**QUESTIONS AND ANSWERS**

***QUES:- Linux operating system uses***

(A) Affinity Scheduling(B) Fair Preemptive Scheduling(yes) (C) Hand Shaking

(D) Highest Penalty Ratio Next

**Explanation:**

Linux uses two process-scheduling algorithms:

1. A time-sharing algorithm for fair preemptive scheduling between multiple processes

2. A real-time algorithm for tasks where absolute priorities are more important than fairness

***QUES:- How large can a file be with 1KB blocks?***

• Single indirect block:– Assuming 32-bit addresses, we have 4 bytes per block pointer, so 1 KB/4 = 256 blocks So … 256 \* 1 KB = 256 KB

• Double-indirect block: – 256 \* 256 \* 1 KB = 64 MB

• Triple Indirect block: – 256 \* 256 \* 256 \* 1 KB = 16 GB

• Total: ~16 GB

***QUES:- How large can a file be with 4KB blocks?***

• Single indirect block:– Assuming 32-bit addresses, we have 4 bytes per block pointer, so 4 KB/4 = 1024 B blocks. So … 1024 \* 1 KB = 1 MB

• Double-indirect block: 1024 \* 1024 \* 1 KB = 1 GB

• Triple Indirect block:–1024 \* 1024 \* 1024 \* 1 KB = 1 TB

• Total: ~1 TB

***QUES:- The aging algorithm with a=0.5 is used to predict run times. The previous four runs from oldest to most recent are 40, 20, 20 and 15 msec. The prediction for the next time will be:***

(A) 15 msec (B) 25 msec (yes) (C) 39 msec (D) 40 msec

**Exp:-** If take all four previous run times into consideration, the prediction

is (((40 + 20) / 2 + 40) / 2 + 15) / 2 = ((30 + 40) / 2 + 15) /2 = (35 + 15) / 2 = 25

**Or** if take only two previous run times into consideration, the prediction is (40 + 15) /2 = 27.5



***How much space will be required to store the bit map of a 1.3 GB disk with 512 bytes block size?***

a)332.8 KB(yes) b)83.6 KB c)266.2 KB d)256.6 KB

**Expl :-** We need a bit for each block. Number of blocks = disk size/ block size = 1.3 GB / 512 B

more expl get figure from http://internetnotes.in/q49-p2-d13-how-much-space-will-be-required-to-store-the-bit-map-of-a-1-3-gb-disk-with-512-bytes-block-size/

***QUES:- How much space will be required to store the bit map of a 1.3 GB disk with 512 bytes block size ?***

(A) 332.8 KB|(Ans)     (B) 83.6 KB (C) 266.2 KB     (D) 256.6 KB

**Explanation:**block size=29bytes (512 bytes)

disk size=1.3×230bytes (1.3 GB)

1.3 GB = 1.3 \* 1024 \* 1024 \* 1024 bytes. = 1.3 \* 230 bytes

n=1.3×230/29=1.3×221 bits=1.3×210x28x23 bits=1.3×28 Kilo Bytes

=332.8 KB

***QUES:- There are n processes in memory. A process spends a fraction p of its time waiting for I/O to complete. The CPU utilization is given by :***

(A) pn (B)1-pn(yes) (C)(1-p)n  (D)1-np

***QUES:- Files that are related to input/output and are used to model serial I/O devices such as terminals, printers and networks are called :***

(A) regular files (B) character special files

(C) directories (D) block special files(yes)

***QUES:- What is the most appropriate function of Memory Management Unit (MMU) ?***

(A) It is an associative memory to store TLB

(B) It is a technique of supporting multiprogramming by creating dynamic partitions

(C) It is a chip to map virtual address to physical address(yes)

(D) It is an algorithm to allocate and deallocate main memory to a process

***QUES:- Function of memory management unit is :***

(A) Address translation(yes) (B) Memory allocation

(C) Cache management (D) All of the above

***QUES:- In \_\_\_\_\_ method, the word is written to the block in both the cache and main memory, in parallel.***

(1) Write through(YES) (2) Write back (3) Write protected (4) Direct mapping

***QUES:- An Operating System (OS) crashes on the average once in 30 days, that is, the Mean Time Between Failures (MTBF) = 30 days. When this happens, it takes 10 minutes to recover the OS, that is, the Mean Time To Repair (MTTR) = 10 minutes. The availability of the OS with these reliability figures is approximately?***

