved memory utilization Provides virtual memory 	7	7

XIII	<ul style="list-style-type: none"> • Dynamic linking and loading • Facilitate shared segments • Infer controlled access • Average segment size >> average page size • less overhead (smaller tables) compared to variable partitioning <p>There are two types of file organizations. 1. Sequential file and 2. Indexed File</p> <p>Sequential File organization</p> <ul style="list-style-type: none"> • The simplest file organization. • Sequential access is used • Reading or writing data records in sequential order, that is, one record after the other. • Information in the file is processed in order, one record after the other. <p>Indexed File Organization</p> <ul style="list-style-type: none"> • The index, contains pointers to the various blocks. • To find a record in the file, we first search the index and then use the pointer to access the file directly and to find the desired record. • This structure allows us to search a large file doing little I/O. • The primary index file would contain pointers to secondary index files, which would point to the actual data items. 	7 1 3 3	
XIV	<p>Directory Structures : Single level, Two level and Tree level</p> <p>1) Single-Level Directory</p> <ul style="list-style-type: none"> • The simplest directory structure is the single-level directory. • All files are contained in the same directory, which is easy to support and understand <p>Since all files are in the same directory, they must have unique names.</p> <p>2) Two-Level Directory</p> <ul style="list-style-type: none"> • In the two-level directory structure, each user has his own user file directory (UFD). • The UFDs have similar structures, but each lists only the files of a single user. • When a user job starts or a user logs in, the system's master file directory (MFD) is searched. • The MFD is indexed by user name or account number, and each entry points to the UFD for that user. 	1 2 2	7

3) Tree-Structured Directories

- A tree is the most common directory structure.
- The tree has a root directory, and every file in the system has a unique path name.
- A directory (or subdirectory) contains a set of files or subdirectories.
- A directory is simply another file, but it is treated in a special way.
- With a tree-structured directory system, users can be allowed to access, in addition to their files, the files of other users.

2