INTERNAL ASSESSMENT TEST 1 – MAY 2021
Scheme

Sub:	**OPERATING SYSTEMS**					Sub Code:	17CS64	Branch:	**ISE & CSE**
Date:	19-05-2021	Duration:	90 min's	Max Marks:	50	Sem / Sec:	VI		OBE

Answer any 5 Questions (5 X 10 = 50)

1 (a)	Explain the user and system view of Operating system with required diagram.	[05] 2.5 +2.5	CO1	L2

1. User View
- The user's view of the computer depends on the interface being used:
  1. Most users use a PC consisting of a monitor, keyboard and system-unit.
  - The OS is designed mostly for ease of use.
  - Some attention is paid to performance.

    No attention is paid to resource utilization.
  - The OS is optimized for the single-user experience.

  2. Some users use a terminal connected to a mainframe or (a minicomputer).
  - The OS is designed

    → to maximize resource utilization.

    → to assure that no individual user takes more than her fair share.

  3. Some users use a workstation connected to network.
  - The users have dedicated resources such as networking and servers.
  - The OS is designed to compromise between

    → individual usability and

    → resource utilization.

  4. Some users use a handheld computer.
  - The OS is designed mostly for individual usability.
  - Performance per unit of battery life is a very important factor.

2. System View

1. An OS as a resource allocator
   - Resources used to solve a computing-problem:

     → CPU time

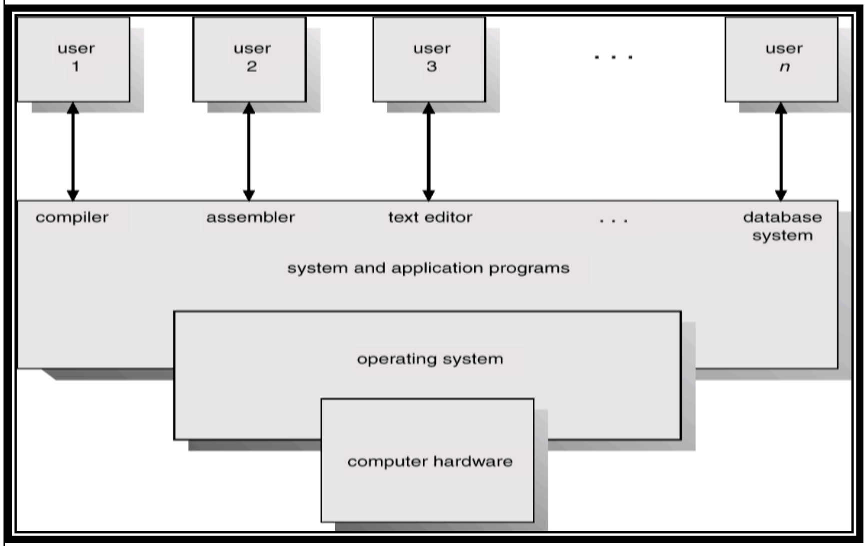
     → memory-space

     → file-storage space and

     → I/0 devices.

- The OS manages and allocates the above resources to programs and the users.

2. An OS is a control program
   - The OS is needed to control:
     → operations of I/0 devices and
     → execution of user-programs to prevent errors.



(b) What is multiprocessor system? Compare Multiprogramming with Multiprocessing system. [05] 2+ 3 CO1 L2

ANS:

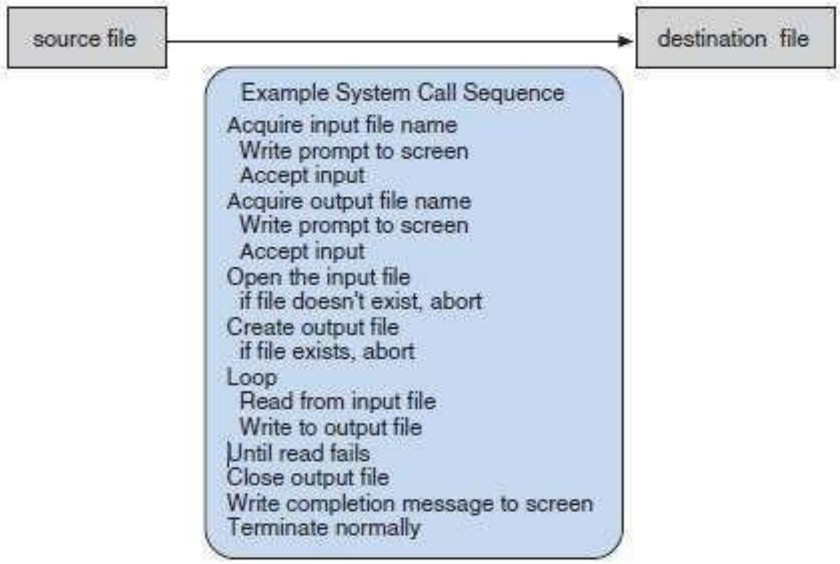
In a uni-processor system, only one process executes at a time. Multiprocessing is the use of two or more CPUs (processors) within a single Computer system. The term also refers to the ability of a system to support more than one processor within a single computer system. Now since there are multiple processors available, multiple processes can be executed at a time. These multi processors share the computer bus, sometimes the clock, memory and peripheral devices also.

