TERNA ENGINEERING COLLEGE, NERUL

COMPUTER ENGINEERING DEPARTMENT

Assignment No. 3

Subject: Operating System **Class: S.E./ B** **Year: FH2020**

Q1. Explain hardware support for paging.[CO5]

Q2. Consider reference string 0,1,2,3.0,1,2,3,0,1,2,3,4,5,6,7 with frame size 3 and 4. Calculate page fault.[CO5]

Q3. What are the different allocation methods with respect to the file system?[CO6]

Q4. Suppose that the head of a moving disk with 200 tracks, numbered 0 to 199, is currently serving a request at track 143 and has just finished a request at track125. The queue of request is kept in the FIFO order 86,147, 91, 177, 94, 150, 102, 175, 130. What is the total number of head movements needed to satisfy these requests for the following disk scheduling algorithm?[CO6]

1. FCFS **2.** SSTF **3.** SCAN **4.**C-SCAN **5.** LOOK

Q.1

Ans:

- Each operating system has its own methods for storing page tables. Most allocate a page table for each process.
- A pointer to the page table is stored with the other register values. in the process control block.
- When the dispatcher is told to start a process, it must reload the user registers and define the correct hardware page table values from the stored user page table
- Hardware implementation can be done in several ways
- In the simplest case, the page table is implemented as a set of dedicated registers. These registers should be built with very high speed logic to make the paging address translation efficient.
- Every access to memory must go through the paging map, so efficiency is a major consideration.
- The CPU dispatcher reloads these registers, just as it reloads the other registers. Instruction to load or modify the page table registers are privileged so that the only OS can change the memory map.
- The use of registers for the page table is satisfactory if the page table is reasonably small
- Most contemporary computer allows the page table to be very high large.

Q2.

Ans:

Reference String:	Frame size:
0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 4, 5, 6, 7	3 & 4

Using FIFO Algorithm.

Frame size 3

0	1	2	3	0	1	2	3	0	1	2	3	4	5	6	7
0	0	0	3	3	3	3	2	2	1	1	1	4	4	4	7
	1	1	1	0	0	0	3	3	3	2	2	2	5	5	5
		2	2	2	1	1	1	0	0	0	3	3	3	6	6

Total page fault = 16

Now,

Frame size 4

0	1	2	3	0	1	2	3	0	1	2	3	4	5	6	7
0	0	0	0									4	4	4	4
	1	1	1									1	5	5	5
		2	2									2	2	6	6
			3									3	3	3	7

Total page fault = 8

Frame size	Page Fault
3	16
4	8

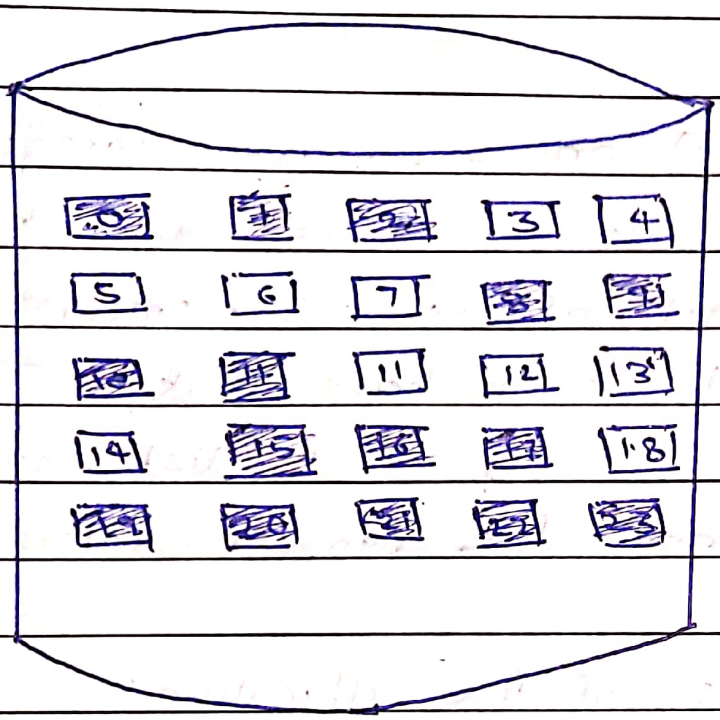
Q3.

Ans:

- Files are usually stored on secondary storage devices such as disk. These files are later called back when needed. As part of their implementation, files must be stored in the hard disk. This has to be done in such a way that disk space is utilized effectively and files can be accessed quickly.
- There are 3 methods of file allocation:
  1. Contiguous allocation
  2. Linked allocation
  3. Indexed allocation.

1. Contiguous allocation.

- It requires that each file occupy a set of contiguous block on the disk. Contiguous means Continuous.
- Because of contiguous allocation, there is minimal or no disk head movement which reading/writing the blocks of the file.
- The seek time is minimal over here. Consequently, access time of a file, and I/O performance is greatly improved.
- To access a file, there we only need to know the starting location and length of file which are stored in directory.



