SHAMBHUNATH INSTITUTE OF ENGINEERING AND TECHNOLOGY

Subject :Operating System Subject Code :KCS-401

B.Tech. : 2nd Year SEMESTER : 4th

SOLUTION FIRST SESSIONAL EXAMINATION, EVEN SEMESTER, (2019-2020)

Branch : Computer Science and Engineering

Time – 1 hr. 30 min. **Max Marks – 30**

SECTION – A

Question 1(a): short-term scheduler (CPU scheduler)

Question 1(b): Output is 15

Question 1(c): SJF scheduling algorithm

Question 1(d): Round-robin scheduling

Question 1(e): 11

SECTION – B

Question 2(a):

Long Term Scheduler/Job Scheduler

- It selects process from Job Queue and assigns it to Ready Queue.
- It changes state of processes from New State to Ready State.

Short Term Scheduler/CPU Scheduler

- It selects process from Ready Queue and assigns it to CPU.
- It changes state of processes from Ready State to Running State.
- Dispatcher is responsible for saving the context of one process (i.e. content of PCB) and loading the context of another process.

Medium Term Scheduler

- It removes the processes from Main Memory.
- It reduces degree of multiprogramming.
- It is responsible for swapped out process.

Dispatcher: Dispatcher is part of Short Term Scheduler that performs following functions.

- Context switching
- Switching to User Mode
- Jumping to the proper location in the user program to restart that program

Question 2(b):

Life Cycle of Process

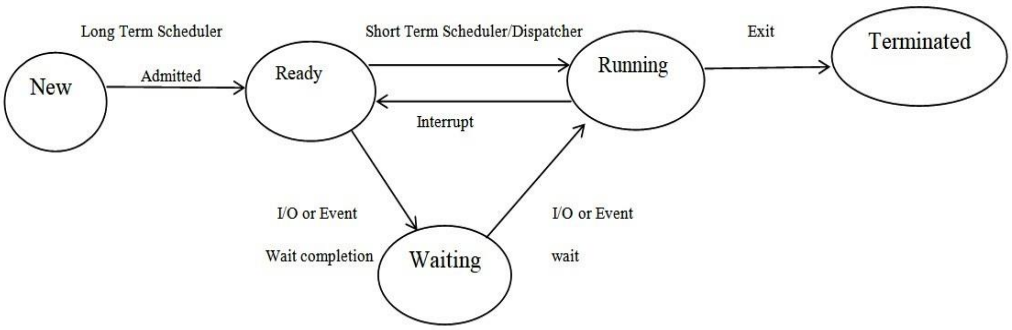
New State: When a process is created it is in new state. In new state process resides in Job Queue (Secondary Memory).

Ready State: State of process is called Ready State when Process resides in Ready Queue (Primary Memory) and waiting for CPU.

Running State: State of process is called Running state if CPU is assigned to process.

Waiting State: State of process is called Waiting State when Process resides in waiting Queue (Primary Memory) and waiting for some event like I/O to occur.

Terminated State: State of process is called Terminated State when the Process has completed its execution successfully.



Question 2(c):

Gantt Chart:

P1	P2	P3	P2	P4	P1
0	**3**	**7**	**8**	**10**	**13**	**20**

Process TAT=Completion time − Arrival time

P1 20
P2 7
P3 1
P4 5

Avg TAT= 33/4=8.25

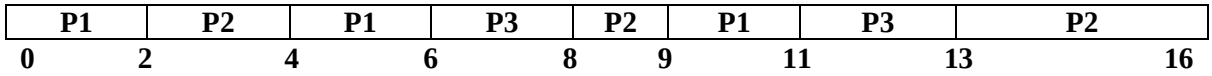
WT:-

P1 10
P2 1
P3 0
P4 2

Avg WT=13/4=3.25

Question 2(d):

Gantt chart:



WT-

P1-> (0-0)+(4-2)+(10-6)=6

P2-> (2-1)+(8-4)+(13-10)=8

P2-> (6-3)+(11-8)=6

AVG WT:- 20/3=6.67

TAT:-

P1:- (11-0)=11

P2:- (16-1)=15

P3:- (13-3)=10

AVG TAT:- 36/3=12

SECTION – C

Question 3(a): <u>Batch System</u>

- ❖ Batch means set of jobs with similar need.
- ❖ In this system, Operator makes Batch of jobs and loads it into punch card.
- ❖ Similar types of jobs batch together and execute at a time.

