* Differentiate and explain Contiguous file allocation system & linked list allocation

Contiguous Allocation	Linked list allocation
=> Here, each files occupy a contiguous set of blocks on the disk.	=> Here, each file is a linked list of disk blocks, ~~so~~,
=> It is contiguous	=> It is not contiguous
=> Directory entry entry contains:- Address of starting Block	=> Directory entry contains a pointer pointing to starting and ending file block
=> length of allocated portion	=> each Block contains pointer to next block
=> Both seavential and dired access are supported.	=> File Blocks are distributed randomly on disk.
=> Not flexible in terms of file size	=> flexible in terms of file size.

=> It is extremely fast	=> It is slow.
=> It suffers from both internal and external fragmentation	=> It does not suffer from external fragmentation.
=> It support random or direct Access	=> It does not support random access
=> Increasing file size is difficult	=> Increasing file size is easy

(1) ialloc → It loops over the inode structures on the disk, one block at a time, looking for one that is marked free.

(2) namei → It interprets its arguments as pathnames to any type of Unix file.

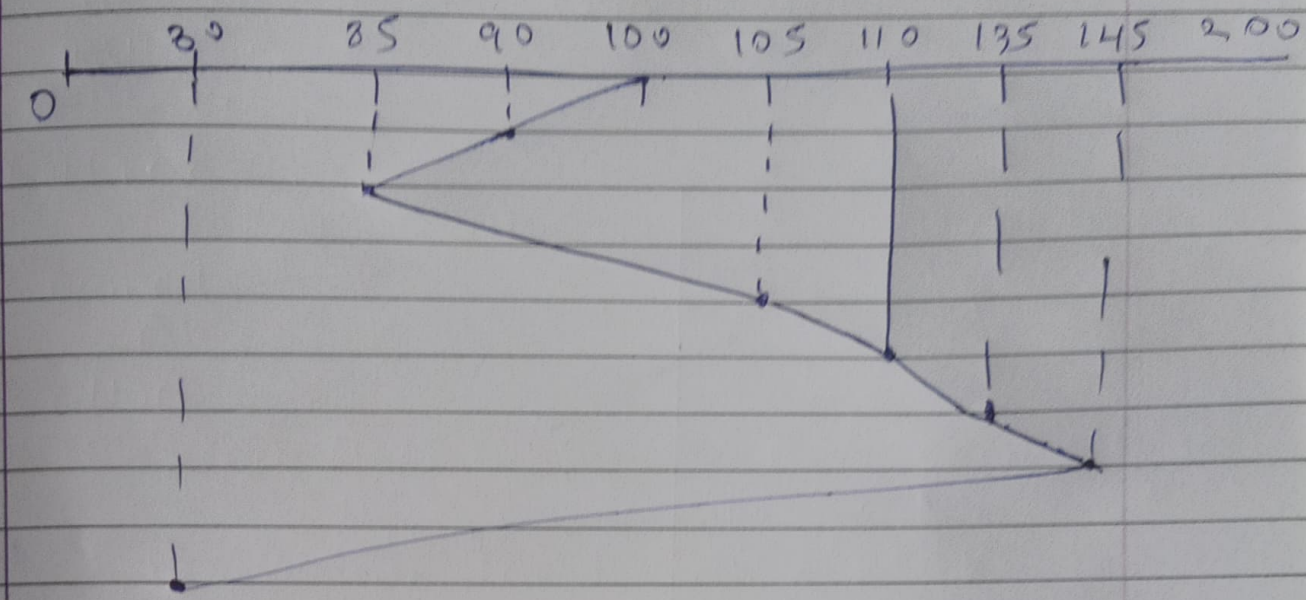
→ Then, it follows each pathname until an endpoint is found.

(3) alloc → It allocates memory within the current function stack frame.

⇒ memory allocated is automatically freed.

(4) ifree → used for creating/deleting file.

(1)



⇒ Total head movement

= (100 - 85) + (145 - 85) + (145 - 30)

= 15 + 60 + 115

= 190

⇒ First request served ⇒ 100, then,

105, then 110 and then arm comes to service request for 90.

⇒ 80; disk would have serviced

3 request