Assignment : 1 [Operating System]

Date
Page

1. what is difference between Timesharing and multiprogramming System.

ans :- • Multiprogramming :

=> Partition the memory into several pieces with different job in each partition.

=> while one job was waiting for I/O to complete, another job could use CPU.

=> whenever running job finished, the OS load new job from the disk into the empty partition of memory. This is known as SPOOLING (simultaneous Peripheral Operation On line).

• Timesharing :

=> Timesharing system is interactive.

=> First general purpose time sharing system in CTSS (compatible time sharing system).

=> After the success of CTSS system, M.I.T, Bell labs and general Electrical decide to develop "computer utility, a machine that supports hundreds of simultaneous time sharing users. This machine as MULTICS (MULTiplexed Information and Computing System.

	Time sharing	Multiprogramming
1.	Time sharing is the logical extension of multiprogramming.	It allows to execute multiple processes ~~Eg~~ at same time.
2.	It has fixed time slice	It has no time slice.
3.	It minimize response time	It maximize processor use
4.	Ex - Windows NT	Eg - Mac OS

2. What are system call? Explain various types of system calls with example for each.

ans- A system call is a way for programs to interact with the operating system. A computer program makes a system call when it makes a request to the operating system kernel.

→ It provides the services of the operating system to the user program via Application Program Interface (API).

- Types of system calls:
  (i) File system
  (ii) Process control
  (iii) Memory management
  (iv) Interprocess Communication
  (v) Device management

1. File management system calls:
→ open(): opens a file for reading or writing. A file could be of any type like text file, audio file etc.
→ read(): reads data from a file. Just after the file is opened through open() system call then if some process want to read data from file, then it will make a read() system call.
→ close(): closes a previously opened file

2. Process (control) system calls:
→ fork(): create a new process by duplicating the current process.
→ exec(): loads and runs a new program in the current process and replaces the current process with a new process.
→ wait(): The primary purpose of this call is to ensure that the parent process doesn't proceed further with its execution until child process have finished their execution.

→ exit(): It simply terminates the current process.
→ kill(): This call sends a signal to a specific process and has various purpose including - requesting it to quit.

3. Memory management System call:
→ brk(): changes the data segment size for a process in HEAP.
→ sbrk(): This call is also for memory management in heap.
→ mmap(): Memory map - It basically map a file or device into main memory and further into a process's address space for performing operations.
→ munmap(): Unmap's memory mapped file from a process's address space and out of main memory.
→ mlock() and unlock(): memory lock defines a mechanism through which certain pages stay in memory and not swapped out to the swap space in the disk.

4. Inter-Process Communication (IPC) System call:
→ pipe(): Creates a unidirectional communication channel between process
→ socket(): creates a network socket for communication. Processes on same or other network can communicates.
→ shmget(): It is short for shared memory get.
→ semget(): It is short for semaphore-get.
→ msgget(): It is short for message-get.

5. Device management System call:
→ SetConsoleMode(): This call is set mode to set the mode of console.
→ WriteConsole(): It allow us to write data on console screen.
→ ReadConsole(): It allows us to read data from console screen.
→ open(): This calls is made whenever a device or a file is opened
→ close(): This call is made when system or the user closes the file or device.

Date
Page

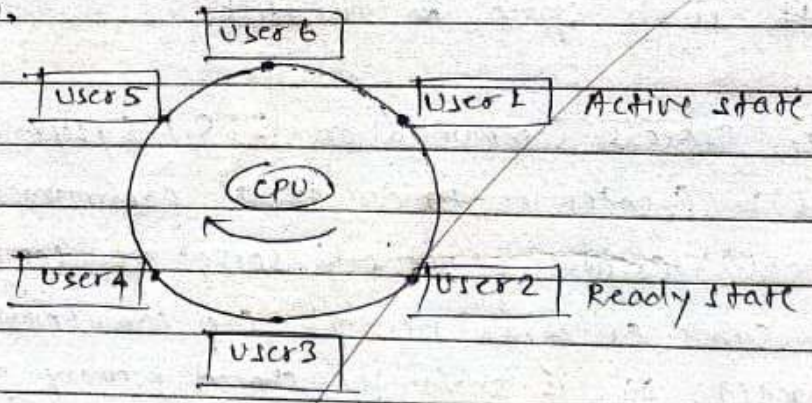
5. Discuss the following types of OS with its essential properties :