Advantages

- ✓ Save time from activities like loading compiler.
- ✓ No manual intervention by user is needed.

Disadvantages

- ✓ Limited Memory
- ✓ Interaction of I/O directly with CPU.
- ✓ CPU is often idle.

Question 3(b): Multiprogramming Operating System

- ❖ Main objective of multiprogramming is to maximize CPU utilization.
- ❖ More than one process can reside in main memory which are ready to execute i.e. more than one process is in ready state.
- ❖ In this system if any running process performs I/O or other event which does not require CPU, then instead of sitting idle, CPU performs context switching and picks another process for execution.

Advantages

- ✓ High CPU utilization.
- ✓ Less waiting time, response time etc.
- ✓ Useful in current scenario when load is high.

Disadvantages

- ✓ Process scheduling is difficult.
- ✓ Main memory management is required.
- ✓ Problems like memory fragmentation may occur.

Multiprocessing System

- ❖ A system is called multiprocessing system if two or more CPU within a single computer communicate with each other and share system bus memory and I/O devices.
- ❖ It provides true parallel execution of processes.

Advantages:

- ✓ **Increased Throughput:** By increasing number of processors, we expect to get more work done in less time.
- ✓ **Economy of Scale:** multiprocessor system can cost less than equivalent multiple single processor systems.
- ✓ **Increased Reliability:** Since work is distributed among several processors; failure of one processor will not halt the system only slow it down.

Disadvantages

- ✓ It is complex system.
- ✓ Process scheduling is difficult in this system.
- ✓ It required large size of Main memory.
- ✓ Overhead reduces throughput.

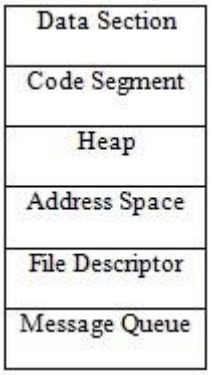
Multithreading

Thread: Light weight Process is called Thread.

- Light weight Process
- Thread Switching does not need interaction with OS
- Thread can share Code segment, Data Section, File Descriptor, address space, heap etc

Threads has its own

Threads of same process share



Note: Local variables of the Process are stored in Stack, Global variables are stored in Data Section and dynamically created variables are stored in Heap.

Question 4(a): Necessary conditions for deadlock

1. **Mutual Exclusion:** At least one resource must be held in a non sharable mode i.e. only one process at a time can use the resource. If another process requests the resource, the requesting process must be delayed until the resource has been released.
2. **Hold and wait:** A process must be holding one resource and waiting to acquire additional resources that are currently being held by other processes.
3. **No Preemption:** Resources cannot be preempted i.e. a resource can be released only voluntarily by the process holding it, after that process has completed its task.
4. **Circular wait**: A set $\{P_0,P_1,\ldots,P_n\}$ of waiting processes must exists such that P_0 is waiting for a resource held by P_1,P_1 is waiting for a resource held by $P_2 \ldots P_{n-1}$ is waiting for a resource held by P_n and P_n is waiting for a resource held by P_0.

