IIT Guwahati - Department of Computer Science & Engineering

CS343-Operating Systems- End Semester Exam [22.11.2021]

Total 40 marks, 120 minutes.

Section I: Essay Type (4 questions, 4 marks each). Write answers with necessary explanation in the given space.

Question 1:
Explain all possible scenarios where a miss in page table results in updating a TLB entry.

Solution:
When there is a page fault, the demanded new page is brought to main memory, the page table is updated by adding the new page and its corresponding frame number. Bringing of a new page can evict another page from main memory. Now there can be two scenarios based on whether the evicted page has a TLB entry or not.
Case 1: The evicted page information is not present in TLB. However, the updated page table entry is entered in the TLB. This can cause one of the TLB entry to get deleted for giving room for new TLB entry.

Case 2: The evicted page information is present in TLB. This will result in corresponding entry in TLB to get invalidated. Moreover, the updated page table entry is entered in the TLB.
[2 marks each for describing each case]

Question 2:
Consider a disk scheduling scenario. The disk queue gets new entries at regular intervals such that at any given point in time there will be always a minimum of 3 entries in the disk queue. Two entries A and B that entered queue long back is still not serviced due to the newly arriving entries and the operating rules of the scheduling algorithm. Mention a standard disk scheduling algorithm, and specific characteristics of incoming requests, positions of A and B under which starvation of A and B happens indefinitely?

Solution:
The disk scheduling algorithm should be SSTF. **[2 marks]**
The newly arriving requests should be far from A and B and at any given point in time, there should be at least one request (R) in the disk queue whose cylinder number is closer to the present head position than the cylinder numbers of A and B. In SSTF scheduling, the head will move towards R from its current position than moving to A and B. Hence A and B will starve indefinitely. **[2 marks]**

Question 3:
Digital signature-based authentication and asymmetric encryption both use public key as well as private key. Is the key usage same between these two techniques? Why/Why not? Explain the reason.

Solution: In digital signature-based authentication, the signature (hash) of a message is encrypted using the private key (secret key) of the sender. The signature (hash) can be verified only by decrypting with the sender's public key (shared key). This authenticates the genuine sender. **[2 marks]**

In asymmetric encryption, public key of the receiver is used to encrypt the message and the message can be decrypted only by using the private key of the recipient, which is known only to the receiver. This ensures that only the intended recipient can see the message. **[2 marks]**

Question 4:
Let D1 and D2 be two distributed file systems. Let f1 and f2 are file in D1 and D2, respectively. D1 has location transparency as well as location independence, whereas D2 has location transparency, but not location independence. What does this convey with respect to storage of f1 and f2?

Solution: A transparent DFS hides the location where in the network the file is stored. Location transparency–file name does not reveal the file's physical storage location. Location independence–file name does not need to be changed when the file's physical storage location changes. **[2 marks]**

Here, physical storage of f1 and f2 are transparent. This means that, from the file names the location of actual storage of these files cannot be identified. But for f1, even if the actual physical storage of f1 changes, still access to f1 is not affected. Since f2 is on D2, which is not location independent, access to f2 can get impacted if the file is moved. **[2 marks]**

Section II: File upload Type (3 questions, 8 marks each). Write answers with necessary explanations in a paper and upload the scan copy of the page(s). Maximum 2 pages per question.

Question 5:
A system uses 64KB virtual address space and has a main memory of capacity 32KB. It uses a 5-entry hashed page table. The page size is 4KB. The hash function uses i= v%5 to index into the hash table, where 'v' is the virtual page number and 'i' is the index in the hashed table. Consider two active user processes P1 and P2 whose hashed page table entries are all initialized with null. The memory management unit uses a global frame allocation algorithm for user processes. The OS kernel code is permanently residing in continuous frames in the lower half of the main memory starting from frame 0. For user processes, the system uses LRU page replacement policy. When user frame space is empty, frame allocation is done from lower frames to higher frames. The following set of requests (process ID, virtual address) generated from the CPU, given in the order of arrival. (P1, 0x7424), (P1, 0xD200), (P1, 0x7244), (P1, 0x2424), (P2, 0x4040), (P2, 0xF404), (P2, 0xF000), (P1, 0xA024), (P1, 0x6420), (P2, 0x4848). Answer the following questions after successfully mapping the above address sequences on the respective hashed page tables.

