1. Consider the following page reference string:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

What will be the ratio of page faults for the following replacement algorithms - FIFO replacement and Optimal replacement, assuming three frames?
Remember all frames are initially empty.

A) 1.6 B)2.1 C) 1.45 D) 1.7
Correct Option: C

Explanation: In FIFO, number of page faults is 16, whereas in optimal replacement it is 11. Ratio is 16 / 11 = 1.45

2. Consider a machine which implements virtual memory with pure paging. Suppose virtual addresses are 32 bit and page size is 2KB (kilobytes). How large can a process's page table get (assume page table entry is 8 bytes)?

A) 4 MB B) 8 MB
C) 16 MB D) 32 MB

Correct option: C

Explanation:-

Number of bits for page offset = 11 (2^{11} = 2KB)
Number of pages = $2^{32-11} = 2^{21}$
Max size of page table = $2^{21} * 8$ B = 2^{24} B = 16 MB

3. **Consider the following two simultaneous processes**

```
Process A
While(1)
{
        X
        Print 2;
        Print 2;
        Y
}
Process B
While(1)
{
        Z
        Print 1;
        Print 1;
        W
}
```

Let S and T be two binary semaphores, P(S) & P(T) denotes the wait() operation and V(S) & V(T) denotes the signal() operation on the respective variables. If we want to print '1122112211', then values of X, Y, Z, W will be

A) P(S),P(T),V(S),V(T) when S and T are both initialized to 1
B) P(S),V(T),P(T),V(S) when S and T are initialized to 1 and 0 respectively
C) P(S),V(T),P(T),V(S) when S and T are initialized to 0 and 1 respectively
D) P(S), V(T),P(T),V(S) when S and T are both initialized to 1

Correct Option: C

Solution:

At first S is 0, so P(S) at X will wait. So 2 will not be printed. But T is initialized with 1, So after P(T), 1 will be printed and then V(S) will make S 1, then only process A can execute, but then T is 0, so process B cannot execute. After V(T) at Y, process B can again resume.

4. Which of the following is false?

A. **The run time mapping from virtual to physical address is done by memory management unit.**
B. **The compile time and load-time address-binding methods generate identical logical and physical addresses.**
C. **First fit strategy suffers for memory allocation suffers from external fragmentation but Best fit strategy does not.**
D. **An advantage of paging is the possibility of sharing common code.**

Correct Option: D

Explanation: Both first fit and best fit strategy suffers for memory allocation suffers from external fragmentation**.**

5. Consider a system with a 16KB memory. The sequence of processes loaded in and leaving the memory are given in the following.

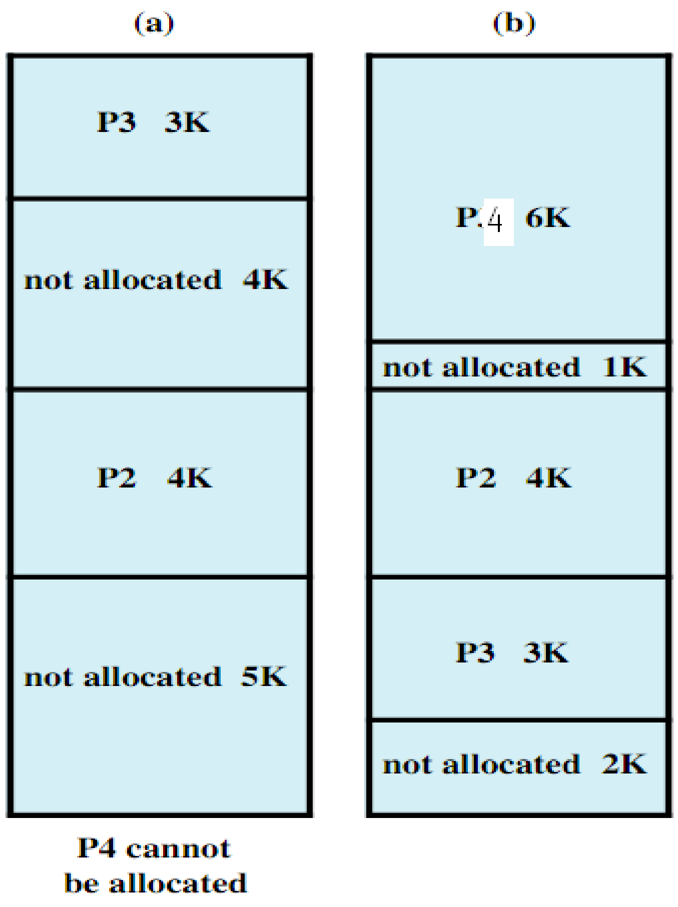
P1 7K loaded
P2 4K loaded
P1 terminated and returned the memory space
P3 3K loaded
P4 6K loaded
Assume that when a process is loaded to a selected "hole", it always starts from the smallest address. E.g. P1 will be loaded in memory from location 0 since the entire memory is one hole. Give the difference between the wasted memory in case of

a. **first fit and**
b. **best fit memory allocation.**

A) 0 B) 2 C) 5 D) 6

Correct Option: D

Explanation:



