Theory Assignment 1- 2025

Design Principle of Operating Systems (CSE 3249)

Programme: B.Tech. (CSE & CSIT) Semester: 5th

Submission Date: 10 Nov 2025

Course Learning Outcome	*Taxonomy Level
To understand the different components of operating System and various ways of structuring an operating system.	L4
To differentiate the basic design issues involved in creating process and threads.	L4
To analyze the mechanisms involved in handling, scheduling and synchronizing processes.	L4

This assignment is designed to give you practice with the concepts of

- Instruction Execution
- Memory and cache memory
- Uniprogramming and Multiprogramming
- Process creation using fork()
- Process scheduling algorithm

1. Suppose the hypothetical machine contain instructions and data are 16-bits long, and memory is organized as a sequence of 16-bit words. Partial list of opcode has given below:
   0001=Load AC from memory
   0010=Store AC to memory
   0110=Subtract the value of AC from memory
   In these cases, The instructions format provide 4 bit for the opcode and remaining 12-bit of instruction format is word of memory can be directly addressed. Assumed that the program counter is set to location 300. It will fetch instructions from location 301 and 302. Show the program execution using the fetch stage and Execute stage for the following program:
   a. Load AC from memory 940.
   b. Subtract from AC to content of memory location 941.
   c. Store AC to memory location 941.

   Assume that the memory location 940 and 941 contain value of 10 and 5.

2. The program execution of above question (1) is described in the text using six steps. Expand this description to show the use of the memory address register (MAR) and memory buffer register (MBR).

3. Suppose that a large file is being accessed by a computer memory system comprising of a cache and a main memory. The cache access time is 50 ns. Time to access main memory (including cache access) is 550 ns. The file can be opened either in read or in write mode. A write operation involves accessing both main memory and the cache (write-through cache). A read operation accesses either only the cache or both the cache and main memory depending upon whether the access word is found in the cache or not. It is estimated that read operations comprise of 80% of all operations. If the cache hit ratio for read operations is 0.9, what is the average access time of this system?

4. Consider a computer with 400Mbytes of available memory (not used by os). Three program, JOB1, JOB2 and JOB3 are submitted for execution at the same time with attribute listed below.

	JOB1	JOB2	JOB3
Type of job	Heavy compute (70% CPU used)	Heavy I/O (10% CPU used)	Heavy I/O (10% CPU used)
Duration	10 min	20 min	15 min
Memory required	150M	100M	125M

For simple Batch environment, these job are executed in sequence JOB1, JOB2 then JOB3. Find out CPU utilization, memory utilization and Throughput in case of uniprogramming and multiprogramming system.

5. In a batch operating system, three jobs are submitted for execution. Each job involves an I/O activity, CPU time and another I/O activity of the same time span as the first. Job JOB1 requires a total of 23 ms, with 3 ms CPU time; JOB2 requires a total time of 29 ms with 5 ms CPU time; JOB3 requires a total time of 14 ms with 4 ms CPU time. Illustrate their execution and find CPU utilization for uniprogramming and multiprogramming systems.

6. Trace the following program segment and determine how many processes are created. Draw a graph that shows how the processes are related.

```
a. int main( )
{
   pid_t c1,c2;
  c2=0;
  c1 = fork();
  if (c1 == 0)
      c2 = fork();
  if (c2 > 0)
      fork();
  printf(" 1”);
  return 0;
}
```

```
b. int main( )
{
  pid_t c1=1,c2=1;
  c1 = fork();
  if (c1 != 0)
        c2 = fork();
   if (c2== 0)
        fork();
   printf(" 1”);
   return 0;
}
```

```
c. int main( )
  {
    if (fork() || fork())
        fork();
    printf(" 1 ");
    return 0;
}
```

```
d. int main( )
  {
    if (fork() && (!fork()))
         if (fork() || fork())
               fork();
    printf("2 ");
    return 0;
  }
```

7. Consider the following four processes with arrival times (in milliseconds) and their length of CPU bursts (in milliseconds) as shown below:

Process	Arrival time	Burst Time
P1	0	3
P2	1	1
P3	3	3
P4	4	x

Find the value of x, such that the average waiting time of the processes is 1 millisecond, if the processes execute on a single processor using SRTF scheduling.

8. Consider the set of processes with arrival time (in milliseconds), CPU burst time (in milliseconds), and priority (**high number implies high priority**) as shown below.

Process	Arrival time	Burst Time	Priority
P1	0	11	1
P2	0	8	0
P3	12	2	3
P4	2	6	2
P5	9	16	4

Find the average turnaround time, average waiting time and average response time for each of the following scheduling algorithm along with their Gantt chart illustrating the execution of these processes:

- FCFS (Consider the order specified in the table),
- SJF (Non preemptive),
- SRTF,
- Non preemptive Priority based scheduling,
- Preemptive Priority based scheduling,
- Highest Response Ratio Next (Consider the order specified in the table),
- Round robin scheduling (quantum =2ms)

Which of the algorithms results in the minimum average waiting time over all processes?

9. Consider the set of processes with arrival time (in milliseconds), CPU burst time (in milliseconds), and priority (**lowest number implies high priority**) as shown below.

Process	Arrival time	Burst Time	Priority
P1	0	4	3
P2	0	2	1
P3	1	3	2
P4	2	2	4

Find the average turnaround time, average waiting time and average response time for each of the following scheduling algorithm along with their Gantt chart illustrating the execution of these processes:

- FCFS (Consider the order specified in the table),
- SJF (Non preemptive),
- SRTF,
- Non preemptive Priority based scheduling,
- Preemptive Priority based scheduling,
- Highest Response Ratio Next (Consider the order specified in the table),
- Round robin scheduling (quantum =2ms)

Which of the algorithms results in the minimum average waiting time over all processes?

10. Consider a multilevel feedback queue scheduling (MLFBQ) with three queues q1, q2 and q3, where q1 has the highest priority and q3 has the lowest priority. q1 and q2 use round robin algorithm with time quantum equal to 3 and 5 milliseconds respectively. q3 uses first-come first-serve algorithm. Assume the arrival time of all processes as 0. A process entering the ready queue will put in queue 0. Processes in queue q1, q2 will be demoted to lower priority queue, if not completed on specified time quantum. Find the average waiting time (A.W.T) and average turnaround time (A.T.A.T) for executing the following process?

process	Burst time
P1	8
P2	22
P3	4
P4	12