a) Time sharing system

b) multi processor system

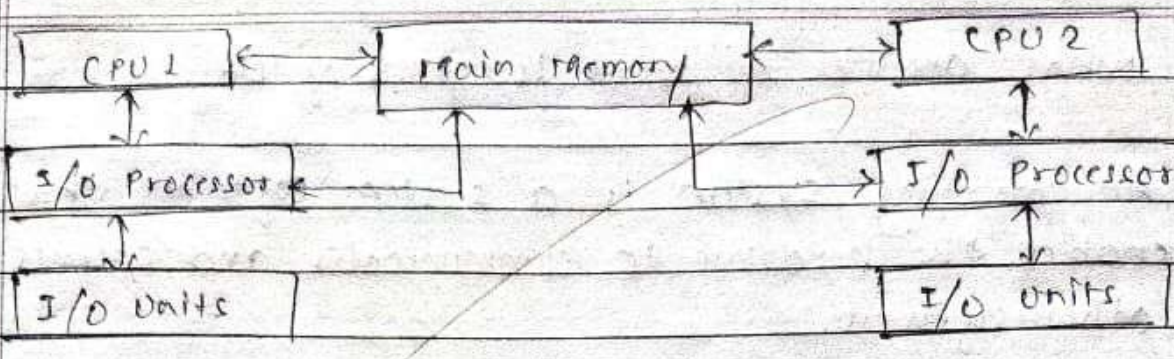
c) Distributed system.

ans- a) Time sharing system: A time shared operating system uses CPU the scheduling and multiprogramming to provide each user with a small portion of a shared computer at once. Each user has atleast one sperate program in memory. A program is loaded into memory and executes, it peoforms a short period of time either before completion or to complete I/O. This short period of time during which the user gets the attention of the CPU's - known as time slice, time slot, or quantum.



Time sharing operating system

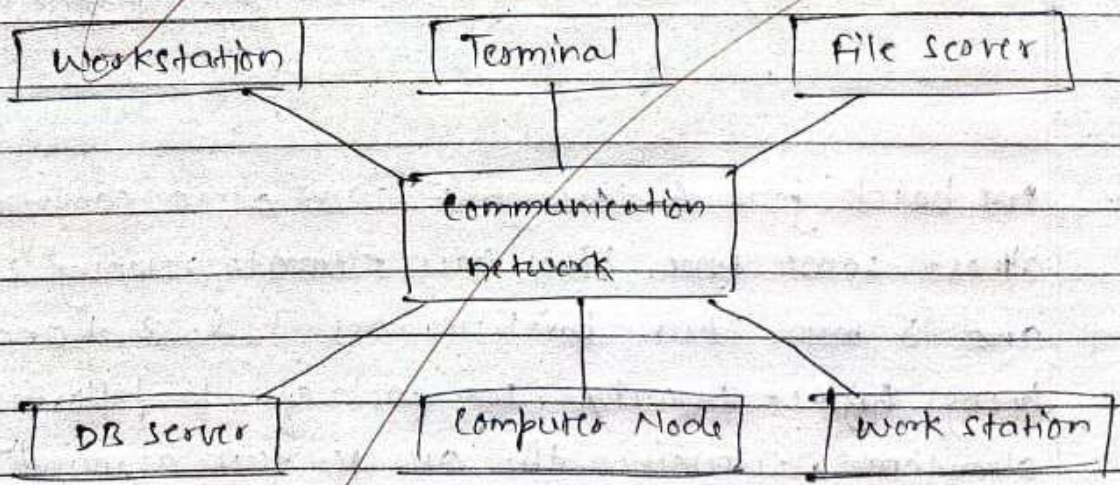
b) multi processor operating system: It is a type of OS that makes use of more than one CPU to improve performance multiple processors work parallely in multi processing OS to perform the given task. All the available processors are connected to peripheral devices, computer buses, physical memory and clocks. It increase the speed of execution of the system. It improve all overall performance of system.



Multiprocessor operating system

c) Distributed operating system: In a distributed os, multiple CPU's are utilized, but for end users, it appears as a typical centralized operating system. It enables the sharing of various resources such as CPU's, disks, network, interfaces, nodes and computers across different sites, thereby expanding the available data within the entire system.