6. Consider the two statements about thread:-

S: When designing a multithreaded application, you must use synchronization primitives to make sure that the threads do not overwrite each other's registers
R: Threads belonging to the same process share their registers.

A) Both S and R are true and R is the correct reason for S.
B) Both S and R are true but R is not the correct reason for S.
C) S is true but R is false.
D) Both S and R are false

Correct answer: D

Explanation: Registers are kept in the TCB and are private to each thread. Consequently, one thread will not overwrite another thread's register under normal circumstances.

7. Match the following:

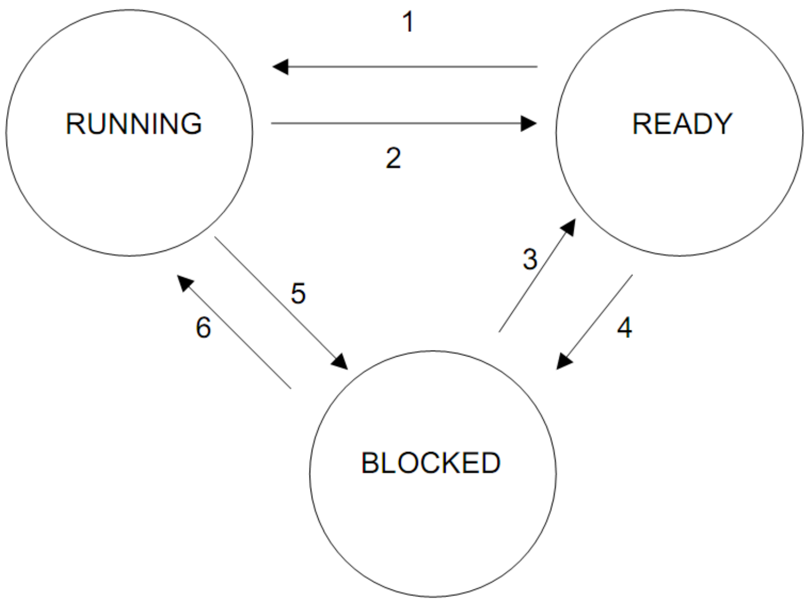
1 . Deadlock prevention	a. Use Banker's Algorithm
2. Deadlock detection	b. Order all resources in the system, and only allow ascending allocations.
3. Deadlock avoidance	c. Check for cycles in the resources of the blocked processes.

A **1-a, 2-c, 3-b** **B** **1-b, 2-a, 3-c**
C **1-b, 2-c, 3-a** **D** **1-c, 2-b, 3-a**

Correct Option: C

Explanation: Refer to ways of deadlock prevention, detection and avoidance.

8. What conditions cause a thread to move between each of the 3 states and what causes each arrow? Label it N/A if it doesn't happen.



a: Process is scheduled and run by the scheduler.
b: Time slice runs out, but process is still wanting to run. Yield().
c: I/O completes, or lock is acquired. Woken up by a semaphore or conditional.
d: Any blocking action. I/O request, lock blocks
e: N/A.

A) 1- a, 2 –c, 3-b, 4-e, 5-d, 6-e
B) 1- c, 2 –b, 3-a, 4-e, 5-e, 6-e
C) 1- a, 2 –b, 3-c, 4-d, 5-e, 6-e
D) 1- a, 2 –b, 3-c, 4-e, 5-d, 6-e

Correct Option: D

Explanation:

Arrow 1: Process is scheduled and run by the scheduler.
Arrow 2: Time slice runs out, but process is still wanting to run. Yield().
Arrow 3: I/O completes, or lock is acquired. Woken up by a semaphore or conditional.
Arrow 4: N/A
Arrow 5: Any blocking action. I/O request, lock blocks.
Arrow 6: N/A

9. Which of the following statements is/are true?

1. Modify bit can be used to minimize the number of page read requests.
2. Segmentation supports user view of memory.
3. The minimum number of frames per process is defined by the architecture.
4. C-scan is designed to provide more uniform waiting time than SCAN algorithm.
5. In low-level formatting, special data structure like maps of free and allocated space are stored onto the disc.
6. Logical level formatting fills the disc with a special data structure (i.e. header, data area..)
for each sector.