(1) 96.97% (2) 97.97% (3) 99.009% (4) 99.97%

**Explanation:-**

MTBF = 30 days

MTTR = 10 min = 0.17 hours = 0.007 day

MTTF = 29.993 days

Availability is 29.993/30 \* 100 = 99.97%

***QUES:- Consider a system which have ‘n’ number of processes and ‘m’ number of resource types. The time complexity of the safety algorithm, which checks whether a system is in safe state or not, is of the order of:***

(A) O(mn) (B) O(m2n2) (C) O(m2n) (D) O(mn2)Yes

**QUES:- *A memory management system has 64 pages with 512 bytes page size. Physical memory consists of 32 page frames. Number of bits required in logical and physical address are respectively*** *:*

(1) 14 and 15 (2) 14 and 29 (3) 15 and 14(yes) (4) 16 and 32

**Explanation:-**

we know that Number of pages = virtual memory space / page size.

and Number of frames = physical memory space / frame size.

and page size is equal to frame size.

According to question and given data:

virtual memory space = Number of pages \* page size

i.e. virtual memory space = 64 \* 512 Bytes

virtual memory space = 26 \* 29 Bits

i.e. = 215 Bits

So, 15 bits are required for virtual memory space.

physical memory space = Number of frames \* frame size.

physical memory space = 32 \* 512 Bytes

physical memory space = 25 \* 29 Bits

i.e. = 214 Bits

So, 14 bits are required for virtual memory space.

**QUES:-*Consider a disk queue with requests for I/O to blocks on cylinders 47, 38, 121, 191, 87, 11, 92, 10. The C-LOOK scheduling algorithm is used. The head is initially at cylinder number 63, moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is:***

(A) 346 (B) 165(yes) (C) 154 (D) 173

**Explanation:** The head movement would be :

63 => 87 24 movements

87 => 92 5 movements

92 => 121 29 movements

121 => 191 70 movements

191 --> 10 0 movement

10 => 11 1 movement

11 => 38 27 movements

38 => 47 9 movements

Total head movements = 165

***Consider a disk queue with request for input/output to block on cylinders 98, 183, 37, 122, 14, 124, 65, 67 in that order. Assume that disk head is initially positioned at cylinder 53 and moving towards cylinder number 0. The total number of head movements using Shortest Seek Time First (SSTF) and SCAN algorithms are respectively***

(A) 236 and 252 cylinders (B) 640 and 236 cylinders

(C) 235 and 640 cylinders (D) 235 and 252 cylinders

Answer: 236 and 208 cylinders

**Explanation:-**

SSTF-

Initial head position =53

The closest queue to initial head position=65

head moves from 53 to 65=12

head moves from 65 to 67=2

head moves from 67 to 37=30

head moves from 37 to 14=23

head moves from 14 to 98=84

head moves from 98 to 122=24

head moves from 122 to 124=2

head moves from 124 to 183=59

Total head movement=236

SCAN-

53---37-----14----0-------65----67-----98---122----124----183

Acc to SCAN 53 to 37 then to 14 and then to 0 (16+23+14=53) now direction will be reversed so 0 to 65 then to 67 then to 98,122,124,183 ( 65+2+31+24+2+59=183) total head movements=183+53=236

***A specific editor has 200 K of program text, 15 K of initial stack, 50 K of initialized data, and 70 K of bootstrap code. If five editors are started simultaneously, how much physical memory is needed if shared text is used ?***

(A) 1135 K (B) 335 K(yes) (C) 1065 K (D) 320 K

**Explanation:-** 5 editor works simultaneously. Each editor must have separate code in each. But they could share initial stack, initialized data and bootstrap code.So, total memory required is 200+15+50+70=335

***Object modules generated by assembler that contains unresolved external references are resolved for two or more object module by a/an***

(A) Operating system (B) Loader (C) Linker(yes) (D) Compiler

***Which command allows you to view your file 24 lines at a time ?***

(A) More(yes) (B) Cat (C) Pg (D) None of the above

***Block or Buffer caches are used to***

(A) improve disk performance (B) handle interrupts

(C) increase the capacity of main memory

(D) speed up main memory Read operations(yes)

***Which of the following can be accessed by transfer vector approach of linking ?***

(A) External data segments (B) External subroutine(yes)

(C) Data located in other procedure (D) All of the above

***A file organization component VSAM file is***

(A) Relative records data set (B) Keyed sequential data set

(C) Entry sequential data set (D) All of the above(yes)

**Explanation:-** VSAM stands for Virtual Storage Access Method. It is also relative, keyed sequential data set.

***Virtual memory is***

(A) related to virtual reality (B) a form of ROM (C) a form of RAM(yes) (D) None of the above

