**Semester: 4th**
**Sub & Code: OS & CS - 2009**
**Branch (s): CSE, IT, CSSE, CSCE**

# SPRING MID SEMESTER EXAMINATION-2020

School of Computer Engineering
Kalinga Institute of Industrial Technology, Deemed to be University
Operating Systems
[CS 2002]

**Time: 1 1/2 Hours** **Full Mark: 20**

---

***Answer any four Questions including Q.No.1 which is Compulsory.***
***The figures in the margin indicate full marks. Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.***

Answer all the questions:- **[ 1 × 5]**

Define throughput of a system.

**Ans.** Number of processes completed per unit time.

How do you prevent starvation in priority allocation?

**Ans.** Aging technique :- as time progresses, the priority of the waiting process should be increased

Consider the following state changes for a preemptive scheduling OS:

I. Ready to running II. Running to ready

III. Blocked to running IV. Blocked to ready

Which of the above statement(s) are TRUE?

**(A)** I and II only **(B)** I only **(C)** I and III only **(D)** I, II and IV only

**Ans. D**

Following is the code for three cuncurrent processes P0, P1 and P2.

| **Process P0** | **Process P1** | **Process P2** |
|---|---|---|
| while (1){<br>P (S0);<br>print '0';<br>P (S1);<br>P (S2);<br>} | V(S0);<br>V(S1); | V(S0);<br>V(S2); |

The binary semaphores S0, S1, and S2 in the above code are initialized to 1, 0, and 0 respectively. What is the maximum number of "0" printed?

**(A)** 1 **(B)** 2 **(C)** 3 **(D)** None of the above

**Ans. B**

The enter( ) and exit( ) functions are used to implement a critical section of a process with the help of test-and-set( ) instruction as follows:

| | |
|---|---|
| void enter(X){<br>while (test-and-set(X)) *do no-operation* ;<br>} | void exit(X){<br>X = 0 ;<br>} |

(Hint: test-and-set( X ) instruction return previous value of X and set it to 1)

In the above solution, X is a memory location associated with the critical section (CS) and is initialized to 0. Now, consider the following statements:

The above solution to CS problem is deadlock-free

The solution is starvation free

The processes enter CS in FIFO order

More than one process can enter CS at the same time.

Which of the above statements is true?

**(A)** I only **(**B**)** I and II **(C)** II and III **(D)** IV only

**Ans. A**

Write necessary conditions to be satisfied to implement solution of a critical section. Explain how those conditions are satisfied in Peterson's two-process solution? **[5]**

**Ans.** Mutual exclution, Progress and bounded wait (Explain each one them) :- **1.5 Mark**

Peterson's two-process algorithm :- **2 Marks**

Explain how above three conditions are satisfied in Peterson's two-process solution :- **1.5 Mark**

What is a Semaphore? Explain the solution for the bounded buffer problem using a Semaphore with the following criteria. **[5]**

The producer has to fill the buffer from the end, and the consumer has to take the data on a first come first serve basis. The buffer has to fill and release in a circular manner.

**Ans.** Semaphore definition with P and V operation :- **1 Mark**

bounded buffer Solution:- **4 Mark**

| | |
|---|---|
| buffer[MAX<br>in=out=Max-1 | Semaphore mutex = 1<br>Semaphore full = 0<br>Semaphore empty = N. |

| **Produce** | **Consumer** |
|---|---|
| while (true) {<br>item=produce()<br>wait (empty);<br>wait (mutex);<br>buffer[in]=item;<br>in=(in+MAX-1)%MAX<br>signal (mutex);<br>signal (full);<br>} | while (true) {<br>wait (full);<br>wait (mutex);<br>receive_item= buffer[out]<br>out=(out+MAX-1)%MAX<br>signal (mutex);<br>signal (empty);<br><br>} |

**[ 2.5 + 2.5]**

The arrival time and duration of the CPU and I/O bursts for each of the three processes A, B, and C are given in the table below. Each process has a CPU burst followed by an I/O burst followed by another CPU burst. Assume that each process has its own I/O resource.