(a) Draw the final contents of main memory frames clearly indicating process number and virtual page number mapped into each frame.
(b) Draw the hashed page table of P1.
(c) Draw the hashed page table of P2.

Solution: Virtual address space = 64 KB = 2^{16}
Page size = frame size= 4 KB = 22 * 210 = 212
So, total number of pages = 24 = 16
Main memory space = 32 KB = 25 * 210 = 215
So, total number of frames= 23 = 8
(P1, 0x7424), (P1, 0xD200), (P1, 0x7244), (P1, 0x2424), (P2, 0x4040), (P2, 0xF404), (P2, 0xF000), (P1, 0xA024), (P1, 0x6420), (P2, 0x4848)

Main memory frames and contents

Frame #	Contents (Process, page number)
0	OS kernel
1	
2	
3	
4	~~(P1,7)~~ (P1,A)
5	~~(P1,D)~~ (P2,F)
6	~~(P1,2)~~ (P1,6)
7	(P2,4)

HPT (P1)	
0	→(A,4)
1	→ (6,6)
2	~~→(7,4) → (2,6)~~ NULL
3	~~→(D,5)~~
4	NULL

Red indicates deleted entry
(P1, 0x7244), (P2, 0xF000) & (P2, 0x4848) are hits.

HPT (P2)	
0	→(F,5)
1	NULL
2	NULL
3	NULL
4	→(4,7)

Each chain in HPT stores page number and corresponding frame number.
Even if only final entries are shown, give full mark. No need for deleted entries.
[4 marks for MM and 2 marks each for Hashed Page tables of P1 and P2]

Question 6:
Consider a file system F and a file f in it that stores data in 10 disk blocks. Assume that the file control block/index block is always kept in main memory. Let O be a standard operation of either adding a new block or removing an existing block (Eg: insert a new block after nth block, remove the nth block) that is done on f with A being one of the three standard file allocation schemes of F. The total number of disk I/O reads and writes in 5 different cases are given below. For each case, identify the set of all possible combinations of O and A. Explain the process. [Hint: Your answer should be like O=delete 3rd block, A=indexed allocation]

(a) 5 reads and 2 writes.
(b) 3 reads and 3 writes.
(c) one read and 2 writes.
(d) one read and zero writes.
(e) zero reads and one write.

Solution:

(a) 5 reads and 2 writes. O=insert new block after 5th block, A=linked allocation **[1 mark]**
5 reads to reach the 5th block, from 5th block get the pointer to 6th block, add it in new block and write the new block, add address of new block to 5th block, write 5th block.

(b) 3 reads and 3 writes. **[Write any 1 case of the following, 1 mark]**
1. O=delete the 7th block, A=contiguous allocation
   Read the 8th block, write it to place of 7th block, Read the 9th block, write it to place of 8th block, Read the 10th block, write it to place of 9th block.
2. O=delete the 4th block, A=contiguous allocation
   Read the 3rd block, write it to place of 4th block, Read the 2nd block, write it to place of 3rd block, Read the 1st block, write it to place of 2nd block. [assumption: memory is not fixed at beginning]

(c) one read and 2 writes. **[Write any 2 cases, 2x1 = 2 marks]**
1. O=insert a new block after 10th block, A=linked allocation
   Since 10th is the last block read it direct from file control block, write the new block, add address of new block to 10th block, write the 10th block.
2. O=insert a new block after 1st block, A=linked allocation
   Read the 1st block, get the address to the 2nd block, write the new block with this address to the 2nd block, update the address of new block in the first block and write first block to disk.
3. O=insert a new block after 9th block, A=contiguous allocation
   1 read 1 write for shifting the last element and 1 write for writing the new block

(d) one read and zero writes. O=delete the 1st block, A=linked allocation **[1 mark]**
Read the first block, get pointer to second block, update this in file control block such that file begins from the 2nd block thereby removing (deleting) the 1st block