A) 2,3,4 B) 2,4,5,6 C) 1,5,6 D) 1,2,4

Correct Option: A

Explanation:

Modify bit can be used to minimize the number of page write requests (i.e. when a page is not modified in memory, there is not need of writing that page to disc while we need to replace that page).
Low level formatting fills the disc with a special data structure (i.e. header, data area..)
for each sector.
In logical-level formatting, special data structure like maps of free and allocated space are stored onto the disc.

10. Two processes, P1 and P2, need to access a critical section of code.

Consider the following synchronization construct used by the processes:

```
/* P1  */
while (true)   {
 w1 = true;
  while (w2 == true);
  /* critical section */
 w1 = false;
}
/* Remainder section */

/*   P2   */
while   (true)    {
 w2 = true;
  while (w1 == true);
/* critical section */
 w2 + false;
}
/* Remainder section */
```

Here, w1 and w2 are shared variables, which are initialized to false. Which one of the following statements is TRUE about the above code?

A) It does not ensure mutual exclusion.
B) It does not ensure bounded waiting.
C) It requires that processes enter the critical section in strict alternation.
D) It does not prevent deadlocks, but ensures mutual exclusion.

Correct Option: D

Explanation:
P1 has control of the critical section provided w1 is true and w2 is false. P2 has control of the critical section provided w2 is true and w1 is false. So if P1 has control it excludes P2 till it completes and vice versa, so mutual exclusion is ensured. This eliminates choice (A).
(B) is false as the time spent by P1 and P2 in their critical sections is controlled or bounded.
(C) is not correct for one can easily see that P1 can use the resource, release it, use it again, release it and so on without P2 ever demanding it.
(D) A deadlock can arise as the assignment to w1 and w2 is not done as an indivisible operation. So when w1 is set to true at the same time w2 can be set to true. This results in endless waiting.

11. For the following disk accesses, compute the absolute difference between the number of head movements for the seeks to disk cylinder: 26 37 100 14 88 33 99 12 for SSTF and SCAN(going up) algorithm.
Assume head is initially positioned over 26.

A)	38	B)	49
C)	59	D)	27

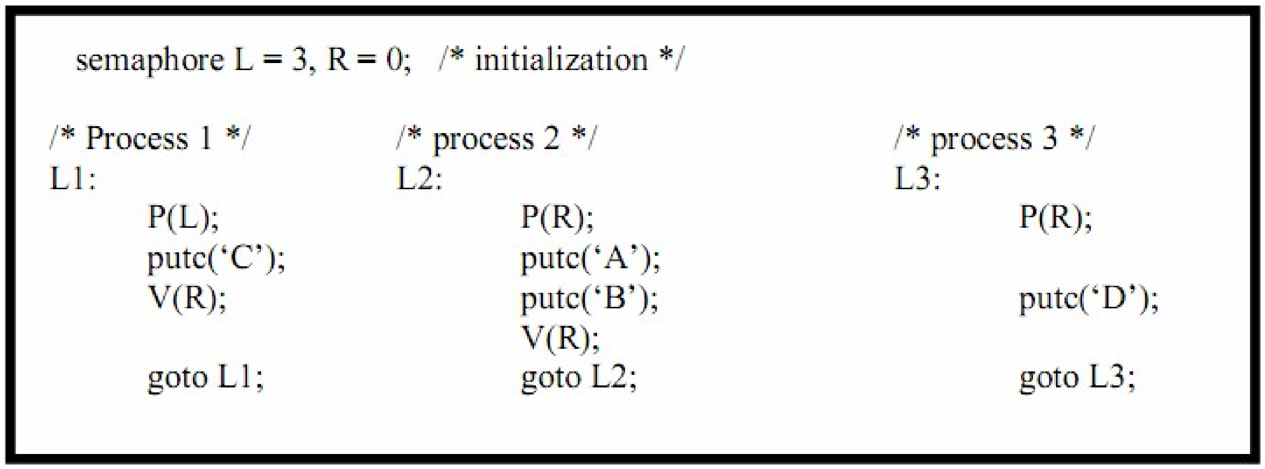
Correct Option: A

Explanation:-

SSTF – 7 + 4 + 23 + 2 + 76 + 11 + 1 = 124
SCAN – 7 + 4 + 51 + 11 + 1 + 86 + 2 = 162

Common data question (12 & 13)

In the following code, three processes produce output using the routine “putc” and synchronize using two semaphores “L” and “R.”



```
semaphore L = 3, R = 0;   /* initialization */

/* Process 1 */          /* process 2 */            /* process 3 */
L1:                      L2:                        L3:
        P(L);                    P(R);                      P(R);
        putc('C');               putc('A');
        V(R);                    putc('B');                 putc('D');
                                 V(R);
        goto L1;                 goto L2;                   goto L3;
```

Question Number: 12

How many D’s are printed when this set of processes runs?