***A relationship between processes such that each has some part (critical section) which must not be executed while the critical section of another is being executed, is known as***

(A) Semaphore (B) Mutual exclusion(yes) (C) Multiprogramming (D) Message passing

***Thrashing can be avoided if***

(A) pages, belonging to the working set of the programs, are in main memory(yes)

(B) speed of CPU is increased

(C) speed of I/O processor is increased

(D) all of the above

***The system administrator can hunt for all files owned by ‘‘Kamal’’, and larger than 2000 blocks; by;***

(A) grep / – user Kamal – a + 2000 – print

(B) find / – user Kamal + 2000 – print

(C) find / \ C-user Kamal – a – size + 20000 \) – print(yes)

(D) none of the above

***Which command is used to change protection mode of files starting with the string emp and ending with 1, 2 or 3?***

(A) chmod u x emp [1 –3](yes)

(B) chmod 777 emp

(C) chmod u r ??? emp

(D) chmod 222 emp?

***A virtual memory based memory management algorithm partially swaps out a process. This is an example of***

(A) short term scheduling (B) long term scheduling

(C) medium term scheduling(yes) (D) mutual exclusion

***The mv command changes***

(A) the inode (B) the inode-number

(C) the directory entry(yes) (D) both the directory entry and the inode

**Explanation:-**

The mv command, say. try x y is not going to change the tile content, the i-node number or other information in the i-node. Only the file name is going to change. The file name is present only in the directory.

***Let the page fault service time be 10 millisecond(ms) in a computer with average memory access time being 20 nanosecond(ns). If one page fault is generated for every 106 memory accesses, what is the effective access time for memory ?***

(A) 21 ns (B) 23 ns (C) 30 ns(yes) (D) 35 ns

given that one page fault for every 106 memory access, so the probability of page fault is p-> 1/106

effective access time would be = p \* page fault service time + (1-p) \* memory access time

1millisecond = 10,00,000 nano second

1 second = 1000 milli second

1 second = 1,00,00,00,000 nano second





***The hit ratio of a Translation Look Aside Buffer (TLB) is 80%. It takes 20 nanoseconds (ns) to search TLB and 100 ns to access main memory. The effective memory access time is \_\_\_\_\_\_.***

(A) 36 ns (B) 140 ns (C) 122 ns (D) 40 ns

Effective access time = hit ratio \* time during hit + miss ratio \* time during miss

hit ratio = 80/100 = 0.8%

EAT = 0.8 \* ( 20 + 100) + 0.2( 20 + 100 + 100)

if no tlb miss occur then search into tlb (20ns) than access the main memory 100ns

if tlb miss occur then searching for tlb (20ns) + 100ns for page table + 100ns for ram

***In a demand paging memory system, page table is held in registers. The time taken to service a page fault is 8 m.sec. if an empty frame is available or if the replaced page is not modified, and it takes 20 m.secs.,if the replaced page is modified. What is the average access time to service a page fault assuming that the page to be replaced is modified 70% of the time ?***

(A) 11.6 m.sec. (B) 16.4 m.sec (C) 28 m.sec. (D) 14 m.sec.

**Explanation:-**

Page fault service time(PFST) = 20 msec( page is modified)

Page replacement frequency = 70% = 0.7

Page not modified = 100 – 70 = 30% = 0.3

Average access time = (0.3 \* 8) + (0.7 \* 20) = 2.4 + 14 = 16.4 m.sec

***For the implementation of a paging scheme, suppose the average process size be x bytes, the page size be y bytes, and each page entry requires z bytes. The optimum page size that minimizes the total overhead due to the page table and the internal fragmentation loss is given by***

1. x/2 (B) (xz)/2 (C) sqrt(2xz)(yes) (D)sqrt(xz)/2

**Application:-**

So for example, if the average segment size were 256K, and the page table entry size were 8 bytes, the optimum page size, to minimise overhead due to page table entries and internal fragmentation, would be sqrt(2 × 256K × 8) = 2048 = 2K.