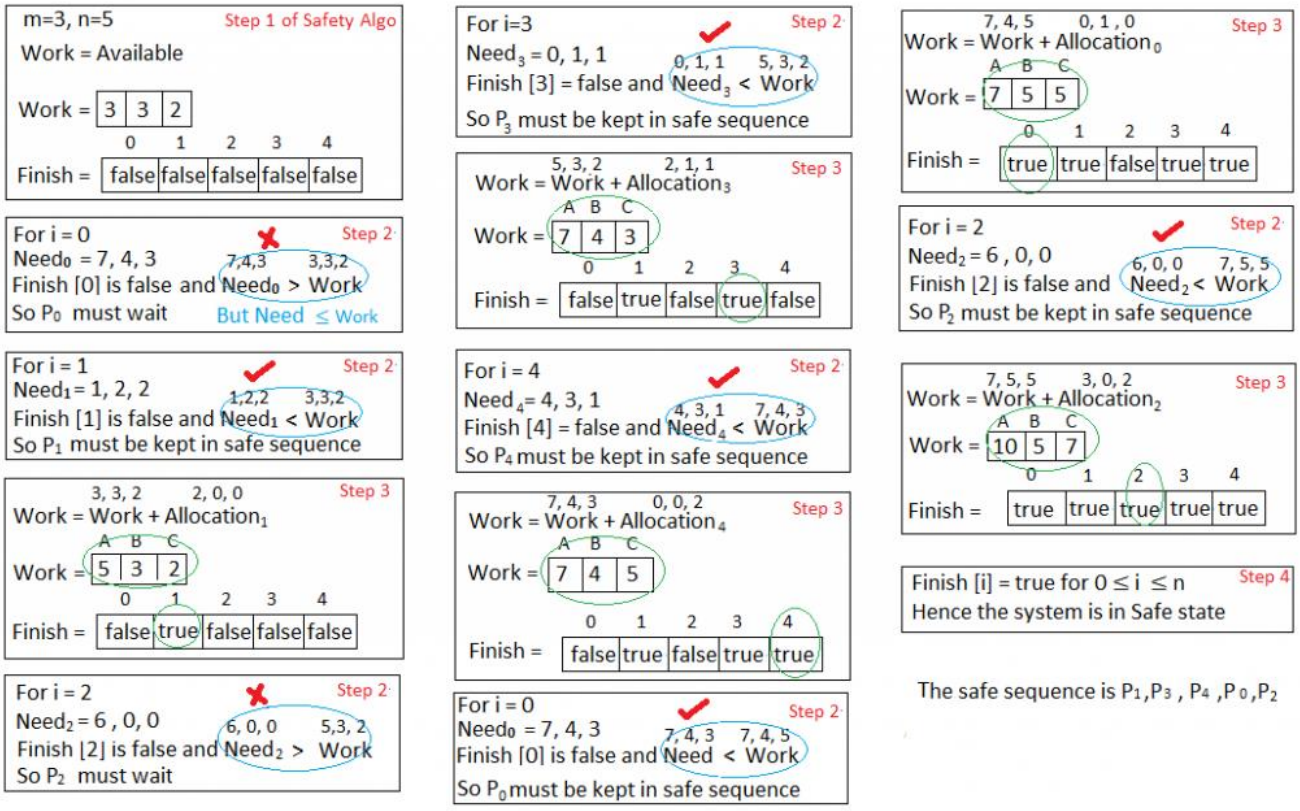
Question 4(b):

Need [i, j] = Max [i, j] – Allocation [i, j]

So, the content of Need Matrix is:

Process	Need		
	A	B	C
P_0	7	4	3
P_1	1	2	2
P_2	6	0	0
P_3	0	1	1
P_4	4	3	1

Applying the Safety algorithm on the given system,



Question 5(a):

Operating System

- Operating system is a system program that act as an interface between hardware and user.
- It manages system resources.
- It provides a platform on which other application programs are installed.

- Core Part of Operating System is called kernel.

Goals of operating System

Primary Goal: To make system user friendly (convenient).

Secondary Goal: To make system efficient.

Role of Operating System: We can explore operating Systems from two viewpoints that of the user and that of the system.

User's View: In user's view operating system is designed to maximize the work (CPU utilization) that the user is performing. In this context OS is designed mostly for **ease of use** with some attention paid to performance and none paid to resource utilization.
System's View: In system point of view operating system is a program that works as a **resource allocator**. Every computer system has many resources like CPU Time, Memory Space, File Storage Space, I/O Device and so on. The operating system act as manager of these resources.

Functions of Operating System

Operating System is responsible for following activities.

Process Management: Operating System is responsible for following activities in connection with Process Management.

- ✓ Scheduling Processes and threads on the CPUs.
- ✓ Creating and deleting both user and system processes.
- ✓ Suspending and resuming processes.
- ✓ Providing mechanisms for process synchronization.
- ✓ Providing mechanism for process communication.

Memory Management: Operating System is responsible for following activities in connection with Memory Management.

- ✓ Keeping track of which parts of memory are currently being used and by whom.
- ✓ Deciding which processes and data to move into and out of memory.
- ✓ Allocating and de-allocating memory space as needed.

