Q.1) Max Marks: 1

Consider a demand-paging system with a paging disk that has average access and transfer time of 20 milliseconds. Addresses are translated through a page table in main memory, with an access time of 1 microsecond per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time, we have added an associative memory that reduces access time to one memory reference if the page-table entry is in the associative memory. Assume that 80 percent of the accesses are in the associative memory and that, of those remaining, 10 percent (or 2 percent of the total) cause page faults. What is the effective memory access time in milliseconds?

A 0.2 ms

B 1.0 ms

C 0.4 ms — Correct Option

Solution: (C)

Answer: C

Explanation:

EMAT= 80% of Reference from associative memory + 18% form page table + 2% from page fault

=0.8(1 μs) + 0.18(2 μs) + 0.02(20000 μs + 2ns)

=401.2 μsec or 0.4ms

D 0.8 ms

Q.2) Max Marks: 1

Given the following pairs of scheduling algorithms:

1. **Multi-level queue and Multi-level feedback queue.**
2. **Shortest Remaining Time First and Priority Scheduling algorithms.**

Which of the above pairs of scheduling algorithms is a subset of the other?

A 1 only

B 2 only — Correct Option

Solution: (B)

Explanation:

Multilevel queue scheduling and Multilevel Feedback Queue are different algorithms. Neither behaves like the other. So, 1 is false.

In Shortest Remaining Time First, the priority is given to the process with the shortest remaining burst time. So it is also a kind of priority scheduling algorithm.

In priority scheduling algorithm generally the priority are given earlier based on user defined parameter whereas in SRTF, the priority is on the basis of the burst time. That is why 2 is correct.

C Both 1 and 2

D Neither 1 nor 2.

Q.3) Max Marks: 1

Consider the solution to readers writers process by using Binary Semaphore A and C.

Reader Process	Writer Process
1. Wait(A)	1. Wait(C)
2. rc = rc + 1	2. Writing is performed
3. if(rc == 1) wait (C)	3. Signal(C)
4. Signal(A)	
5. Reading is performed	
6. Wait(A)	
7. rc = rc - 1	
8. if(rc == 0) signal(C)	
9. Signal(A)	

The semaphore A is the mutual exclusion semaphore initialized to 1. C is initialized to 1 and rc is a variable which is used to count the readers entering

the code.

Consider the following statements:

S1: Removing code lines 1 and 4 from readers process will not affect the solution.

S2: Inserting wait(A) before code line 1 and signal(A) after code line 3 of writers process will lead to a deadlock free solution.

S3: Removing code lines 6 and 8 from readers process will not affect the solution.

How many of the statements are correct?

A Only S2

B Only S1 and S2

C Only S1 and S3

D None of these — Correct Option

Solution: (D)

Answer: D

Explanation:

S1: False, because if it these lines are removed then rc value will become inconsistent as more than 1 reader can access the rc value at the same time

S2: False, because run reader from line1 to line 5. Run modified writer having wait(A) and signal(A), the writer will run wait(A) and will wait on wait(c) and reader is waiting on wait(A). Hence, deadlock.

S3: False, reason same as in s1, more than 1 reader can change the rc value.

Q.4) Max Marks: 1

```
int main()
{
   if (fork() && (!fork())) {
      if (fork() || fork()) {
         fork();
      }
   }
   print("9 ");
   return 0;
}
```

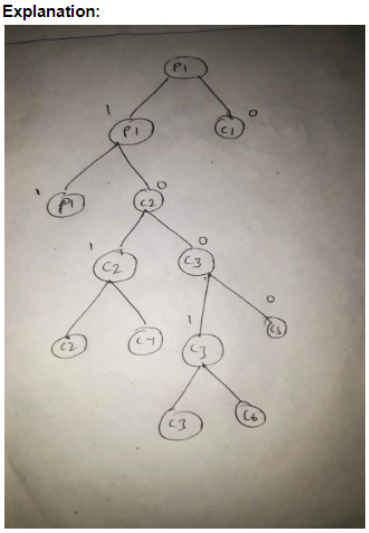
It will print '9' ____ number of times.

Correct Answer

Solution: (7)

Answer: 7

Explanation:



Firstly , fork will be called on p1 and it will return one parent p1 and one child c1 process . parent will contain some +ve value (let it be 1) and child will contain 0.

Now there is and in between two forks and the property of and is if the first value will be 1 then only we will execute other else we will not execute other. So as p1 is 1, we will call fork for only p1 not for c1. And as a result p1 will give two process parent(p1) and child (c2) .

Now as it's **not(fork),** so c2 will return not 0 i.e, 1 and p1 will return not 1 i.e., 0 and so c2 will continue on next line as it will return 1.

Now c2 will give two process parent (c2) and child (c3) and as it is or so if the first condition is true we will not execute 2nd condition, so c2 will directly go to next line and c3 as it returned 0 will execute 2nd condition fork.