File	Start	length
A	0	3
Hello	8	4
marks	15	3
Photos	19	5

- It has certain problems associated with it

① Finding the space for new file

② Problem of external fragmentation.
It occurs whenever free space is broken into tiny chunks

③ Compaction can take up lot of time and may need system to be down where in normal operation will not be permitted

④ Determining space to be allocated for file if it needs to grow.

Advantage:

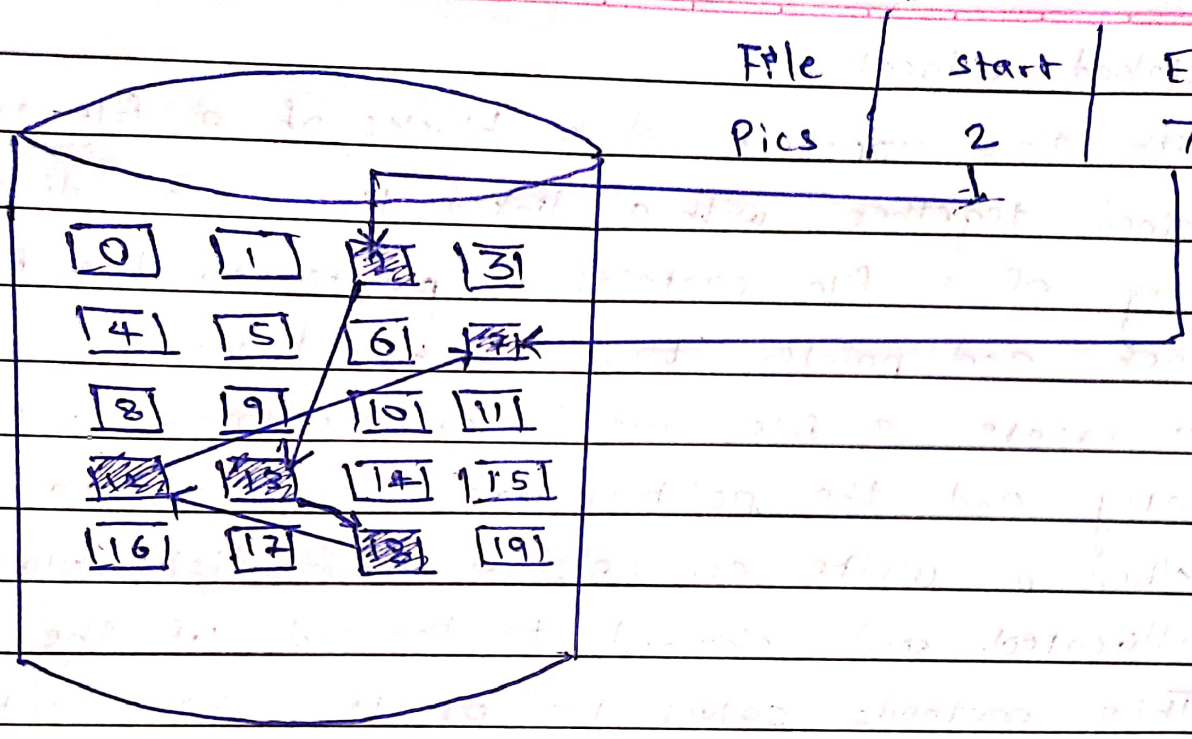
- Contiguous allocation is easy to implement

Disadvantage:

- It can be considered as a form of dynamic memory allocation, and external fragmentation may occur and compaction may be needed

