International Institute of Information Technology, Hyderabad
(Deemed to be University)

CS3.301: Operating Systems and Networks- Monsoon 2022

Question cum Answer Booklet

End Semester Examination

Max. Time: 3 Hrs Max. Marks:80

Roll No:_____ Programme: _____ Date of Exam:_____

Room no: _____ Seat No: _____ Invigilator's Signature:_____

1. Identify whether the following statements are TRUE or FALSE. If the statement is FALSE, correct it and justify the corrected sentence. If the statement is TRUE, justify it. Restrict the justification to a few (less than five) sentences. [10*1=10] - **Bhaskar Joshi**

1.1 As compared to CPU-bound process, OS gives higher priority for I/O bound process.

Ans: TRUE; The I/O bound processes require short CPU bursts. To improve the CPU and I/O utilization, OS gives high priority to I/O processes.

1.2 With monitors, it is possible to reduce busy waiting over semaphores.

Ans: FALSE; With monitors it is easy to develop programs for critical section problems. The same kind of busy waiting, if any, exists in both structures.

1.3 It is easier to port micro-kernel operating system from one hardware to another hardware as compared to non-micro-kernel operating system.

Ans: TRUE; Micro-kernel OS contains only essential features of OS. The code size is small. So it is easy to port to another hardware as compared to normal OS.

1.4 The notion of "load balancing" counteracts with the notion of "processor affinity".

Ans: True. Processor affinity: In multi-processor systems, to improve the performance, a process is attached to one CPU and system makes all the efforts to not to snatch the process from the allocated CPU. In case of load balancing, we remove the processes from the heavily loaded processor and allocate them to the lightly loaded processor. In this way, the notion of load balancing counteracts with the notion of "processor affinity".

1.5 Global frame allocation algorithm gives better throughput over local frame allocation algorithm.

Ans: TRUE; Global allocation algorithm can use the unutilized frames of one process to allocate more frames to other process to improve the performance. In this way, based on the locality requirement, the

approach can adjust the frames to meet the demands of the processes. As a result, the memory utilization and throughput could be improved.

1.6 **From the system side, it is easy to provide Session Semantics feature as compared to UNIX semantics feature.**

Ans: FALSE; It is difficult from the system side. Providing session semantics requires the maintenance of several images of file and implementation of the protocols for access by subsequent processes in a consistent manner.

1.7 **Fourier theorem helps to improve the efficiency of the channel/link.**

Ans: TRUE. A channel has limited bandwidth. Fourier theorem allows identification of the required number of major frequency components of the signal so that a signal can be transmitted successfully. As a result bandwidth of the channel can be conserved and more signals can be transamitted in the given frequency range of the channel/link

1.8 **Hierarchical routing results in shorter paths than flat routing**

Ans FALSE. Hierarchical routing results in longer paths than flat routing. In the hierarchical routing, the nodes are divided in to clusters. The routing information has only information about cluster representative. Within the cluster, a different path can be followed. In the flat routing, the path to the destination will formed, which leads to shorter paths.

1.9 **A change in the service provided at layer "k", will impact services at both layers "k-1" and "k+1".**

Ans: FALSE. A change in the service at layer "k" will only effect the services at "k+1" layer. The services of lower layers are not effected.

1.10 **"Leaky bucket algorithm" performs better than the "token bucket algorithm".**

Ans: FALSE. Leaky bucket algorithm provides constant output. But, a token bucket algorithm provides a variable output based on number of tokens. So, a token bucket algorithm, adapts well in the changes in the traffic burst.

2. Answer the following briefly [5*3=15]

2.1 **Suppose that a multi-programmed system has a load of N processes with individual total execution times of $t_1,t_2,\ldots,t_N$. How would it be possible that total execution time could be as small as maximum $(t_1, t_2,\ldots,t_N)$? Karthik**

Ans: In multi programmed system, CPU is given to another process, when the existing process blocks due to I/O. Let "t" be the maximum execution time of one process Pt. While

executing, Pt might have blocked several times and during that time, there is a possibility that, other processes might have finished.

2.2 **Explain the reason why transport protocols use larger size sliding windows as compared to the size of the size of sliding windows of data link layer? Vegi**

Ans:Data link layer is concerned with direct communication. So, the delay in case of data link layer is highly predictable (low variance) and in the order of microseconds. Also, ack. is rarely delayed.

The Transport layer is concerned with the communication between two nodes which are not directly connected. The message should be passed through several intermediate nodes or routers. The delay in TCP has a large variance (milli second) range. As, there will be more unack. Packets, the sliding window size is more.

2.3 **Discuss the following program threats: trap door, virus and denial of service. Aneesh**

Ans: A trapdoor is a secret entrypoint build into a program designed to give someone access to a system without the usual security measures. This can include Trojan horses or trapdoors included in malicious updates.

A virus is a maliciously written computer program designed to alter the way a computer system behaves. Viruses typically self replicate, spread themselves to other computer systems and execute without the users knowledge.

A denial or service attack is an attack on a server where an attacker aims to overload a server by maliciously sending a large number of bogus requests, taking up resources. This aims to disrupt service provided to other legitimate users. Worms or other agents might do this offline too.

