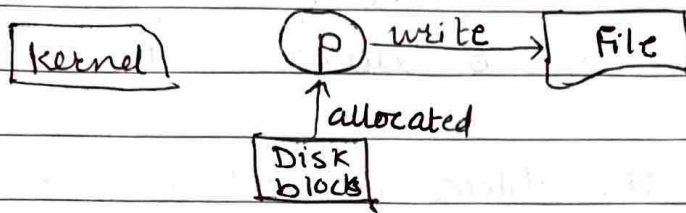
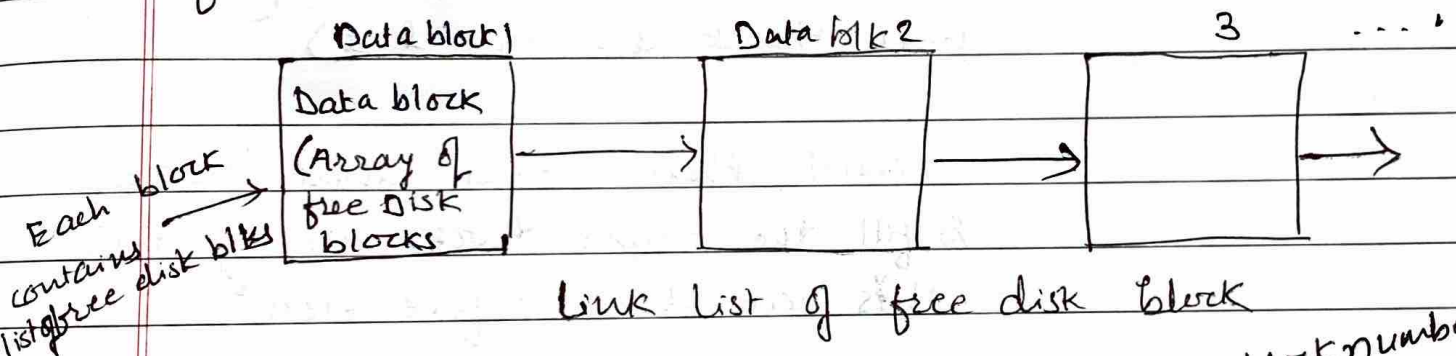


# Allocation of Disk Blocks



\* Whenever process wish to write contents to file, kernel provides free disk blocks

\* To keep track of free disk block, super block maintains an array (mkfs - ~~sys~~ make file system - maintains free disk block in linked list manner)



