Dhirubhai Ambani Institute of Information & Communication Technology

IE411/MC214 Operating Systems

Final Examination | Semester I 2022-23 | Time: 2 Hours | Max Marks: 40

All questions carry equal marks.

1. What is the effect of allowing two entries in a page table to point to the same page frame in memory? Explain a scenario where this may improve performance.

   By allowing two entries in a page table to point to the same page frame in memory, users can share code and data. If the code is reentrant, much memory space can be saved through the shared use of large programs such as text editors, compilers, and database systems. "Copying" large amounts of memory could be effected by having different page tables point to the same memory location.

2. Most systems allow programs to allocate more memory to its address space during execution. What is required to support dynamic memory allocation in the following schemes:
   (a) contiguous-memory allocation
   (b) pure segmentation
   (c) pure paging

   (a) contiguous-memory allocation: might require relocation of the entire program since there is not enough space for the program to grow its allocated memory space
   (b) pure segmentation: might also require relocation of the segment that needs to be extended since there is not enough space for the segment to grow its allocated memory space
   (c) pure paging: incremental allocation of new pages is possible in this scheme without requiring relocation of the program's address space.

3. Consider a paging system with the page table stored in memory.
   (a) If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?
   (b) If we add associative registers, and 75 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.)

   (a) A paged memory reference requires 2 memory accesses: one to get the frame number for the page number from memory and actually accessing the data. So if a memory reference takes 200 nanoseconds then a paged memory reference will take 400 nanoseconds.
   (b) If associative registers are added and the hit ratio is 75% then effective memory access time is given as EMAT = 0.75*200 + 0.25*400 = 250 nanoseconds

4. Consider a computer system with a 32-bit logical address and 8-KB page size. The system supports up to 1 GB of physical memory. How many entries are there in each of the following?
   (a) A conventional, single-level page table
   (b) An inverted page table

   (a) 32 bit address. Page is 8KB = 2^{13} bits. Hence number of entries in the table is 2^{19}.
   (b) 1 GB = 2^{30} bytes. Page size is 2^{13}. Number of entries is 2^{17}.

5. What is the copy-on-write feature, and under what circumstances is its use beneficial? What hardware support is required to implement this feature?

   Copy-on-write takes advantage of a system's paging hardware to avoid copying the contents of a frame when a page needs to be duplicated. When a page copy is requested, the OS creates a new page table entry for the copied page. However, this entry doesn't point to a new frame. Instead, it points to the frame that holds the original page. Thus, the two page-table entries point to the same frame.

   The hardware support required to implement is the following: on each memory access, the page table needs to be consulted to check whether the page is write-protected. If it is indeed write-protected, a trap would occur and the operating system could resolve the issue.

6. Consider a file system that uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

   (There is some ambiguity in the number of indirect pointers. As long as your calculation is consistent with the numbers you have assumed, you get full credit. You may have assumed 1 each or 12 each)

   Number of pointers/block = 8K/4 = 2048

   Max file size = (12 * 8KB) + (1 * 2048 * 8KB) + (1 * 2048 * 2048 * 8KB) + (1 * 2048 * 2048 * 2048 * 8KB)

7. Describe and compare the performance of the three techniques for allocating disk blocks (contiguous, linked, and indexed) for both sequential and random file access.

Contiguous allocation:
In this scheme, each file occupies a contiguous set of blocks on the disk. It supports both accesses (sequential and direct). It's extremely fast as the number of seeks are minimal because of contiguous allocation of file blocks. But on the other hand it has some disadvantages as it cannot deal with external fragmentation. Also increasing the size of it may get difficult.

Linked allocation:
In this type of allocation,each file is a linked list of disk blocks and the disk blocks can be scattered anywhere on the disk. This type is very flexible with the file size. This method also does not suffer from external fragmentation. This makes it better in terms of memory utilization. Random access is quite slow.

Indexed allocation:
There is a special block known as the index block contains the pointers to all the blocks occupied by a file in this type of allocation. Each file has its own index block. This type supports direct access therefore provides fast access to the file blocks. It also overcomes with the problem of external fragmentation. Access can be quite fast for small files and it can support very large files as well.

8. Discuss the strengths and weaknesses of implementing an access matrix using
   (a) access lists that are associated with objects.
   (b) capabilities that are associated with domains.

(a) The strength of storing an access list with each object is the control that comes from storing the access privileges along with each object, thereby allowing the object to revoke or expand the access privileges in a localized manner.

The weakness with associating access lists is the overhead of checking whether the requesting domain appears on the access list. This check would be expensive and needs to be performed every time the object is accessed

(b) Capabilities associated with domains provide substantial flexibility and faster access to objects. When a domain presents a capability, the system just needs to check the authenticity of the capability and that could be performed efficiently. Capabilities could also be passed around from one domain to another domain with great ease allowing for a system with a great amount of flexibility. However, the flexibility comes at the cost of a lack of control; revoking capabilities and restricting the flow of capabilities is a difficult task.

9. Explain the functioning of polling based and interrupt driven I/O. Give an example each where polling or interrupt based I/O is preferable.

   An interrupt is an event that is triggered by external components other than the CPU that alerts the CPU to perform a certain action. When an interrupt occurs, the interrupt handler is executed. In contrast, polling is a synchronous activity that samples the status of an external device by the CPU at a regular interval.

   Polling wastes CPU cycle even when there is no I/O due. In interrupt driven I/O, CPU executes the handler only when some I/O is due. Hence for any task that is periodic (e.g. reading sensor data at fixed interval), polling may be suitable. For sporadic tasks, interrupt based mechanism is better.

10. Consider the parameter Δ used to define the working-set window in the working-set model. When Δ is set to a low value, what is the effect on the page-fault frequency and the number of active (nonsuspended) processes currently executing in the system? What is the effect when Δ is set to a very high value?

    A smaller value of $\triangle$ means that less number of frames out of 'the total number of frames in the system' will be allotted to a particular process. That being said, we can also conclude that more number of frames will be available for other processes. Therefore, we can say that number of active processes will increase but since they have less number of allotted frames, therefore it will lead to more page faults.

    Now, on the contrary, a larger value of $\triangle$ will lead to a larger number of frames being allotted to a particular process. That means less number of frames will be available for other processes. Hence the number of active processes will decrease but also page faults for the active process will decrease.

    Also when the value of $\triangle$ is large, but the frame requirement of any process is less than that, then this will lead to unnecessary blocking of the process even when sufficient number of frames are available, hence it will make it difficult for a process to become active.