2.5 **Explain the tradeoffs between "Go-Back-N" Sliding Window Protocol and "Selective Repeat" Sliding Window Protocol. -Karthik**

Ans:

Trade-off:

Go-back N does not require large number of buffers. Performance suffers.

Selective repeat: Needs buffers. More complex. But, gives better performance.

****

Go-Back N: Receiver only accepts/acks frames that arrive in order:

- Discards frames that follow a missing/errored frame
- Sender times out and resends all outstanding frames

Selective Repeat: Receiver accepts frames anywhere in receive window

- Cumulative ack indicates highest in-order frame
- NAK (negative ack) causes sender retransmission of a missing frame before a timeout resends window

3. **The kernel of a multiprogramming system classifies a program as CPU-bound or I/O-bound and assigns appropriate priority to it. What would be the consequence of a wrong classification of programs for throughput and response times in a multi-programming system ? [5]-Greeshma**

   Ans:

   Normally or usually, the system gives high priority to i/o bound programs. The I/O bound programs will get good response times and good throughput is realized for CPU bound programs.

   Suppose, the system considers CPU bound programs as I/O bound programs and vice versa.

   In such a case the system gives high priority to CPU bound programs. The ready queue will have more I/O bound programs waiting. As a result I/O utilization will decrease. The I/O bound users wait for long time, i.e., they will realize longer long response times. If there is continuous stream of CPU bound processes, the I/O bound process starve for CPU cycles.The throughput of I/O bound process will decrease and the throughput of CPU bound processes will increase.

4. **Given that the monitor construct protects critical section by allowing one process at a time, why you require "c.wait" and "c.signal" operations in the Monitor. How they are different from "wait" and "signal" operations of semaphore?[5] -Vegi**

   Ans:

   There are two kinds of waiting to access a resource. One is mutual exclusion waiting(a fast process has to wait if the buffer is full for writing and, also, a fast process has to wait if the buffer is empty for

reading) to avoid a race condition and the other is conditional waiting to avoid inconsistency. The monitor construct takes care of mutual exclusion waiting. So we do not need "wait" and "signal" constructs of semaphore because by keeping a resource in the monitor the protection is achieved, i.e., there is no danger of more than one process accessing a resource in parallel. However, it does not provide the protection against the danger of violating conditional waiting. For example, a producer process can not overwrite the full buffer or a consumer process can not read empty buffer. To be safe from such actions, "c.wait" and "c.signal" operations are provided in the monitor construct. The usual "wait" and "signal" operations do not provide such safety. So, "c.wait" and "c.signal" operations of monitor construct are different from "wait" and "signal" operations of semaphore.

5. **In addition to hardware support such as page table and secondary memory what kind of software support is needed to implement demand paging ? Explain clearly. [5] - Karmanjyot**

Ans:

(i) Restart of instruction after page fault.For example consider the following statement: Add the contents of A and B and replace the result in C. This results into the following assembly sequence :Fetch and decode the instruction; Fetch A; Fetch B; Add A and B; Store the sum in C. If the page is faulted if we try to store C, we have to restart the instruction.

(ii) Difficulty occurs if the instruction modifies several locations.For example, using IBM 360/370 MVC (Move character) instruction, we can move 256 bytes from one location to another. Sometimes, source and destination may overlap. If the page fault occurs after partial moving, we can not redo the instruction, if regions overlap.

As a solution, we can use a micro code to access both ends of blocks. If page fault is going to occur, it will occur. Or use temporary registers to hold the values of temporary registers. If a page fault occurs old values are written back to memory, restoring the memory state to before the instruction was started.

(iii)Difficulty occurs in machines that use special addressing modes. Uses register as a pointer. For example consider Auto-increment and auto decrement instructions. Auto increment means increment after using and auto-decrement means decrement before using; Consider the instruction MOV (R2)+, -(R3); If the page fault occurs while storing in R3, we have to restart the instruction by restoring the values of R2 and R3. As a solution, we can use status register to record the register number and amount modified so that OS can undo the effect of partially executed instruction that causes a page fault.

6. **A computer has a cache, main memory, and a disk used for virtual memory. If a referenced word is in cache, 20 nsec(nano second) are required to access it. If it is in main memory but not in cache, 60 nsecare needed to load into the cache, and then the reference is started again. If the word is not in main memory, 12 msec (milli seconds) are required to fetch the word from the disk, followed by 60 nsec to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.6. What is the average time, in nanoseconds, required to access a referenced word on this system ? [5] - Tanishq**

Ans:

Ans: Effective memory access time when the word is in main memory (without disk involvement)= cache hit ratio* access time when the word is in cache+ (cache miss ratio* word access time when the word is in main memory; equal to = cache access time+ memory access time)

0.9*20 nsec+(1-0.9)*(60+20) nsec = 18 nsec+8 nsec= 26 nsec

Effective access time with disk= Page hit ratio*access time when the word is in main memory+ page miss ratio* access time when the word is in disk.

=0.6*26nsec+(1-0.6) (12 msec+60 nsec+20 nsec) =15.6 nsec+0.4*12msec =4.8 msec.