***Consider a system with a two-level paging scheme in which a regular memory access takes 150 nanoseconds, and servicing a page fault takes 8 milli seconds. An average instruction takes 100 nanoseconds of CPU time, and two memory accesses. The TLB hit ratio is 90%, and the page fault rate is one in every 10,000 instructions. What is the effective average instruction execution time?***

1) 645 ns 2) 1050 ns 3) 1215 ns 4) 1230 ns

***If an instruction takes ‘i’ microseconds and a page fault takes an additional ‘j’ microseconds. The effective instruction time, if on the average a page fault occurs every k instructions, is***

(A) i + j/k(yes) (B) i + j \* k (C) (i + j)/k (D) (i + j) \* k

**Explanation:-**page fault rate=1/k

page hit rate=1 – 1/k

service time=i

page fault service time= I +j

now effective memory access time=1/k\*(i+j)+(1-1/k)\*i

= (i+j)/k+i-i/k

=i/k+j/k+i-i/k=i+j/k

***Some of the criteria for calculation of priority of a process are :***

a. Processor utilization by an individual process.

b. Weight assigned to a user or group of users.

c. Processor utilization by a user or group of processes

In fair share scheduler, priority is calculated based on :

(1) only (a) and (b) (2) only (a) and (c)

(3) (a), (b) and (c)(yes) (4) only (b) and (c)

***One of the disadvantages of user level threads compared to Kernel level threads is***

(1) If a user level thread of a process executes a system call, all threads in that process are blocked.(yes)

(2) Scheduling is application dependent.

(3) Thread switching doesn’t require kernel mode privileges.

(4) The library procedures invoked for thread management in user level threads are local procedures.

***Linking :***

(A) cannot be performed before relocation

(B) cannot be performed after relocation(yes)

(C) can be performed both before and after relocation

(D) is not required if relocation is performed

***A system has 3 processes sharing 4 resources. If each process needs a maximum of 2 units, then***

A) Deadlock can never occur(YES)

B) Deadlock may occur

C) Deadlock has to occur

D) None of these

**EXPLANATION:-**

If the system is deadlocked, it implies that each process is holding one resource and is waiting for one more. Since there are 3 processes and 4 resources, one process must be able to obtain two resources. This process requires no more resources and therefore it will return its resources when done.

Hence,Option(A)deadlock can never occur.

**Explanation:** A condition for deadlock can not occur: P(n-1) + 1 <= R where, P = number of processes ; n = max requirement of each process ; R = Total number of available resources So, if a system has 3 processes sharing 4 resources and each process needs a maximum of 2 units then: P = 3, n = 2 and R = 4 =3(2-1) + 1 <= 4, deadlock can never occur.

***If holes are half as large as processes, the fraction of memory wasted in holes is***

A) 2/3 B) 1/2 C) 1/3(yes) D) 1/5

**EXERCISE REGARDING SHORTEST JOB FIRST**

****







***Consider a disk with 16384 bytes per track having a rotation time of 16 msec and average seek time of 40 msec. What is the time in msec to read a block of 1024 bytes from this disk?***

A) 57 msec B) 49 msec C) 48 msec D) 17 msec

**Explanation1:-**

Transfer time = ((1024 bytes \*16 ms)/ (16384 bytes )) = 1 msec

Rotational letency = 16 msec / 2 = 8 msec

total time = seek + rotational letency + transfer time

= 40 + 8 + 1 = 49 msec

**Explanation2:-**

Time in msec to read a block of 1024 bytes (Access time or Disk Latency) = seek time + average rotational delay + transfer time.

If there are 16384 bytes per track there are 1024/16384 tracks to be read for this block.

Seek time = 40 msec

Rotational delay = 16 msec

Transfer time = (sectors\_read/sectors per rev.) x rotational delay

= (1024/16384) x 16 = 1

average rotational delay = rotational delay/2 = 16/2 = 8

access time = 40 + 8 + 1 = 49 msec

***If a disk has a seek time of 20 msec, rotates 20 revolutions per second, has 100 words per block, and each track has capacity of 300 words. Then the total time required to access one block is***

1) 60 2) 45 3) 50 4) 40

**Explanation 1:-**

Time taken to access one block = seek time + rotational delay + block transfer time

Seek time = 20 ms (given)

Rotational delay = on an average taken to be the time to rotate by half = 1/2 X time for 1 rotati X 1/20 sec s = 25 ms

Block Transfer time = block size / transfer rate = 100 / transfer rate

Now, transfer rate = Track capacity / rotation rate = 300 / ( 1/ 20) = 6000 word per sec = 6 word per ms

Block Transfer time = block size / transfer rate = 100 / 6 ~ 16.67 ms per block

Time taken to access one block = seek time + rotational delay + block transfer time= 20 + 25 + 16.67 = 61.67 ms

**Explanation 2:-**

Time taken to access one block = seek time + rotational delay + block transfer time

Seek time = 20 ms (given)

Rotational delay = on an average taken to be the time to rotate by half = 1/2 X time for 1 rotation = 1/2 X 1/20 seconds = 1/40 s = 25 ms

Block Transfer time = block size / transfer rate = 100 / transfer rate

Now, transfer rate = Track capacity / rotation rate = 300 / ( 1/ 20) = 6000 word per sec = 6 word per ms