The ability of a distributed OS to share processing resources and I/O files while providing users with a virtual machine abstraction is an important features.

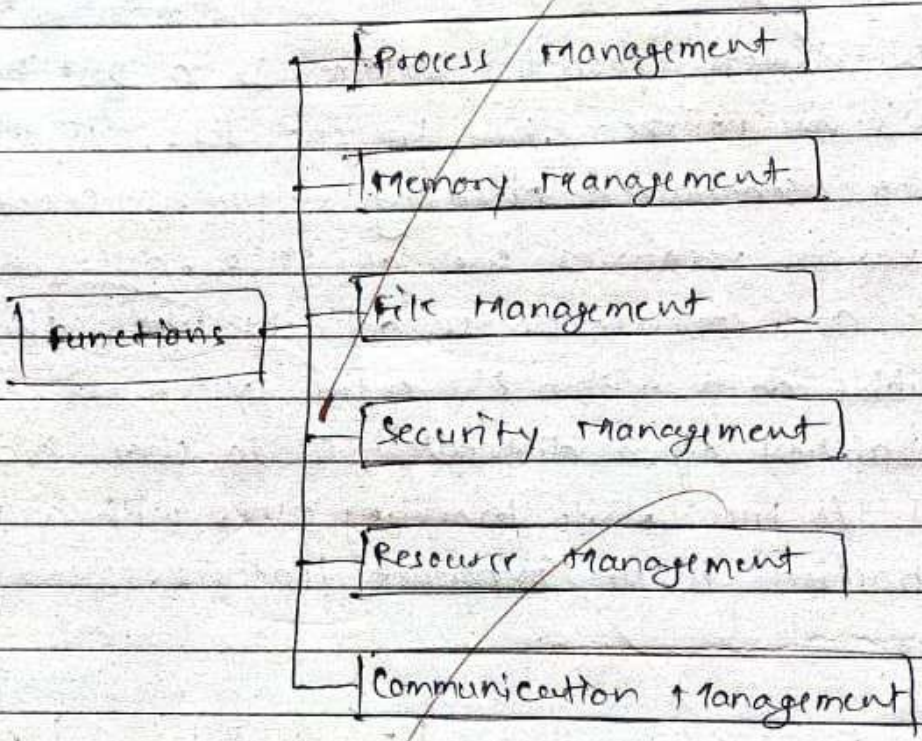


Distributed operating System

4. what are the different functions of OS?

ans- An operating system is a system software that enables the hardware to communicates and operate with other software.

⇒ Functions of operating system:



5. Five batch jobs A through E arrive at a computer center at almost same time. They have estimate running time of 10, 6, 2, 4 and 8 min. Their priorities are 3, 5, 2, 1 and 4 with 5 being highest priority. For each of the following scheduling algorithm, determine the average turn around time and waiting time.

For a) assume quantum = 2

For b) assume that only one job run at a time until it finished

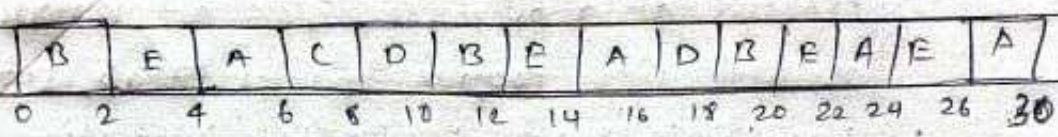
For c) All jobs are completely CPU bound.

Date
Page

Q1" - (A) Round Robin Scheduling :

Process_id	Arrival Time	Burst Time	CT	Turn Around Time	WT	Priority
A	0	10	30	30	20	3
B	0	6	20	20	14	5
C	0	2	8	8	6	2
D	0	4	18	18	14	1
E	0	8	26	26	18	4

Gnatt chart



Ready Queue | B, E, A, C, D, B, E, A, D, B, E, A, E, A |

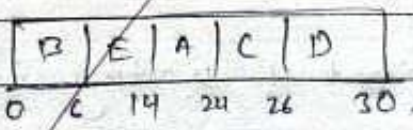
Avg TAT = (30 + 20 + 8 + 18 + 26) / 5 = 20.4