What is the smallest number of A’s that might be printed when this set of processes runs?

A) 1, 1 B) 3, 0 C) 3,3 D) 0,0

Correct Option: B

Explanation:

D can only be printed if process 3 runs. Now notice L=3 and R=0 initially. As L=3, process 1 can run three times and it will increase the value of R to 3 by V(R) operation. Now process 2 or process 3 can execute P(R) operation and execute. Even if process 2 executes first by P(R) operation first, then also it will increase R value to the same value as before because of the last V(R) operation at the end of process 2. So, in any case, ‘D’ will be printed thrice.

‘A’ may not be at all printed. As mentioned, when process 1 has increased the value of R, either process 2 or 3 can run. Now consider the case when process 3 runs. It will invoke P(R) operation and print ‘D’. But, there is no V(R) operation at the end of process 3. So, there is no guarantee that process 2 will run or not.

Question Number: 13

Is CABABDDCABCABD a possible output sequence when this set of processes runs?

Is CABACDBCABDD a possible output sequence when this set of processes runs?

A) yes, yes B) yes, no C) no, yes D) no, no

Correct Option: C

Explanation:

<u>CABABDDCABCABD</u>

Process1: print C ; L=2, R=1 { L & R values represent the values of the variable after the corresponding process has executed}

Process 2: print A, B ; L=2, R=1

Process 2: print A, B ; L=2, R=1

Process 3: print D ; L=2, R=0

To print ‘D’ again, process 3 should again execute, but it will not execute as R=0, so, this is not as possible output.

<u>CABACDBCABDD</u>

Process 1: print C ; L=2, R=1

Process 2: print A, B ; L=2, R=1

Process 2: print A ; L=2, R=0

Process 1: print C ; L=1, R=1

Process 3: print D ; L=1, R=0 .. notice process 2 and 1,3 can execute simultaneously

Process 2: print B ; L=1, R=1

Process 1: print C ; L=0, R=2

Process 2: print A,B ; L=0, R=2

Process 3: print D ; L=0, R=1

Process 3: print D ; L=0, R=0

Linked Data Question (14 & 15):

Consider the following workload:

Process	Burst Time	Priority	Arrival Time
P1	50 ms	4	0 ms
P2	20 ms	1	20 ms
P3	100 ms	3	40 ms
P4	40 ms	2	60 ms

Consider schedule using Shortest Remaining Time(pre-emptive), non-preemptive Priority (a smaller priority number implies higher priority) and Round Robin with quantum 30 ms.

Question Number: 14

Which one has the lowest waiting time?

A) Shortest Remaining Time scheduling

B) Priority scheduling

C) Round-robin scheduling

D) Both SRT and Priority

Correct Option: A

Explanation:

Shortest Remaining Time:

P1	P1	P2	P2	P1	P1	P1	P4	P4	P4	P4	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Non-preemptive Priority:

P1	P1	P1	P1	P1	P2	P2	P4	P4	P4	P4	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Round Robin with quantum of 30 ms:

P1	P1	P1	P2	P2	P1	P1	P3	P3	P3	P4	P4	P4	P3	P3	P3	P4	P3	P3	P3	P3
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

For waiting time,

Shortest Remaining Time: (20+0+70+10)/4 = 25ms.
Explanation: P2 does not wait, but P1 waits 20ms, P3 waits 70ms and P4 waits 10ms.

Non-preemptive Priority: (0+30+10+70)/4 = 27.5ms
Explanation: P1 does not wait, P2 waits 30ms until P1 finishes, P4 waits only 10ms since it arrived at 60ms and it is scheduled at 70ms. P3 waits 70ms.
Round-Robin: (20+10+70+70)/4 = 42.5ms

Question Number: 15

Suppose, among these three scheduling, scheduling A1 has minimum waiting time(T1) and scheduling A2 has maximum waiting time(T2).
What is the difference between T1 and T2?

A) 11.5 B) 14.5 C) 17.5 D) 21.5

Correct Option: C

Explanation:
T1 = 25, T2=42.5