Difference between Multi programming and Multi processing –

- A System can be both multi programmed by having multiple programs running at the same time and multiprocessing by having more than one physical processor. The difference between multiprocessing and multi programming is that Multiprocessing is basically executing multiple processes at the same time on multiple processors, whereas multi programming is keeping several programs in main memory and executing them concurrently using a single CPU only.
- Multiprocessing occurs by means of parallel processing whereas Multi programming occurs by switching from one process to other (phenomenon called as context switching).

2 (a)	List and explain the activities of Operating Systems for the following. i) Process Management ii) Memory management Ans: i) Process Management A *process* is a program in execution. A process needs certain resources, including CPU time, memory, files, and I/O devices, to accomplish its task. The operating system is responsible for the following activities in connection with process management. • Process creation and deletion. • process suspension and resumption. • Provision of mechanisms for: 1.process synchronization 2. process communication ii) Memory management • Memory is a large array of words or bytes, each with its own address. It is a repository of quickly accessible data shared by the CPU and I/O devices. • Main memory is a volatile storage device. It loses its contents in the case of system failure. • The operating system is responsible for the following activities in connections with memory management: • Keep track of which parts of memory are currently being used and by whom. • Decide which processes to load when memory space becomes available. • Allocate and deallocate memory space as needed.	[05] 2.5 + 2.5	CO1	L2
(b)	What are System Calls? Discuss the need of System calls with examples. System Calls • These provide an interface to the OS services. • These are available as routines written in C and C++. • The programmers design programs according to an API. (API=application programming interface). • The API → defines a set of functions that are available to the programmer (Figure 1.15). → includes the parameters passed to functions and the return values.	[05] 2+5	CO1	L2

- The functions that make up an API invoke the actual system-calls on behalf of the programmer.
- Benefits of API:
    1. Program portability.
    2. Actual system-calls are more detailed (and difficult) to work with than the API available to the programmer.
- Three general methods are used to pass parameters to the OS:
    1. via registers.
    2. Using a table in memory & the address is passed as a parameter in a register (Figure 1.16).
    3. The use of a stack is also possible where parameters are pushed onto a stack and popped off the stack by the OS.



```
Example System Call Sequence
Acquire input file name
  Write prompt to screen
  Accept input
Acquire output file name
  Write prompt to screen
  Accept input
Open the input file
  if file doesn't exist, abort
Create output file
  if file exists, abort
Loop
  Read from input file
  Write to output file
Until read fails
Close output file
Write completion message to screen
Terminate normally
```

Figure 1.15 Example of how system calls are used.

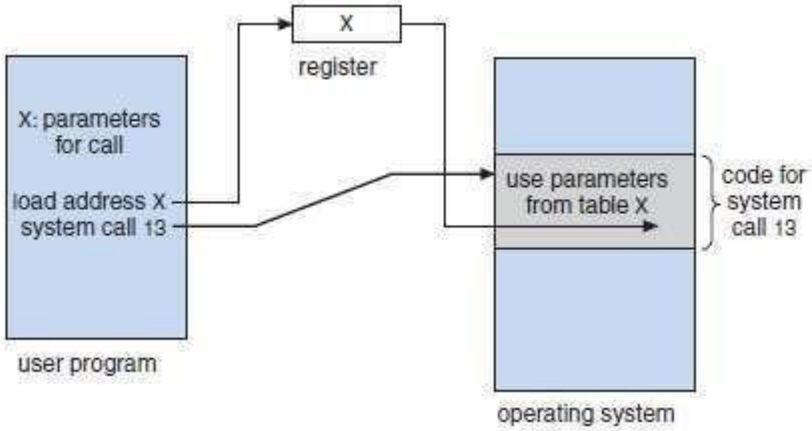
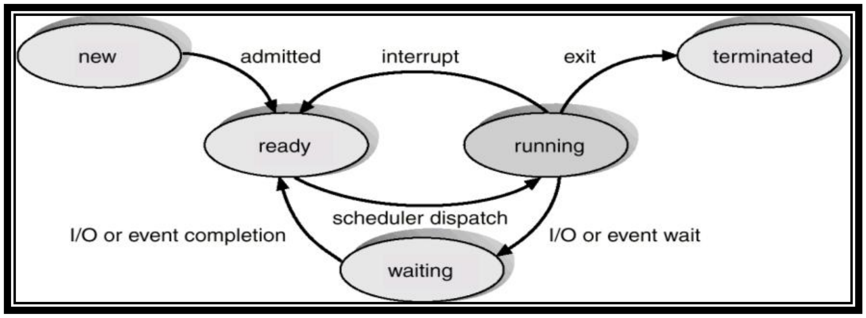
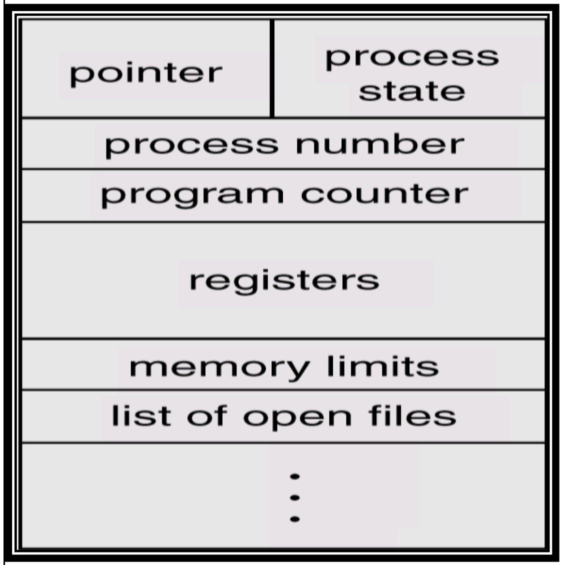