Avg WT = (20 + 14 + 6 + 14 + 18) / 5 = 14.4

(B) Priority Scheduling -

Process_id	Priority	Arrival Time	Burst Time	Turn Around Time	Waiting Time
A	3	0	10	24	14
B	5	0	6	6	0
C	2	0	2	26	24
D	1	0	4	30	26
E	4	0	8	14	6

Gnatt chart



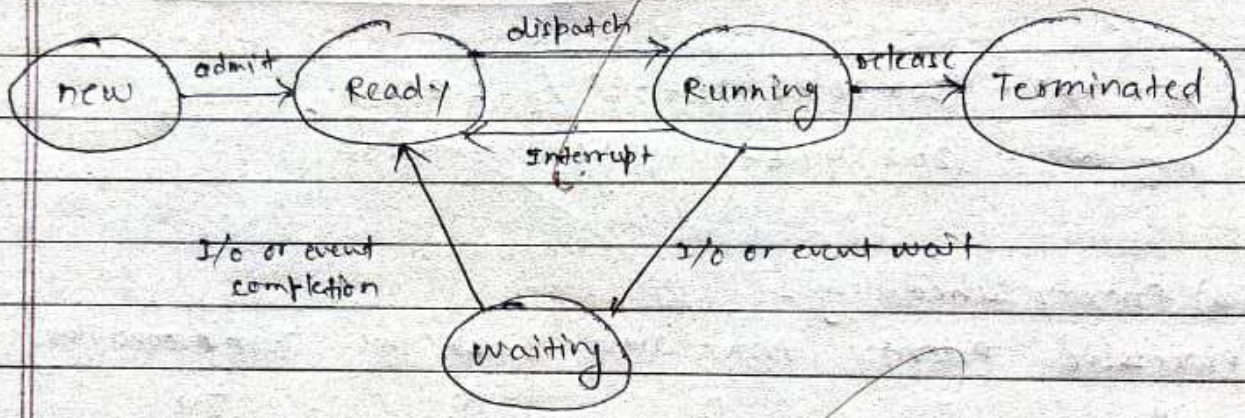
Avg TAT = (24 + 6 + 26 + 30 + 14) / 5 = 20

Avg WT = (14 + 0 + 24 + 26 + 6) / 5 = 14

6. what is process? Discuss the components of process and various states of a process with the help of state transition diagram.

ans- A process is a program in execution, generally also includes the process stack, which contains temporary data. It is also called as an entity that can be assigned and execute on a processor. There are various components of process as well like Program counter, Heap, stack, Process Control Blocked.

- Five-state Process Model State Transition Diagram:



1. Running: The currently executing process
2. waiting/Blocked: Process waiting for some event such as completion of I/o operation, waiting for other process, synchronization signal, etc.
3. Ready: A process that is waiting to be executed.
4. New: The process that is just being created. The program control block is already being made but the program is not yet loaded in the main memory.
5. Terminated/Exit: A process that is finished or aborted due to some reason.

7. Differentiate between the following
a) CPU utilization and Response time.
b) Average turn arround time and maximum waiting time.
c) I/o Device utilization and CPU utilization.
d) User level thread and Kernel level thread.

Ans:- a) **CPU utilization**: This refers to the percentage of time CPU is busy, that is, not in the idle state. A CPU is considered utilized when it is executing instructions. High CPU utilization means the processor is spending most of its time executing instruction rather than sitting idle waiting for tasks.

Response time: In context of operating system, response time is the time interval between a process entering the ready queue and getting scheduled on the CPU for the first time.

Response time = CPU allocation time − Arrival time

b) **Average turn around time**: Turn around time is the amount of time required to execute a specific process. and when we take the average of all turn around time for a period of time that is called as average turn around time.

Turn Around time = Brust time + Waiting time.

Maximum waiting time: It is the amount of time when process or thread waiting in the queue.

Waiting time = Turn around time − Brust time

c) I/O Device utilization : It refers to the measure of how much an Input/output device is being used by the system. Efficient utilization of I/O devices is crucial for robust interaction between users and the system.

CPU utilization : This refers to the percentage of time the CPU is busy, that is, not in the idle state. A CPU is considered utilized when it is executing instruction. High CPU utilization means the processor is spending most of its time executing instruction rather than sitting idle.

d.) User level Thread and Kernel level thread : These threads are implemented by user level software and are managed by a thread library provided by the operating system as an API. The operating system is not aware of these threads and handles them as if they were single threaded processes. User level threads are faster to create and manage than kernel level threads.

