Paper Title: Operating System (IOPS332C)

Quiz 2 Time : 1 hr 30 mts

Q1. State True/False with jusification if the answer is false: [5]

a. If there is no mutual exclusion condition for any resource in the system, then there is no possibility for deadlock. : **True**

b. If the shared resources are numbered 1 through N and a process can only ask for resources that are numbered higher than that of any resource that it currently holds, then deadlock can never happen. **True**

c. The Least Recently Used (LRU) page replacement strategy is based on the principle of spatial locality (locality in space) as opposed to temporal locality (locality in time). : **False**

d. The Shortest Seek Time first disk scheduling algorithm favors middle cylinders over innnermost and outermost cylinders : **True**

e. Thrashing happens when degree of multiprogramming is too high : **True**

Q2. A computer system has a 36-bit virtual address space with a page size of 8K, and 4 bytes per page table entry. **[1+2+2 = 5]**

(a) How many pages are in the virtual address space?
(b) What is the maximum size of addressable physical memory in this system?
(c) If the average process size is 8GB, would you use a one-level, two-level, or three-level page table? Why? Compute the average size of a page table in question (c) above.

Solution:

1. A 36 bit address can address 2^36 bytes in a byte addressable machine. Since the size of a page 8K bytes (2^13), the number of addressable pages is 2^36 / >2^13 = 2^23
2. With 4 byte entries in the page table we can reference 2^32 pages. Since each page is 2^13 B long, the maximum addressable physical memory size is 2^32 * 2^13 = 2^45 B (assuming no protection bits are used).
3. 8 GB = 2^33 B

   We need to analyze memory and time requirements of paging schemes in order to make a decision. Average process size is considered in the calculations below.

   1 Level Paging
   Since we have 2^23 pages in each virtual address space, and we use 4 bytes per page table entry, the size of the page table will be 2^23 * 2^2 = 2^25. This is 1/256 of the process' own memory space, so it is quite costly. (32 MB)

   2 Level Paging
   The address would be divided up as 12 | 11 | 13 since we want page table pages to fit into one page and we also want to divide the bits roughly equally.

   Since the process' size is 8GB = 2^33 B, I assume what this means is that the total size of all the distinct pages that the process accesses is 2^33 B. Hence, this process accesses 2^33 / 2^13 = 2^20 pages. The bottom level of the page table then holds 2^20 references. We know the size of each bottom level chunk of the page table is 2^11 entries. So we need 2^20 / 2^11 = 2^9 of those bottom level chunks.

The total size of the page table is then:

//size of the outer page table		//total size of the inner pages		
1 * 2^12 * 4	+	2^9 * 2^11 * 4	=	2^20 * (2^-6 + 4) ~4MB

3 Level Paging

For 3 level paging we can divide up the address as follows:

8 | 8 | 7 | 13

Again using the same reasoning as above we need 2^20/2^7 = 2^13 level 3 page table chunks. Each level 2 page table chunk references 2^8 level 3 page table chunks. So we need 2^13/2^8 = 2^5 level-2 tables. And, of course, one level-1 table.

The total size of the page table is then:

//size of the outer page table	//total size of the level 2 tables	//total size of innermost tables	
1 * 2^8 * 4	2^5 * 2^8 *4	2^13 * 2^7 * 4	~4MB

As easily seen, 2-level and 3-level paging require much less space then level 1 paging scheme. And since our address space is not large enough, 3-level paging does not perform any better than 2 level paging. Due to the cost of memory accesses, choosing a 2 level paging scheme for this process is much more logical.

4. Calculations are done in answer no. 3.

Q3. (a) Consider a file consisting of 100 block. Assume that the File Control Block is already in memory. Calculate how many disk I/O (read/write) operations are required for contiguous allocation strategy if a new block is to be added assuming there is no room to grow in the beginning but room to grow in the end. Block information to be added is stored in the memory.

Solution : *Since there is no room to add block in the beginning, all 100 have to be shifted one place to the right to make space for the new block in the begining. Shifting one place to the right means reading its present content and writing to the next location i.e. 1 read and 1 write. This has to be done for 100 blocks, Thus, in total 100 r + 100 w. Finally one write for the new block at the vacant position. Thus, # I/O operations = 100 (1r+1w) + 1w = 201 operations.*

(b) An engineer has designed a FAT-like system and he has used 24 bits for each entry. For a 32-GB disk, what is the minimum size of a file allocation in this system? Justify your answer.

Solution:

A 32 GB disk has 2^35 bytes of storage; if each entry in the FAT has 24 bits, then there can be at most 2^24 allocation chunks (one per FAT entry), so each allocation chunk must be 2^11 bytes = 2KB

(c) Consider an index-based file system with the inode containing 64 indexes, 1 indirect index pointing to a disk block containing an array of direct indexes, and 1 2-level index in the usual way. Assume that each index takes 4 bytes.

(i) What is the maximum file size under this arrangement, if a disk block is 1024 bytes? Explain how do you compute this maximum size.

Solution : An indirect block has 1024 bytes * 1 index/4 bytes = 256 pointers
So a file has at most 64 + 256 + 256*2 blocks and (64 + 256 * 256^2) * 1024 bytes.

(ii) How many disk accesses does it take to read one disk block at location 3000321 within a file, assuming no caching. Justify your answer. **[2+2+2+2=8]**

Solution : Block 3000321 is block number 2930, which means we have to read the inode, the 2-level index, a direct index, and the data block. 4 reads.

Q4. Given a disk with 100 cylinders (0 to 99). Exactly one time unit is required to move the read/write head from one cylinder to the next. At time 0 the heads are cylinder 0 and no requests are pending. The following six requests arrive at tmes shown in:

Time :	0	10	20	70	80	90
Cylinder :	21	75	16	68	2	17

Note that once heads are moving arrival of new request will not alter the intended distination of the heads in movement. Assuming, that rotational and transfer times are negligible, give the order in which the cylinders will be visited and the total time required for the Scan algorithm. **[2]**

Solution : The results are as folows :

Time	Cylinder	Direction	Queue
0	0	up	21
21	21	up	75,16
75	75	up	16,68
99	99	down	16,68,2,17
130	68	down	16,2,17
181	17	down	2,16
182	16	down	2
196	2	-	-

************-----------------------**************-------------------------------**********------------------