Figure 1.16 Passing of parameters as a table.

3 (a)	Define Process. Explain different process states with state diagram. Process – a program in execution; process execution must progress in sequential fashion. As a process executes, it changes *state* • **new**: The process is being created. • **running**: Instructions are being executed. • **waiting**: The process is waiting for some event to occur. • **ready**: The process is waiting to be assigned to a CPU. • **terminated**: The process has finished execution.  new admitted interrupt exit terminated ready running scheduler dispatch I/O or event completion I/O or event wait waiting	[07] 1+3+3	CO1	L2
(b)	List and explain the elements of Process Control Block. pointer process state process number program counter registers memory limits list of open files ⋮ Information associated with each process. 1. Process state: waiting, running etc..	[03] 1+2	CO1	L2

	2. Program counter 3. CPU registers 4. CPU scheduling information: priority if any 5. Memory-management information: page table, base and limit address 6. Accounting information: amount of cpu time used, process no. 7. I/O status information: list of I/O devices allocated, list of open files			
4 (a)	Differentiate Scheduler and Dispatcher ANS: Consider a situation, where various processes are residing in the ready queue waiting to be executed. The CPU cannot execute all of these processes simultaneously, so the operating system has to choose a particular process on the basis of the scheduling algorithm used. So, this procedure of selecting a process among various processes is done by the scheduler. Once the scheduler has selected a process from the queue, the dispatcher comes into the picture, and it is the dispatcher who takes that process from the ready queue and moves it into the running state. Therefore, the scheduler gives the dispatcher an ordered list of processes which the dispatcher moves to the CPU over time.	[04] 2 + 2	CO 1	L2
(b)	Discuss the Direct and Indirect communication methods used in Inter Process Communication. ANS: IPC is a Mechanism for processes to communicate and to synchronize their actions. Message system – processes communicate with each other without resorting to shared variables. 1. IPC facility provides two operations: F **send**(*message*) – message size fixed or variable F **receive**(*message*) 2. If *P* and *Q* wish to communicate, they need to: F establish a *communication link* between them F exchange messages via send/receive **Direct Communication:** Processes must name each other explicitly: 1. **send** (*P, message*) – send a message to process P 2. **receive**(*Q, message*) – receive a message from process Q Properties of communication link 3. Links are established automatically. 4. A link is associated with exactly one pair of communicating processes. 5. Between each pair there exists exactly one link. 6. The link may be unidirectional, but is usually bi-directional.	[06] 3 + 3	CO1	L2

Indirect Communication

Messages are directed and received from mailboxes (also referred to as ports).

- Each mailbox has a unique id.
- Processes can communicate only if they share a mailbox.

Properties of communication link

- Link established only if processes share a common mailbox
- A link may be associated with many processes.
- Each pair of processes may share several communication links.
- Link may be unidirectional or bi-directional.