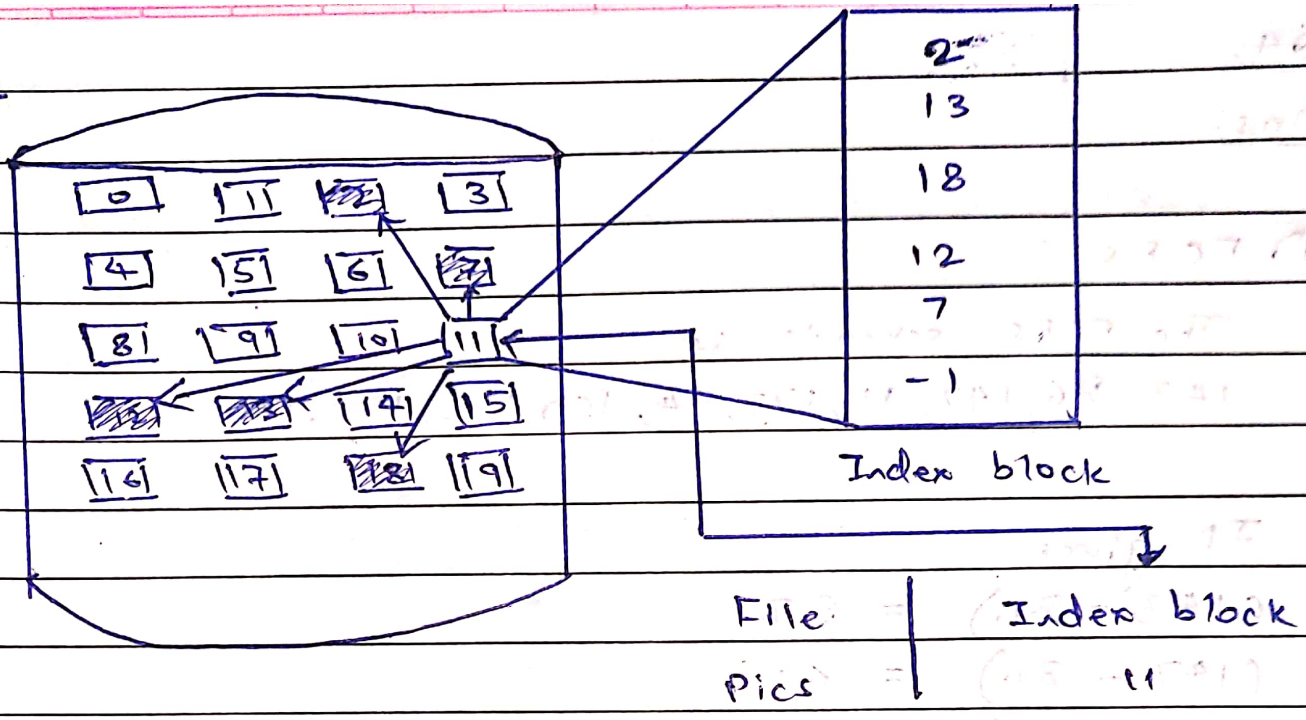
(2) Linked allocation

- With this approach, disk blocks of a file are chained together with a linked list. The directory entry of a file contains a pointer to the first block and pointer to the last block
- To create a file, we create a new directory entry and the pointers are initialized to nil. When a write occurs, a new disk block is allocated and chained to the end of the list.
- This methods solves the problem with contiguous allocation.
- Here the blocks of a single file can be scattered anywhere on the disk. The reason because the entire file is implemented as a linked list.
- Here every file is an element of linked list.
- The directory maintained by the OS contains a pointer to the first and last blocks of a file.
- Each block of a file contains a pointer to the next block after it in the list.
- Advantages includes no external fragmentation, size of file need not be declared at start, a file can grow as long as free blocks are available on disk.
- Disadvantages:
  It works perfectly for sequencial access only. space needs to be allocate in blocks for pointer. error in pointer links can lead to invalid read



③ Indexed allocation

- With file allocation table in memory, linked list allocation support random access, but the entire table must be in memory all the time.
- In indexed allocation, all the pointers are kept in one location called as index block
- There is a index block assigned to each file and this index block holds the disk block addresses of that particular file.
- There is a pointer from i^{th} entry in the index block to the i^{th} block of file.
  It means n^{th} entry in index block holds the address of n^{th} disk block
- The address of the index block of the file is maintained in directory
- In order to locate and read i^{th} block, pointer in the i^{th} index block entry is used
- Initially when the file is created, all pointers in the index block are initialized to nil



- Advantages

① Index allocation supports random access

② There is no external fragmentation, because any free block on the disk can be allocated as per request for more space

- Disadvantages

① The pointer overhead of index block is more with compared to the pointer overhead of linked allocation.

Q4.

Ans:

① FCFS :

The FCFS schedule is

143, 86, 147, 91, 177, 94, 150, 102, 175, 130

It gives

(143 - 86) = 57
(147 - 86) = 61
(147 - 91) = 56
(177 - 91) = 86
(177 - 94) = 83
(150 - 94) = 56
(150 - 102) = 48
(175 - 102) = 73
(175 - 130) = 45

565

Total head movements are 565 cylinders

② SSTF :

The SSTF schedule is

143, 147, 150, 130, 102, 94, 91, 86, 175, 177

It gives

(147 - 143) = 4
(150 - 147) = 3
(150 - 130) = 20
(130 - 102) = 28
(102 - 94) = 8
(94 - 91) = 3
(91 - 86) = 5
(175 - 86) = 89
(177 - 175) = 2

162

Total head movements are 162 cylidders

③ SCAN :

The SCAN schedule is

143, 147, 150, 175, 177, 199, 130, 102, 94, 91, 86

It gives

(147 - 143) = 4
(150 - 147) = 3
(175 - 150) = 25
(177 - 175) = 2
(199 - 177) = 22
(199 - 130) = 69
(130 - 102) = 28
(102 - 94) = 8
(94 - 91) = 3
(91 - 86) = 5

169

Total head movements are 169 cylinders

④ C-SCAN

Schedule is 143, 147, 150, 175, 199, 86, 91, 94, 102, 130

147 - 143 = 4
150 - 147 = 3
175 - 150 = 25
199 - 175 = 2
199 - 177 = 22
199 - 86 = 113
91 - 86 = 5
94 - 91 = 3
182 - 94 = 8
130 - 102 = 28

213

There are 213 head movements

⑤ Look

Schedule is 143, 147, 150, 175, 177, 86, 91, 94, 102, 130

147 - 143 = 4

150 - 147 = 3

175 - 150 = 25

177 - 175 = 2

177 - 86 = 91

91 - 86 = 5

94 - 91 = 3

102 - 94 = 8

130 - 102 = 28

169

Total head movements are 169 cylinders