Free Space Management:-

→ The disk space is limited. we need to reuse the space from deleted files for new files. To keep track of free disk space, the system maintains a free-space list. The free space list records all free disk blocks, those are not allocated to some file.

→ To allocate space to the new file we search the space list ~~where for~~ for to know where the blocks are free in the disk.

→ If we delete a file from the disk then the space is added to the spacelist and if we allot space to the new file then that particular space is removed/deleted from the free-space list.

→ To maintain the free-space list we have some methods. ~~Those are.~~

1) ~~Bit Vector (or) Bit map~~

2) ~~Linked list~~

3)

1) Bit vector/Bit map:-

It says that for every block we have to maintain in a bit. If the bit is 1 then the block is free Other wise the block is alloted.

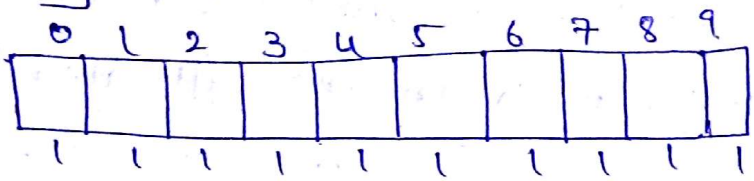
i.e, free block is represented by 1

alloted block is represented by 0.

Initially all blocks are free i.e, before alloting any space to any file. at that time all blocks

are free and bits of all blocks are 1.
Here vector means an array and bit vector means array of bits.

Ex:- Suppose 10 blocks are there in a disk and initially all bits are 1.



Jam alloted files to the 1, 3, 5, 6 and 7 blocks then the bits are set to 0

i.e;

0	1	2	3	4	5	6	7	8	9
1	0	1	0	1	0	0	0	1	1

the bit map is :- 1010100011.

The bit map shows which blocks is free and which blocks are currently occupied.

Advantage:-

→ It is a simple method
→ Easy to find first free block, n consecutive free blocks.
→ Sequentially check each word in the bit map to see whether the value is 0 or not.

It is also possible to calculate a particular free block by using the below formula..

(number of bits per word * number of 0 valued words) + offset of first 1 bit.

→ The problem with the bit map is it requires extra space and is kept in main memory.

Ex:- block size = 2^{12} bytes and.

Disk size = 2^{30} bytes.

number of blocks = $\frac{\text{disk size}}{\text{block size}} = \frac{2^{30}}{2^{12}} = 2^{18}$ bits

To represent the bit map we need 2^{18} bits in space in main memory.

→ If the size of the disk is very large then it need more space to store the bit map.

→ If we are not allocating any file to block to store the file also we need represent / maintain the bit map in main memory it by this also the main memory space is wasted to overcom -e this problem we approach another method i.e., Linked list.

2) Linked List:-

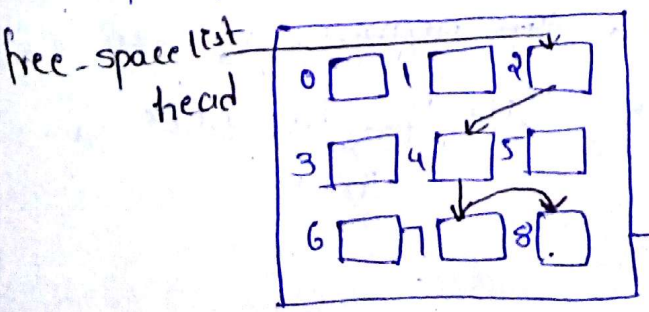
→ In this method all the free blocks are link to gether using the linked list.

→ If we maintain a pointer to the first free block. from their we have pointer / link to the next free block and soon.

→ we have to remember the first free block and cached to the main memory.

→ If we know the address of the first free block. then we can easily know the next free blocks.

Ex:- suppose 9 blocks are there in the disk and 2, 4, 7, 8 are free blocks.



→ Linked free space list on disk.

→ By this we cannot get contiguous space Easily.
→ no waste of space.
→ To know which block is free we have to traverse to the ~~total~~ disk total disk and it takes more time to traverse whole disk.

3. Grouping:-
→ Rather than having a pointer to one disk block to another disk block here we have the addresses of n free blocks that are kept in the first free block.

i.e; first free block contain the addresses of all free blocks and if the block is free then it point to the next free block which contain the address of free blocks.

→ Advantage of this method is we can Easily get the large no. of free blocks quickly.

4. Counting:-
In the previous method we put all free blocks addresses in the first free block rather than putting all free blocks addresses we are giving the address of the first free block and say how many blocks are free after this block. contiguous blocks are free after this block.

The first free block and the number of free blocks contiguous free blocks after this free block is maintained.
first

Ex:- If my first free block is 5, and till 10 I have contiguous free blocks and it is represented by

(5, 6)