Print statements are the leaves. So, finally we will have seven 9's printed.

Q.5) Max Marks: 1

Which of the following options are incorrect:

In priority scheduling algorithm,

A **CPU can be allocated to the process with highest priority**

B **CPU can be allocated to the process with lowest priority**

C **CPU can not be allocated to processes with equal priority.** Correct Option

Solution: (C)
Explanation:
In priority scheduling, in case of equal priorities of two or more processes, there will be a tie-break mechanism (like least process id number or Arrival time of the processes). Thus, equal priority processes can be scheduled.

D **None of the above**

Q.6) Max Marks: 1

Consider a system with processes P0, P1,P2, ... P99, P100, each process requires maximum of 4 resources. System has allocated 2 resources to each process.
The minimum number of resources should release such that above system is deadlock free is _____

Correct Answer

Solution: (102)

Answer: 102

Explanation:

For a system to be deadlock free,
sum of max need of processes < No. of processes + No. of resources ..... (i)

Here system has allocated 2 resources to each process , and Total number of process = 101

So, max. need of processes = 101 * 2 = 202

Let number of resources = R

So from (i) we get, 202 < 101 + R

101 < R

That means the minimum number of resources should be 102 .

Q.7) Max Marks: 1

In a 32-bit machine we subdivide the virtual address into 4 segments as follows:

10-bit	8-bit	6-bit	8 bit

We use a 3-level page table, such that the first 10-bit are for the first level and so on.

What is the size of a page table for a process that has 256K of memory starting at address 0 (in Bytes)?

Assume PTE size is 2 Bytes.

Correct Answer

Solution: (4608)
Using the subdivision above, the first level page table points to 1024 2nd level page tables, each pointing to 256 3rd page tables, each containing 64 pages.

The program's address space consists of 1024 pages, thus we need we need 16 third-level page tables. Therefore we need 16 entries in a 2nd level page table, and one entry in the first level page table.

Therefore the size is: 1024 entries for the first table, 256 entries for the 2nd level page table, and 16 3rd level page table containing 64 entries each.

Assuming 2 bytes per entry, the space required is 1024 * 2 + 256 * 2 (one second-level paget table) + 16 * 64 * 2 (16 third-level page tables) = 4608 bytes.

Q.8) Max Marks: 1

Say a process A has 3 user level threads and a process B has 4 kernel-level threads. Consider while process A is running in CPU, process B is waiting in ready queue. If one of the threads in A is blocked then find the status of A threads and B threads?

A. All A threads are blocked and all B threads are blocked.

B. All A threads are blocked and B threads are not blocked. — Correct Option

Solution: (B)
Answer: B
Explanation:
Blocking is one main problem of user level thread. Here, if one thread is blocked then all of them in it gets blocked but in kernel level thread case is not the same. A is in user level. So A is blocked whereas B is not. Therefore, B is the answer.

C. All B threads are blocked and A threads are not blocked.

D. None of these.

Q.9) Max Marks: 1

Consider the dining philosophers problem and which of the below statements are TRUE to synchronize philosophers problem. Assume there are N philosophers.
S1: (N-1) philosophers should take the left fork and later right fork, and the remaining one philosopher should take right fork first and later left fork.
S2: Odd philosophers should follow left to right and even philosophers should follow right to left.
S3: Odd and even philosophers should follow the left to right fork.
Which of the above are true?

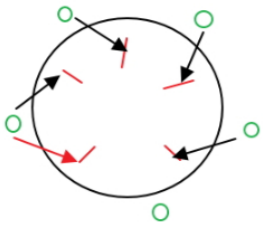
A. S1 and S2 — Correct Option

Solution: (A)
Answer: A
Explanation:

1) is correct

Because if (n-1) philosopher first take left fork fist then right fork, then remaining 1 fork can take a philosopher as a right fork, who already got a left fork

So, 1 philosopher satisfies condition of eating and he eats. And if 1 one satisfies condition, others also will satisfy. So no deadlock.



Similarly, 2) also satisfies condition
So, 1 and 2 are correct

B. Only S1

C. Only S3

D. S1 and S3

Q.10) Max Marks: 1

```
#include <stdio.h>

int main(void) {
   int i=0;
   printf("AppliedCourse\n");
   for(i=1;i<=3;i++)
   {
      if(fork()==0)
         printf("*");
   }

   return 0;
}
```

How many times will 'AppliedCourse' be printed? And how many times will * be printed?