Link List of free disk block

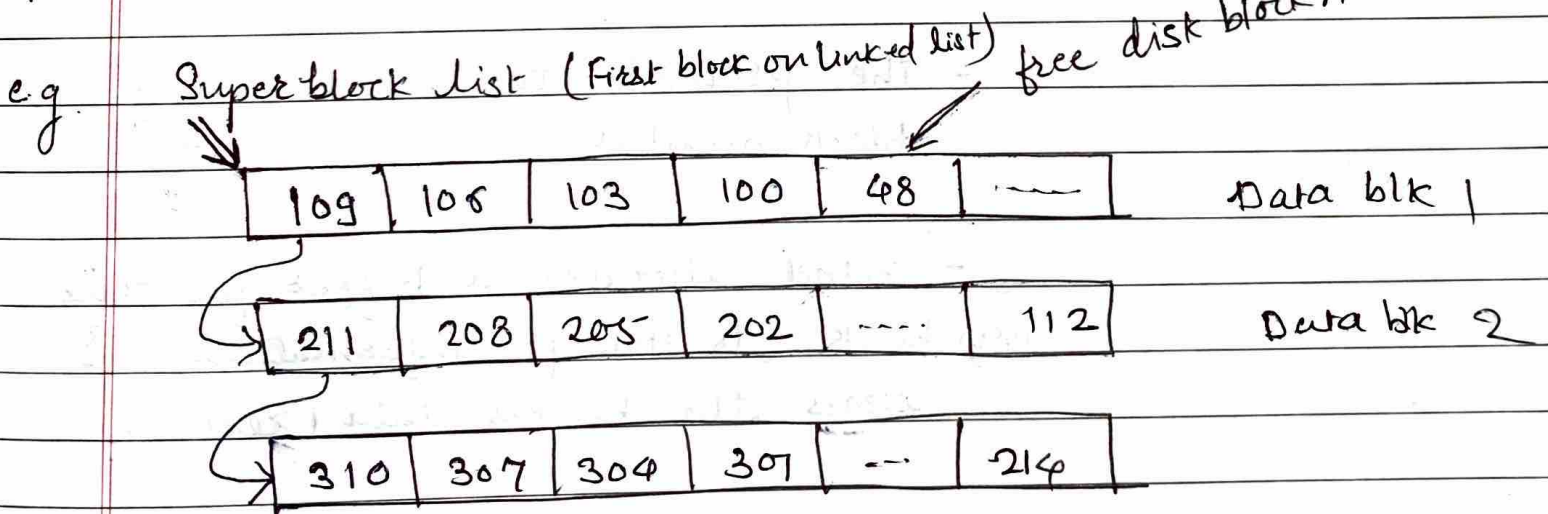


Fig. Original configuration

\* when the kernel wants to allocate a block from file system, it finds the next available block in the super block list & allocates it.

\* Once it is done, the block can not be reallocated until it becomes free

\* For last block - if allocated block is last block in super block cache, the kernel treats it as a ~~se~~ pointer to the block which contains list of free blocks (normally next block in link list)

- kernel then reads that block, & fill the super block array with this new list of free block

- The proceeds to use the original block number.

- kernel allocates a buffer for the (lost) block (to store ptr. information) & clears the buffers data (zero it)

- Now the disk block has been assigned.

\* If there is no free blocks, calling process receives an error

\* kernel assigns only one block at a time even a process demands repeatedly



Super block list

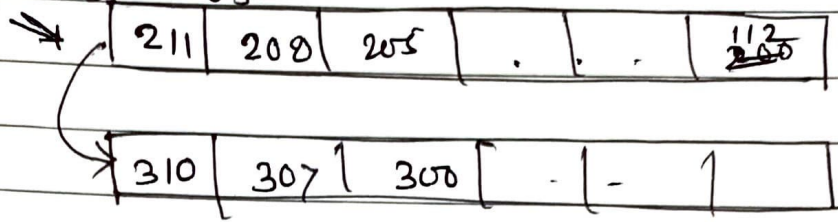
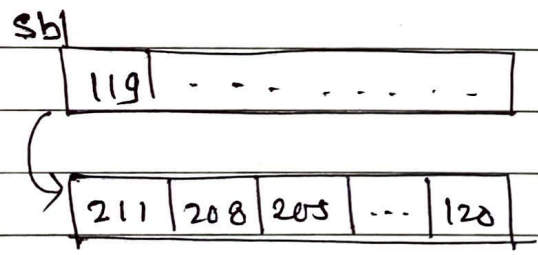


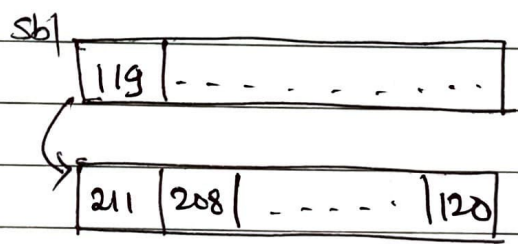
Fig. after allocating block 109 to process.

### Freeing disk block

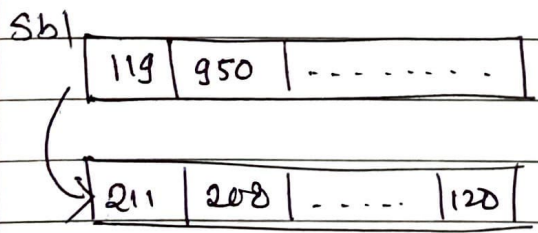
- \* Algorithm for freeing disk block is the reverse of allocating blocks.
- \* If super block list not full  $\rightarrow$  the block number is placed on it.
- \* If super block list is full  $\rightarrow$  newly block becomes link block.



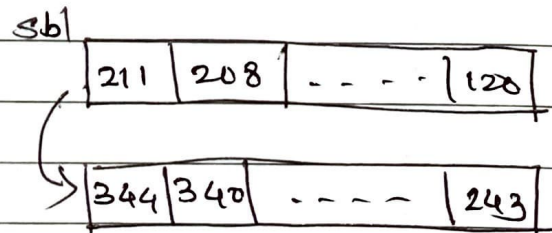
(a) Original configuration



(c) after assigning blk 950



(b) after freeing the block 950



(d) after assigning blk 109