Operations in indirection communication are:

- create a new mailbox
- send and receive messages through mailbox
- destroy a mailbox

Primitives are defined as:

send(*A, message*) – send a message to mailbox A

receive(*A, message*) – receive a message from mailbox A

5 (a) Consider three CPU-intensive processes, which require 10, 20 and 30 time units and arrive at times 0, 2 and 6, respectively. With description, explain how many context switches are needed if the operating system implements a Round Robin Scheduling Algorithm? Do not count the context switches at time zero and at the end. Time Slice is 5 time units. [02] CO2 L3

ANS:

process	Arrival time	Burst time
P1	0	10
P2	2	20
P3	6	30

ANS:

P1	P2	P3	P1	P2	P3	P2	P3	P2	P3	P3	P3

0 5 10 15 20 25 30 35 40 45 50 55 60

Number of Context Switching= 9

<table>
<tr><td>b)</td><td>“Priority Scheduling algorithm may cause starvation”
i) Justify the above Statement
ANS: Priority based scheduling if higher priority process keep on coming then low priority process will suffer from starvation.
ii) Discuss the solution for starvation:
Ageing is a scheduling technique used to avoid starvation. Aging is a technique of gradually increasing the priority of processes that wait in the system for a long time. For example, if priority range from 127(low) to 0(high), we could increase the priority of a waiting process by 1 Every 15 minutes. Eventually even a process with an initial priority of 127 would take no more than 32 hours for priority 127 process to age to a priority-0 process.</td><td>[04]
2 + 2</td><td>CO2</td><td>L3</td></tr>
<tr><td>c)</td><td>For the processes listed in the following table, which scheduling algorithm will give the lowest Average waiting Time? Prove it with Gantt chart.
<table>
<tr><th>Process</th><th>Arrival Time</th><th>Burst time</th></tr>
<tr><td>P1</td><td>0</td><td>3</td></tr>
<tr><td>P2</td><td>1</td><td>6</td></tr>
<tr><td>P3</td><td>4</td><td>4</td></tr>
<tr><td>P4</td><td>6</td><td>2</td></tr>
</table>
1. FCFS
<table>
<tr><td>P1</td><td>P2</td><td>P3</td><td>P4</td><td></td></tr>
<tr><td>0</td><td>3</td><td>9</td><td>13</td><td>15</td></tr>
</table>
WT of P1=0
WT of P2= 3-1-0=2
WT of P3= 9-4-0=5
WT of P4=13-6-0=7
Average Waiting Time=(0+2+5+7)/4=3.5
2. SJF (non-preemptive)
<table>
<tr><td>P1</td><td>P2</td><td>P4</td><td>P3</td><td></td></tr>
<tr><td>0</td><td>3</td><td>9</td><td>11</td><td>15</td></tr>
</table>
WT of P1=0
WT of P2=3-1-0=2
WT of P3=11-4-0=7
WT of P4=9-6-0=3</td><td>[04]</td><td>CO 2</td><td>L3</td></tr>
</table>

Average Waiting Time=(0+2+7+3)/4=3

Conclusion:SJF scheduling algorithm perform better with low average waiting time compared to FCFS.

6 (a) Define the following: i) Turnaround Time of a process ii) Throughput of a CPU — [02] CO2 L1

1 + 1

i) Turnaround time is the total amount of time spent by the process from coming in the ready state for the first time to its completion.

ii) Throughput is a way to find the efficiency of a CPU. It can be defined as the number of processes executed by the CPU in a given amount of time.

(b) Consider the following set of processes with CPU burst time (in ms). Calculate the Average Waiting Time and Average Response Time by drawing Gantt chart using — [08] CO2 L3

4(2+2) + 4(2+2)

I)FCFS (First Come First Serve) Algorithm.

ii) Shortest Job First Scheduling Algorithm(Preemptive).SRTF

Process	Arrival Time	Burst time
P1	0	8
P2	1	6
P3	2	5
P4	3	6

ANS:

I)FCFS (First Come First Serve) Algorithm.

P1	P2	P3	P4

0 8 14 19 2 5

WT of P1= 0

WT of P2=8-1-0=7

WT of P3=14-2-0=12

WT of P4=19-3-0=16

Average Waiting Time=(0+7+12+16)/4= 8.75ms

Response Time:

Response Time of p1=0

Response Time of p2=7

Response Time of p3=12

Response Time of p4=16

Average Response Time=(16+12+7+0)/4 =8.75 ms

ii) Shortest Job First Scheduling Algorithm(Preemptive).

Process	Arrival Time	Burst time
P1	0	8
P2	1	6
P3	2	5
P4	3	6

P1	P2	P3	P4	P1

0 1 7 12 18 25

WT of P1= 18-1-0=17

WT of P2=1-1-0=0

WT of P3=7-2-0=5

WT of P4=12-3-0=9

Average Waiting Time=(17+0+5+9)/4 =7.75ms

Response Time:

Response Time of p1=0

Response Time of p2=0

Response Time of p3=5

Response Time of p4=9

Average Response Time=(0+0+5+9)/4 =3.5ms