A. 1 & 8

B 1 & 13

C 1 & 7 Correct Option

Solution: (c)

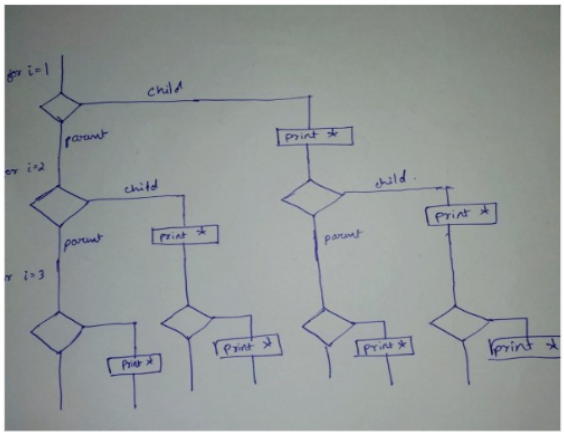
Answer: C

Explanation:

The number of "AppliedCourse" printed will be **1** time which will be at the beginning

The number of "*" printed will be $2^n-1 = 2^3-1 = \mathbf{7}$

The structure will be



where the diamond represents if condition.

D 2 & 12

Q.11) Max Marks: 2

Given m frames, p references and n distinct page numbers, which of the following statement(s) is/are true?

S1: The lower bound on the number of page faults is n.

S2: The upper bound on the number of page faults is p.

A **Only S1 is true.**

B **Only S2 is true.**

C **Both S1 and S2 are true.** Correct Option

Solution: (c)
S1: True
n - these are compulsory misses. This could occur if page replacement is perfect, or there is more memory than needed (m > n).

S2: True
p - you could potentially miss on every page. This could occur if there is very little memory (m = 1) or if page replacement is particularly bad.

D **None of the above**

Q.12) Max Marks: 2

Consider the following table showing the arrival time burst time before io time and followed by CPU time again

Process	AT	Execution time		
		CPU	IO	CPU
1	0	4	3	2
2	1	1	4	1
3	4	3	1	1

Using srtf scheduling policy, the average completion time is _____ units.

Correct Answer

Solution: (10)

Answer: 10

Explanation :

Here we need to keep two things in mind :

a) As we know in "I/O wait state" , there are more than processes possible at the same time. Whereas in "running" state , we can have only one process . Hence if more than 1 processes are there for performing I/O and hence consume I/O time , they can do so simultaneously which means if more than one process have I/O time to execute , then both can do so simultaneously. Whereas for CPU time, we need CPU scheduling.

b) If two processes have the same CPU burst time as well as same arrival time, then the process having less process-id is given priority.

Thus keeping these points in mind, let us find the average completion time :

Q.13) Max Marks: 2

Suppose we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaces page is modified. Memory access time is 100 nanoseconds.

Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds? [Correct upto 6 decimal places].

Correct Answer

Solution: (0.000006)

Therefore,

```
EAT = 0.2micro seconds = (1-P) * 0.1 microseconds +(0.3P) *8milliseconds +(0.7P) * 20 milliseconds

   0.1 =  -0.1P +2400 P + 14000 P
   0.1 ~ 16,400 P
   P ~ 0.000006
```

Q.14) Max Marks: 2

Consider a computer system with single level paging architecture having logical address of 'd'-bit. The memory is byte addressable. If the page table entry size is '32'-bits. What must be the optimal page size P by minimizing the page table overhead and internal fragmentation in paging?

A **3√(2^(d+2))**

B **2√(3^(d+2))**

C **2√(2^(d+1))** Correct Option

Solution: (C)

Answer: C

Explanation:

Two considerations while choosing the size of page

1. Larger Page Size : Smaller Page table but larger internal fragmentation

2. Smaller Page size : Larger Page table which may not be able to fit in one frame and less unused pages in memory

So ultimately we want an optimal size of page so that both we can manage both the things together.

Let Page size is S, size of process is P and entry size is e.

Page Table Size = (P/S)* e

Average amount of internal fragmentation is S/2.

So in order to minimize this quantity differentiate it .

d/dS((P/S) * e + (S/2))=0

-(1/S^2) Pe + 1/2 = 0

S= √(2Pe)

Now, to answer the question:

P = 2d, e = 32bit = 4B

S = √(2*(2^d)*4)

Hence answer is S= 2√(2^(d+1))

D **None of the above**

Q.15) Max Marks: 2

Consider the following page reference string:
g,e,b,c,e,c,a,g,d,a,g,b

With four frames, how many page faults would occur with the optimal page replacement algorithm?

Correct Answer

Solution: (6)

Time	1	2	3	4	5	6	7	8	9	10	11	12
RS	g	e	b	c	e	c	a	g	d	a	g	b
F0	g	g	g	g	g	g	g	g	g	g	g	g
F1		e	e	e	e	e	a	a	a	a	a	a
F2			b	b	b	b	b	b	b	b	b	b
F3				c	c	c	c	c	d	d	d	d
Page Fault?	x	x	x	x	-	-	x	-	x	-	-	-

At time 7, both **e** and **c** could be replaced. The resulting page faults will be the same
Total Page Faults: 6

Q.16) Max Marks: 2

Suppose cylinder head is at 100 currently. The queue of requests are : 30,85,90,100,105,110,135,145. SSTF is used.
Find the #seeks required to service cylinder 90.
Also, find #requests served before serving cylinder 90.

