

Quiz2: CS3.301: Operating Systems and Networks, IIT Hyderabad

Time: 45 minutes; Max. Marks: 20

Student's Roll No.	Date of Exam	Student's Signature	Invigilator's Signature

A

Note: Answer all questions. Make appropriate assumption **A** a brief answer. There are 10 questions. Each question is for 2 marks.

A1	<p>Compare the deadlock prevention algorithms by preventing circular-wait with the various deadlock avoidance algorithms (like the banker's algorithm) with respect to the following: runtime overheads and system throughput</p> <p>Ans:</p> <p>(i) Runtime overheads: Bankers algorithm: The runtime overhead of bankers algorithm is $O(m \times n^2)$ where "m" is number resources of each type and "n" is number of processes. Even though the algorithm is executed for every request. So the runtime overhead is high.</p> <p>Deadlock prevention algorithm: Runtime overhead is very less or nil.</p> <p>(ii) Throughput: Banker's algorithm: No process is aborted. If the contention is high, the algorithm works well and gives highest throughput.</p> <p>Deadlock prevention algorithm: Throughput is reduced significantly. The process may be rejected several times. Repetition of work. CPU cycles are wasted.</p>
A2	<p>Two versions of a program P, called Ps and Pd, are developed using static and dynamic memory allocation, respectively. When an effort is made to execute Ps, an OS gives the message "Insufficient memory, can not run the program." However, the same OS executes program Pd without any difficulties. Explain why this happens.</p> <p>Ans: Case of Ps: In case of Ps, it is compiled using static memory allocation. The object program contains physical addresses. So, the program can only be executed from a fixed location. If the memory is not available, Ps can not be executed from other location. As a result, displays error condition.</p> <p>Case of Pd: In case of Pd, the compiler generates relocatable code. While executing, the base register is filled with the starting address of available memory and relocatable code is converted to absolute code based on the contents of the base register. So, in principle, program can be executed wherever the sufficient contiguous memory is available. So, Pd could be executed successfully.</p>
A3	<p>Identify whether the following statement is 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 few (less than five) sentences: Global frame allocation algorithm gives better throughput over local frame allocation algorithm.</p> <p>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.</p>
A4	<p>Explain why a Read operation to a file results into several block level write operations on DISK.</p> <p>Ans: File system has to track information about files. To store this information i-nodes are used. The inode is the generic name that is used in many file systems to describe the structure that holds the metadata for a given file, such as its length, permissions, and the location of its constituent blocks.</p> <p>Reading each block requires the file system to first consult the inode, then read the block, and then update the inode's last-accessed-time field with a write. So read operation of a file involves writing/updating of blocks.</p>
A5	<p>Explain the merits and demerits of RAID level 3</p> <p>Ans: RAID level 3: RAID 3 improves throughput with bit-level parity organization. Bit level Data stripping is used to improve the throughput: Splitting the bits of each byte across multiple disks. If we have array of 8 disks, we</p>

	<p>write bit i of every byte to disk i. One bit is used for error correcting. Data can be striped among disks and parity bit is stored in another disk. Bit level stripping is being used.</p> <p>Merits: The array of eight disks can be treated as single disk that are eight times normal size and eight times the access rate. Every disk participates in the read. But each access can read eight times as many data.</p> <p>Demerits: Not possible to process several requests in parallel as every request has to access all disks.</p>
A6	<p>Explain its positive and negative aspects of inverted page table approach.</p> <p>Ans: Inverted page table. One entry is maintained for each real page of memory. Entry consists of the virtual address of the page stored in that real memory location, with information about the process that owns that page. Decreases memory needed to store each page table, but increases time needed to search the table when a page reference occurs. There is only one page table in the system. Each virtual address space in the system consists of a triple <Process-id, page #, offset>. When a memory reference occurs part of <process-id, page#> is presented to memory subsystem. The inverted page table is searched for a match. Use hash table to limit the search to one — or at most a few — page-table entries.</p> <p>Positive aspect: Decreases amount of memory for page table.</p> <p>Negative aspect: increases time needed to search the table. It is used for the systems with the address spaces greater than 32 bits, 64-bit UltraSPARC and PowerPC.</p>
A7	<p>Explain the difference between dynamic loading and dynamic linking.</p> <p>Ans: Dynamic loading: Dynamic loading is different from static loading. In static loading, entire program is loaded at once. In dynamic loading, routine is not loaded until it is called. It is used to obtain better memory-space utilization. All routines are kept on disk in a re-locatable load format.</p> <p>Dynamic linking: In static linking, system language libraries are treated like any other object module and loader combines them into binary program image. In dynamic linking, the linking is postponed until execution time. Small piece of code, stub, used to locate the appropriate memory-resident library routine. Stub replaces itself with the address of the routine, and executes the routine.</p> <p>Difference: The notion of stub is not used in dynamic linking.</p>
A8	<p>Explain the problems if you swap a process with pending I/Os.</p> <p>Ans: If you swap a process with pending I/O's, the I/O are executed asynchronously and overwrite the memory of other process which was swapped-in in place of old process. This will lead to incorrect execution.</p>
A9	<p>What is the purpose of "translation lookaside-buffer"?</p> <p>Ans: A translation lookaside buffer (TLB) is a memory cache that stores the recent translations of virtual memory to physical memory.</p> <p>Purpose: Normally, with paging solution for external fragmentation, several main memory accesses are required to access data: some are for page table and one for the data. TLB is used to reduce the main memory accesses. Once a memory pair of <page id, frame id> is retrieved, this information is stored in TLB. If the memory access is for the same page in time, the information in TLB can be used. So, TLB is used to reduce the time taken to access a user memory location.</p>
A10	<p>What is the purpose of dirty bit entry in the page table?</p> <p>Ans: A dirty bit or modified bit is a bit that is associated with a block of computer memory and indicates whether the corresponding block of memory has been modified. The dirty bit is set when the processor writes to (modifies) this memory. The bit indicates that its associated block of memory has been modified and has not been saved to storage. When a block of memory is to be replaced, its corresponding dirty bit is checked to see if the block needs to be written back to secondary memory before being replaced or if it can simply be removed.</p> <p>Purpose: dirty bit reduces the write requirement in case of page fault. When a victim page is selected for page replacement, normally we have to write the victim page information back to disk and bring the data of the requested page. If the dirty bit is not set, we need not write the data.</p>