7. Assume a system with four resource types C=<6,4,4,2>, and the maximum claim table shown below. **The resource allocator is considering allocating resources according to the table shown below. Is this safe state ? Why and why not ? [5]- Vegi**

You can start the answer from here:

Maximum Claim table				
Process	R0	R1	R2	R3
P0	3	2	1	1
P1	1	2	0	2
P2	1	1	2	0
P3	3	2	1	0
P4	2	1	0	1

Current Allocation Table				
Process	R0	R1	R2	R3
P0	2	0	1	1
P1	1	1	0	0
P2	1	1	0	0
P3	1	0	1	0
P4	0	1	0	1

Ans:

8. **Consider three sets of users, professor, TA and student. There exists a folder marks to which only professor can write into and student and TA can only read from. There exists another folder called course material which can be written to by professor and TA and students can only read from it. Use Access Control Matrix, determine a security policy for this criteria. [5].- Samruddhi**

Ans: We use access control matrix. The domains are: Professors, TA and student. There are two objects: Folder 1 and Folder 2. Operations are: write and read.

Security Policy:

When a user requests access to a particular file, the operating system checks the access list associated with that file. If that user is listed for the requested access, the access is allowed. Otherwise, a protection violation occurs, and the user job is denied access to the file.

Object/Domain	Folder1	Folder 2
Professors	Write, Read	Write, Read
TA	Read	Write, Read
Student	Read	Read

9. **Explain the difference between "Pure ALOHA" and "Slotted ALOHA" in terms of implementation? What is the reason for the improved performance of "Slotted ALOHA" over "Pure ALOHA" [5] -Aneesh**

Ans:

Aloha vs Slotted Aloha, and the reason for improved performance.

Aloha and slotted Aloha are both multiple access protocols, where multiple users attempt to send signals asynchronously. If any two signals overlap even partially, both must be retransmitted. The difference between Aloha and slotted Aloha is when users are allowed to transmit signals. In normal Aloha, users can send their packets whenever they want, there is no global synchronization across users. In slotted Aloha, users are only allowed to send packets at the beginning of globally specified timeslots. At the small cost of a slight delay in sending packets, slotted Aloha has half the error rate of regular Aloha. This is because if there are any overlaps in slotted Aloha, they are restricted to one timeslot. There is no potential for a partial overlap to distort a packet like in normal Aloha. (A diagram explaining this will also suffice)

10. **Explain the issues of connection establishment in the Transmission Control Protocol of transport layer?How three-way handshake solve the problem. [5] - Harshita**

Ans:

Issue: Delayed duplicates is the issue of connection establishment: User sends packets and they may follow longest route. When the timeout expires, the packets are sent again. Receiver receives both old and new packets.

Transport layer should not reuse sequence numbers within twice the MSL (Maximum Segment Lifetime).

The main issue is how to keep packet sequence numbers out of forbidden region?If the host sends too much data too fast, it results sequence numbers to enter into forbidden curve.If the data rate is less than the clock rate, we also get into the trouble. Also, we have to remember the sequence numbers.

Three-way handshake solves the problem by assigning fresh sequence numbers by communicating with the host whenever establishes a new connection.

11. **Consider the following network. Distance vector routing is used and the following vectors have just come in to router C; from B: (5,0,8,1,6,2); from D: (16,12,6,0,9,10); from E (7,6,3,9,0,4). The cost of the links from C to B, D, and E, are 6, 3 and 5 respectively. What is C's new routing table? [5] -Jatin**

Network

Ans:

Going via B gives (11, 6, 14, 18, 12, 8). Going via D gives (19, 15, 9, 3, 12, 13). Going via E gives (12, 11, 8, 14, 5, 9). Taking the minimum for each destination except C gives (11, 6, 0, 3, 5, 8). The outgoing lines are (B, B, –, D, E, B).

13.2 Fixed partition multiprogramming systems with absolute translation and loading to fixed partition multiprogramming systems with re-locatable translation and loading. -Harshita

Ans: Motivation: To make programmer job easy. In "Fixed partition multiprogramming systems with absolute translation and loading", the programmer should know the starting address of the program for development. The main motiovation is to relinquish the programmer from the execution details. Able to load the program at different portions in the main memory. Ability to suspend and resume the program.

13.3 Fixed partition multiprogramming to variable partition multiprogramming. -Greeshma

Ans: Motivation: Distribution of memory is not effective. The programmer should know the size of partition to write the program. The memory is underutilized. Some programs may under utilize the partition whereas some other programs may need additional memory.

13.4 Contiguous storage allocation systems to non-contiguous storage allocation systems.-Samruddhi

Motivation: In contiguous storage allocation systems, the allocation scheme is inflexible. The files can not grow.

13.5 Single user dedicated systems with manual job-to-job transition to single user dedicated systems with single stream batch processing systems.-Vegi

Ans: Motivation: Job change time is significant.In serial systems, Machines were very expensive. Wasting time was not acceptable. To improve usage, the concept of batch OS was developed. The main idea is the use of software known as monitor. The user no longer has access to machine. The user submits the job (
operator. The operator batches the jobs together sequentially, plac
the computer.For example, if there