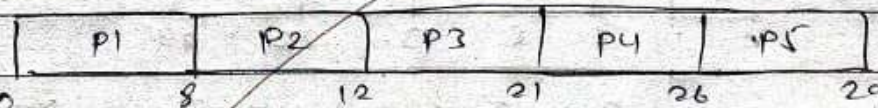
Kernel level threads : These threads are managed directly by the operating system kernel. They are managed entirely in kernel space and do not require any user level library or support.

8. Consider the following set of process with the length of CPU brust time and arrival time as given in ms.

Process ID	Brust Time	Arrival time
P1	8	0
P2	4	1
P3	9	2
P4	5	3
P5	3	4

- Draw four gnatt chart illustrating the execution of these process using FCFS, SJF, SRTS, Priority and round robin (quantum = 2) scheduling. Also calculate average waiting time and turn around time for each.

i) FCFS - First come first serve



Process-id	Brust Time	Waiting Time	Turn Around Time	AT
P1	8	0	8	0
P2	4	8	12	1
P3	9	12	21	2
P4	5	21	26	3
P5	3	26	29	4

Average waiting Time = $(0 + 8 + 12 + 21 + 26)/5 = 13.4$

Average Turn Around Time = $(8 + 12 + 21 + 26 + 29)/5 = 19.2$

(ii) SJF - (Shortest Job First)

P1	P5	P2	P4	P3

0 8 11 15 20 29

Process_id	Brust Time	Waiting Time	Turn Around time	AT
P1	8	0	8	0
P2	4	11	15	1
P3	9	20	29	2
P4	5	15	20	3
P5	3	8	11	4

Average waiting Time = (0 + 11 + 20 + 15 + 8)/5 = 10.8

Average Turn around Time = (8 + 15 + 29 + 20 + 11)/5 = 16.6

(iii) SRTF - (Shortest Remaining Time First)

P1	P2	P5	P4	P1	P3

0 1 5 8 13 20 29

Process_id	Brust Time	Waiting Time	Turn Around Time	AT
P1	8	13	21	0
P2	4	1	5	1
P3	9	20	29	2
P4	5	8	13	3
P5	3	5	8	4

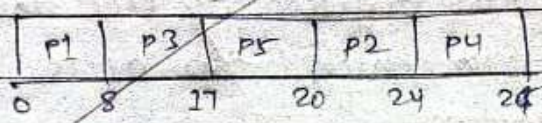
Average waiting Time = (13 + 1 + 20 + 8 + 5)/5 = 9.4

Average Turn Around time = (21 + 5 + 29 + 13 + 8)/5 = 15.2

(iv) Priority : Nature (Non-Pocemptive)

Process_id	Priority	Arrival Time	Burst Time	waiting Time	TAT
P1	3	0	8	0	8
P2	2	1	4	20	24
P3	5	2	9	8	17
P4	1	3	5	24	29
P5	4	4	3	17	20

Gnatt chart



Avg. Waiting Time = (0 + 20 + 8 + 24 + 17)/5 = 13.8

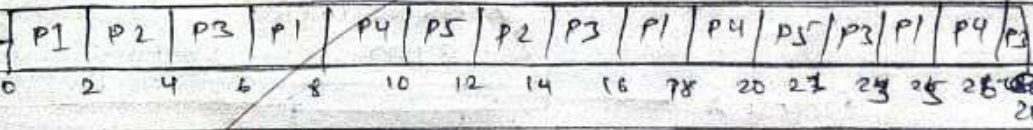
Avg. Turn Around Time = (8 + 24 + 17 + 29 + 20)/5 = 19.6

(v) Round Robin Scheduling :

Process_id	Arival Time	Burst Time	Turn Around Time	waiting Time
P1	0	8		
P2	1	4		
P3	2	9		
P4	3	5		
P5	4	3		