Disk Management: Operating System is responsible for following activities in connection with Disk Management.

- ✓ Free space management in disk.
- ✓ Storage Allocation in disk.
- ✓ Disk Scheduling.

File Management: Operating System is responsible for following activities in connection with File Management.

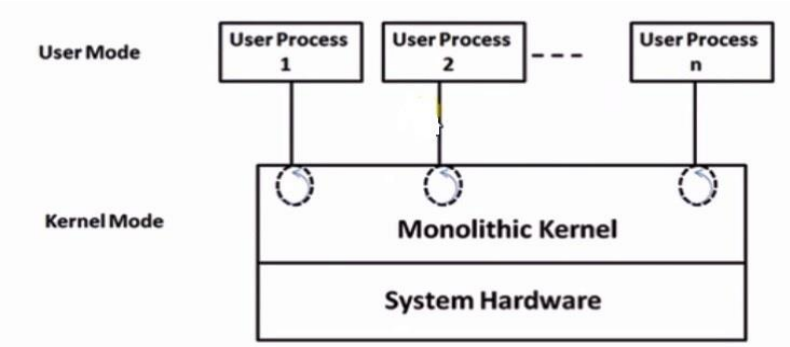
- ✓ Creating and deleting files.
- ✓ Creating and deleting directories to organize files.
- ✓ Supporting primitives for manipulating files and directories.
- ✓ Mapping files onto secondary storage.
- ✓ Backing up files on stable storage media.

I/O Device Management: Operating System is responsible for I/O Device management needed by various processes.
Network Management Operating System is responsible for network management.
Security and Protection: Operating System also provides security and protection to computer system.

Question 5(b):

Monolithic Kernel

- Monolithic means all in one piece.
- In monolithic kernel user services and kernel services are implemented under same address space, It increases size of the kernel thus increase size of operating system.
  Example: Linux



Advantage

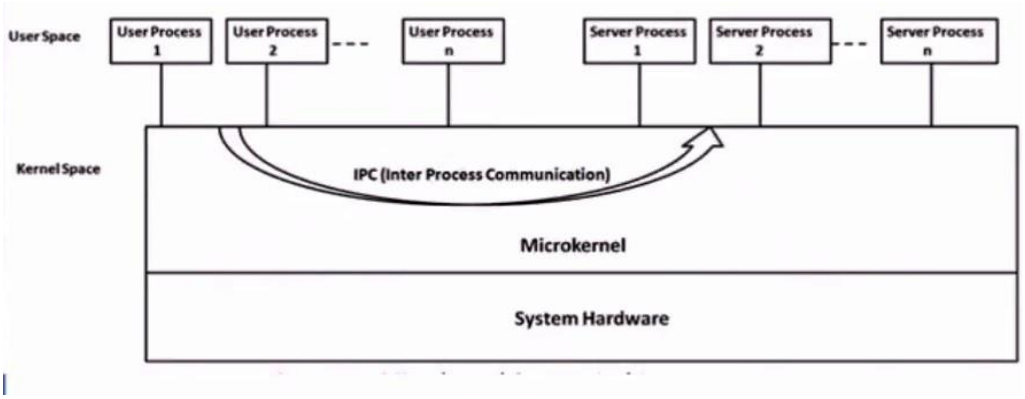
- ✓ Execution is faster.

Disadvantages

- ✓ Size is larger.
- ✓ Hard to extend.
- ✓ Hard to port.
- ✓ More prone to errors and bugs.

Microkernel

- In microkernel, user services and kernel services are implemented in different address space, therefore it also reduces the size of kernel as well as the size of Operating system.
- In this architecture, only the most important services are present inside kernel and rest of the operating system services are present inside system application program.
- Communication between client process and services running in user address space is established through message passing thus reduce the speed of execution.
  Example: Mach OS



Advantages

- ✓ Size is smaller.
- ✓ Easy to extend.
- ✓ Easy to port.
- ✓ Less prone to errors and bugs.

Disadvantage

- ✓ Execution is slower.

Difference between microkernel and monolithic kernel

Parameter	**Microkernel**	**Monolithic Kernel**
Size	Small	Large
Speed of execution	Faster	Slower
Extendibility	Easy	Hard
Portability	Easy	Hard
Error prone capability	Less	More