(e) zero reads and one write. **[1 mark for each, 3x1 = 3 marks]**
1. O=add a new block anywhere, A=indexed allocation
   write the block anywhere and update it location in the index node in memory.
2. O=insert a new block after the 10th block, A=contiguous allocation
   write the block in the free space after 10th block, update the file control block by increasing the length of the file by one more block. (Assumption: room for growing at end available)
3. O=add a new block in beginning, A=linked allocation
   From file control block we will get the existing first block, get its address and write in the pointer field of new block. Write the new block. Now new block is linked to previous first block, thereby adding a new block in beginning.

Question 7:

(a) Consider a disk arm seek movement that operates on SSTF scheduling algorithm. Currently the head is positioned (at zero speed) at sector number 50 of cylinder 100. At time T, two disk requests (R1&R2) of the form [request id, sector number, cylinder number] are in the scheduler queue. [R1, 128, 244], [R2, 16, 86]. When the arm starts from zero speed, it covers 1 track in first 1ms, covers 2 tracks in next 1ms, then thereafter it will cover 4 tracks/ms, 8 tracks/ms etc. The coast speed is 32 tracks/ms. The deceleration for settling at target cylinder is also similar to the acceleration mentioned above. What is the minimum seek time needed to service R1 & R2? Explain how you compute it.

Solution:

Consider only the cylinder number of the request.

[R1, 128, **244**], [R2, 16, **86**].

As per SSTF the head at cylinder 100 will go to 86 (R2) and then to 244 (R1).

Cylinder 100 to 86 movement is as follows

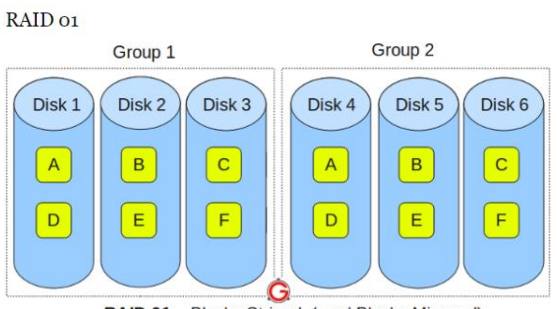
Cylinder #	100	99	97	93	89	87	86
Track/ms		1	2	4	4	2	1
Seek Time (ms)		1	2	3	4	5	6

Cylinder 86 to 244 movement is as follows

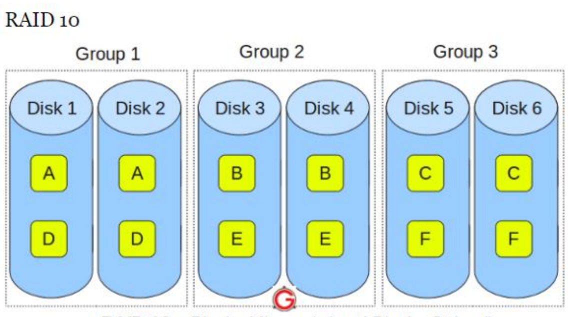
Cylinder #	86	87	89	93	101	117	149	181	213	229	237	241	243	244
Track/ms		1	2	4	8	16	32	32	32	16	8	4	2	1
Seek Time (ms)		1	2	3	4	5	6	7	8	9	10	11	12	13

Total time is 6+ 13= 19 ms **[2 marks for explanation and 2 marks for final answer]**

(b) Consider a RAID system with a total of six identical disks (D1, D1,.., D6) in terms of storage capacity and access speed. Assume we have to store 6 blocks of data viz; A, B, C, .., F in these disks. Draw a schematic diagram of RAID 10 and RAID 01 clearly marking the name, grouping and contents of each disk.



RAID 01 – Blocks Striped. (and Blocks Mirrored)



RAID 10 – Blocks Mirrored. (and Blocks Striped)

2 marks for RAID10 [if grouping, disk naming and data block mapping is correct]
2 marks for RAID01 [if grouping, disk naming and data block mapping is correct]
No partial marks.