A **20 & 2**

B **10 & 1**

C **30 & 3** Correct Option

Solution: (C)

Answer: C

Explanation:
SSTF = shortest seek time first (less head movement)
Initially head is at 100. So head movements are:
100->100->105->110->90->85->135->145->30.

Thus, #seek required to service cylinder 90 = |100-105|+|105-110|+|110-90| = 5 +5 +20 = 30.

Request served before serving cylinder 90=100,105,110.

Therefore #requests served before serving cylinder 90 = 3.

D **40 & 4**

Q.17) Max Marks: 2

You are asked to compare the storage needed to keep track of free memory using a bitmap versus using a linked list.

Assume the 128-MB memory is allocated in units of n bytes. For the linked list, assume that memory consists of an alternating sequence of segments and holes, each 64 KB. Also assume that each node in the linked list needs a 32-bit memory address, a 16-bit length, and a 16-bit next-node field. For values larger than what is a bitmap better than linked list?

A **1 KB** Correct Option

Solution: (A)

A : The bitmap needs 1 bit per allocation unit. With $2^{27}/n$ allocation units, this is $2^{24}/n$ bytes. The linked list has $2^{27}/2^{16}$ or 2^{11} nodes, each of 8 bytes, for a total of 2^{14} bytes. For small n, the linked list is better. For large n, the bitmap is better. The crossover point can be calculated by equating these two formulas and solving for n. The result is 1 KB. For n smaller than 1 KB, a linked list is better. For n larger than 1 KB, a bitmap is better.

B 12 KB

C 8 KB

D 4 KB

Q.18) Max Marks: 2

Consider a file currently consisting of 100 blocks. Assume that the file control block and the index block in the case of indexed allocation is already in memory and assume that updates to these do not need to be written back to disk.

Calculate how many disk I/O operations (i.e., a block read or a block write) are required for contiguous allocation strategy if the block is added at the beginning.

In the contiguous-allocation case, assume that there is not room to grow at the beginning but there is room to grow at the end. Also assume that the block information to be added is stored in memory.

Correct Answer

Solution: (201)

If you add at the beginning, you need to move all blocks, which requires you to read each block and write it back somewhere else. That's 100*2=200.

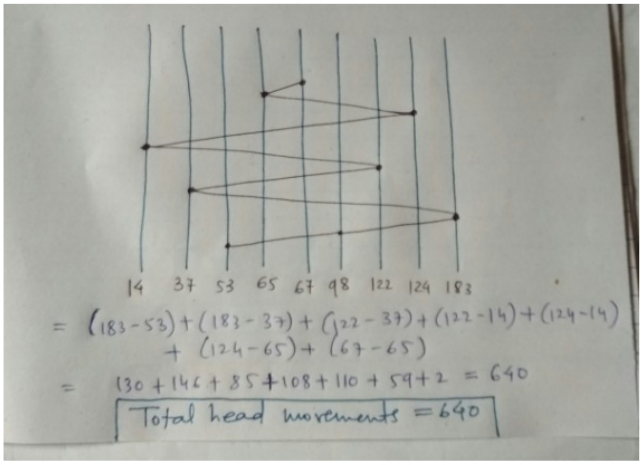
Finally, you still need to write the block you want to add. So total is 201.

Q.19) Max Marks: 2

For FCFS disk scheduling, request queue is 98, 183, 37, 122, 14, 124, 65, 67.
And head starts at 53. The total head movement is _____.

Correct Answer

Solution: (640)



Q.20) Max Marks: 2

A demand paging system has page fault service time as 125 time units if page is not dirty and 400 times units of page fault service time if it is a dirty page.
Memory access time is 10 time units. The probability of a page fault is 0.3. In case of page fault, the probability of page being dirty is P. It is observed that average access time is 50 time units. Then, the value of P is _____? [up to four decimal places]

Correct Answer

Solution: (0.0004)

Answer: 0.0004
Explanation:

Page fault rate = 0.3

Hence page hit rate = 0.7

Memory access time = 10 time units

Page fault service time = 125 time units

Let probability of page being dirty = p

Given effective access time = 50 ns

Thus E.M.A.T = 0.7 * [Memory access time] + 0.3*[p * (400 PFST) + (1-p) * PFST]

==> 0.7 * 10 + 0.3 *[400*125*p + 125 - 125p] = 50

==> 7 + 0.3 * [50000p - 125p + 125] = 50

==> 0.3 * [49875p + 125] = 43

==> *p = 0.0004 [correct upto 4 decimal places]*

close