Block Transfer time = block size / transfer rate = 100 / 6 ~ 16.67 ms per block

Time taken to access one block = seek time + rotational delay + block transfer time= 20 + 25 + 16.67 = 61.67 ms

**Ques:-Consider three CPU-intensive processes, which require 10, 20 and 30 time units and arrive at times 0, 2 and 6, respectively. How many context switches are needed if the operating system implements a shortest remaining time first scheduling algorithm? Do not count the context switches at time zero and at the end.**

1) 2(yes) 2) 3 3) 5 4) 4

**Explanation :-** Let three process be P0, P1 and P2 with arrival times 0, 2 and 6 respectively and CPU burst times 10, 20 and 30 respectively. At time 0, P0 is the only available process so it runs. At time 2, P1 arrives, but P0 has the shortest remaining time, so it continues. At time 6, P2 arrives, but P0 has the shortest remaining time, so it continues. At time 10, P1 is scheduled as it is the shortest remaining time process. At time 30, P2 is scheduled. Only two context switches are needed. P0 to P1 and P1 to P2

***Consider three CPU intensive processes P1, P2, P3 which require 20, 10 and 30 units of time, arrive at times 1, 3 and 7 respectively. Suppose operating system is implementing Shortest Remaining Time first (preemptive scheduling) algorithm, then \_\_\_\_\_ context switches are required (suppose context switch at the beginning of Ready queue and at the end of Ready queue are not counted).***

(1) 3(yes) 2) 2 3) 4 (4) 5

***Consider the two statements.***

***(a) A network operating system, the users access remote resources in the same manner as local resource.***

***(b) In a distributed operating system, the user can access remote resources either by logging into the appropriate remote machine or transferring data from the remote machine to their own machine.***

***Which of the statement is true?***

1) (a) true, (b) false 2) (b) true, (a) false

3) Both (a) and (b) false (ANS) 4) Both (a) and (b) true

***There are three processes P1, P2 and P3 sharing a semaphore for synchronising a variable. Initial value of semaphore is one. Assume that negative value of semaphore tells us how many processes are waiting in queue. Processes access the semaphore in following order:***

P2 needs to access

P1 needs to access

P3 needs to access

P2 exits critical section

Pi exits critical section

The final value of semaphore will be:

A) 0 B)1 C)-1 D)-2

**Explanation:-**

Explanation: Initial value of semaphores S=1

P2 needs to access decreases semaphore by 1, new value will be 0 (no one is waiting)

P1 needs to access decreases semaphore by 1, new value will be -1 (one process is waiting)

P3 needs to access decreases semaphore by 1, new value will be -2 ( 2 process are waiting)

P2 exits critical section increases semaphore by 1, new value will be -1 ( one process is waiting)

P1 exits critical section increases semaphore by 1, new value will be 0 ( no process is waiting)

So, option (A) is correct

**Consider a disk with a sector size of 512 bytes, 2000 tracks per surface, 50 sectors per track, five double-sided platters, and average seek time of 10 msec.**

**1. What is the capacity of a track in bytes? What is the capacity of each surface?**

**What is the capacity of the disk?**

**2. How many cylinders does the disk have?**

**3. Give examples of valid block sizes. Is 256 bytes a valid block size? 2048? 51200?**

**4. If the disk platters rotate at 5400 rpm (revolutions per minute), what is the**

**maximum rotational delay?**

**5. If one track of data can be transferred per revolution, what is the transfer rate?**

**Answer**

1. bytes/track = bytes/sector × sectors/track = 512 × 50 = 25600 bytes = 25600 / 1024 = 25Kbytes/surface = bytes/track × tracks/surface = 25K × 2000 = 50, 000K

bytes/disk = bytes/surface× surfaces/disk = 50, 000K × 5 × 2 = 500, 000K

2. The number of cylinders is the same as the number of tracks on each platter,

which is 2000.

3. The block size should be a multiple of the sector size. We can see that 256 is not a valid block size while 2048 is. 51200 is not a valid block size in this case because block size cannot exceed the size of a track, which is 25600 bytes.

4. If the disk platters rotate at 5400rpm, the time required for one complete rotation, which is the maximum rotational delay, is (1/rpm) \* 60 = 1/5400× 60 = 0.011seconds. The average rotational delay is half of the rotation time, 0.006 seconds.

5. The capacity of a track is 25K bytes. Since one track of data can be transferred

per revolution, the data transfer rate is 25K/0.011= 2, 250Kbytes/secon