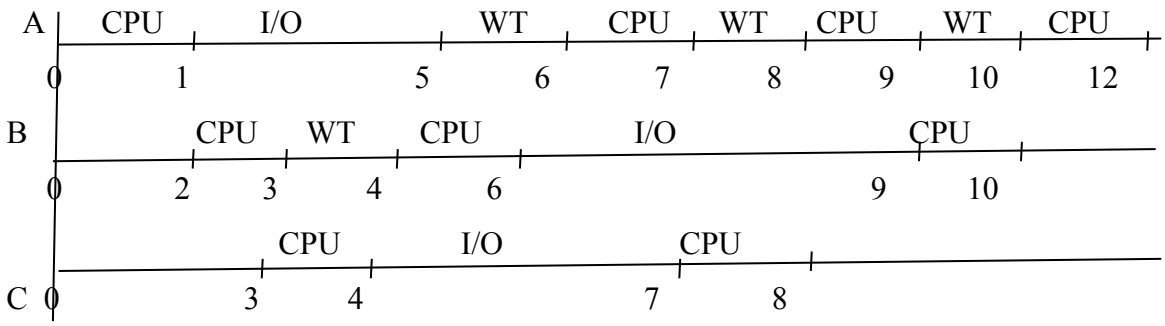
| Process | Arrival time | CPU burst | I/O burst | CPU burst |
|---|---|---|---|---|
| A | 0 | 1 | 4 | 4 |
| B | 2 | 3 | 3 | 1 |
| C | 3 | 1 | 3 | 1 |

The multi-programmed operating system uses the shortest remaining time first ( SRTF) scheduling. What are the completion times of the processes A, B, and C and find individual waiting times of processes?

**Ans.**

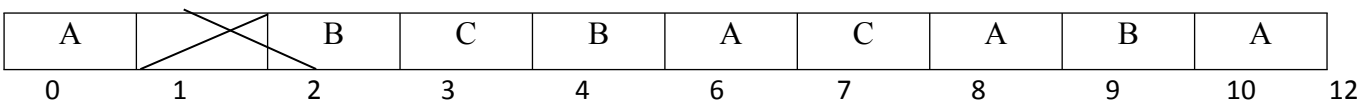
| Process | Arrival time | CPU burst | I/O burst | CPU burst | Completion Time | Wait time |
|---|---|---|---|---|---|---|
| A | 0 | 1, 0 | 4 ( 1 – 5 ) | 4, 3, 2, 0 | 12 | 3 |
| B | 2 | 3, 2, 0 | 3 ( 6 – 9 ) | 1, 0 | 10 | 1 |
| C | 3 | 1, 0 | 3 ( 4 – 7 ) | 1, 0 | 8 | 0 |

Wait time (WT) :-



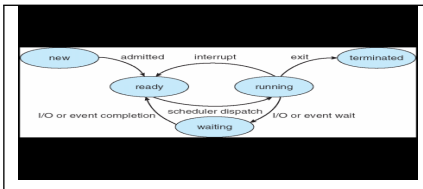

| Process name | Waiting Time | | Waiting Time (Total) |
|---|---|---|---|
| | Start Time | End Time | |
| A | 5 | 6 | 3 |
| | 7 | 8 | |
| | 9 | 10 | |
| B | 3 | 4 | 1 |
| C | Nil | Nil | 0 |

Gantt Chart:-

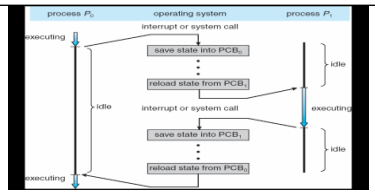



Draw process state diagram to explain the life cycle of a process. What are the steps involed during the context switching?

Ans.




Explain process state diagram : - **1.5 mark**




Explain context switching :- **1 mark**

**[ 2.5 + 2.5]**

Consider the following set of processes arrive in a system.

| Process | CPU Burst Time | Arrival Time |
|---|---|---|
| $P_1$ | 04 | 00 |
| $P_2$ | 19 | 03 |
| $P_3$ | 02 | 01 |
| $P_4$ | 13 | 09 |

Use the Round-robin algorithm (Given Quantum Time = 3units) to calculate average turnaround time of the above process and give the Gantt chart.

**Ans.**

| Process | CPU Burst Time | Arrival Time | Turnaround Time | Avrage TAT |
|---|---|---|---|---|
| $P_1$ | 4, 1, 0 | 00 | 9 – 0 = 9 | 19 |
| $P_2$ | 19, 16,13,10,7,4,1,0 | 03 | 38 – 3 = 35 | |
| $P_3$ | 02, 0 | 01 | 5 – 1 = 4 | |
| $P_4$ | 13, 10,7,4,1,0 | 09 | 37 – 9 = 28 | |

Gantt Chart:-

| $P_1$ | $P_3$ | $P_2$ | $P_1$ | $P_2$ | $P_4$ | $P_2$ | $P_4$ | $P_2$ | $P_4$ | $P_2$ | $P_4$ | $P_2$ | $P_4$ | $P_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 5 | 8 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 37 | 38 |

Explain the difference between batch, multiprogramming, and time-sharing operating systems.

**Ans.** Explain each one of the above OS

*** Best of Luck ***