Quantum = 2

Gnatt chart



Ready Queue - P1, P2, P3, P1, P4, P5, P2, P3, P1, P4, P5, P3, P1, P4, P3

Avg TAT = 21

Avg waiting time = 15.2

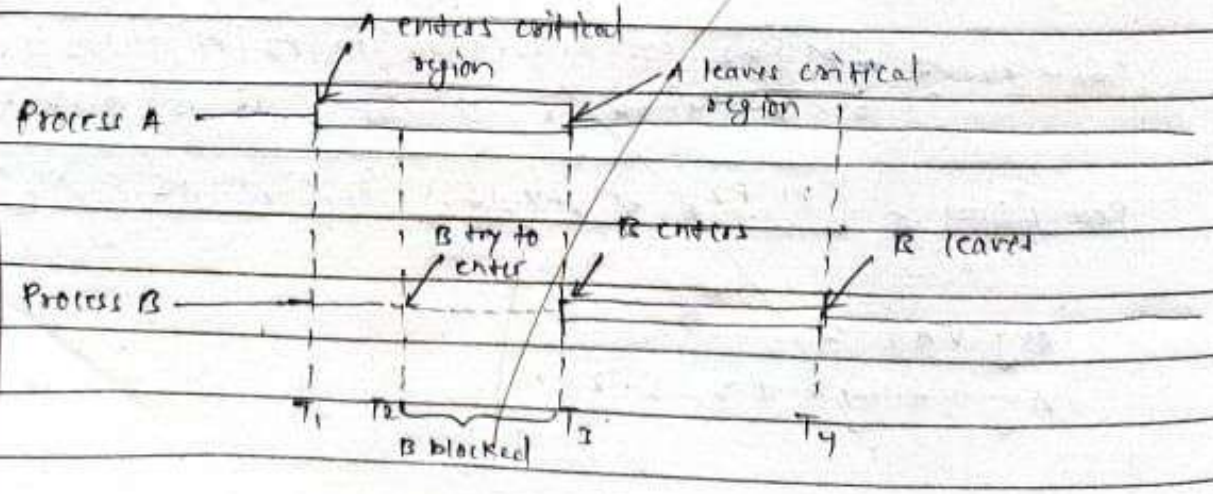
9. Explain Race condition. what is critical section and how implementing critical section can help resolve race condition.

ans - A **Race condition** is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time.
Situation like this where processes access the same data concurrently and the outcome of execution depends on the particular order in which the access takes place is called race condition.

- **Critical section**: The part of program where the shared resources is accessed is called critical section or critical region.

⇒ Sometimes a process has to access shared memory or file, or do other critical things that can lead to races. This type of part of program where shared memory is accessed is called the critical region or critical section.

⇒ If no two processes are ever in their critical regions at the same time then we could avoid race.



10. Explain Dining philosopher Problem in details.

ans- The dining philosopher's problem is the classical problem of synchronization which says that five philosopher's are sitting around a circular table.

=> Their job is to think and eat alternatively.

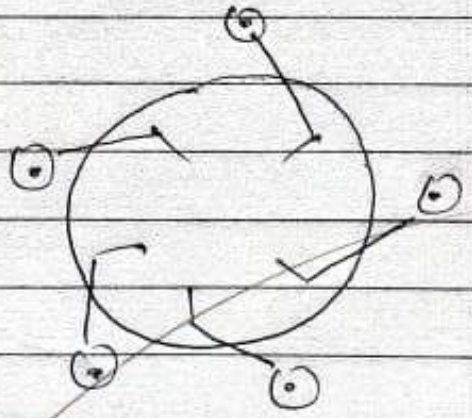
=> A bowl of noodle is placed at the center of the table along with five chopsticks for each of the philosophers.

=> To eat a philosopher needs both their right and left chopstick. A philosopher can ~~ely~~ only eat if both immediate left and right chopsticks of the ~~philosopher~~ philosopher is available.

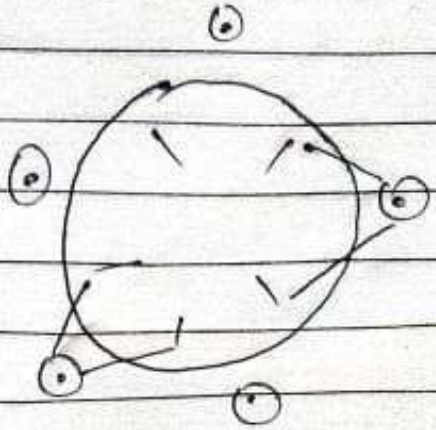
=> In case if both immediate left and right chopsticks of the philosopher are not available then the philosopher puts down their either left or right chopsticks and starts thinking again.

Problem:

1. Deadlock



2. Starvation



9/2