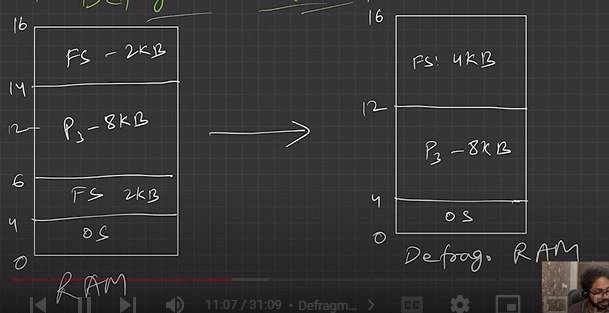
How OS manages Free Space?

Lets say we have process allocated in RAM i.e. P1 [2KB], P2 [8KB], P3[2KB]. Assuming the total ram size as 10 KB. After some time, when P1 and P3 exits leaving 2 +2 = B of space. Now there is new process P4[3KB] trying to enter but it is not able to find continues block of 3 KB to get fitted into the RAM, instad total memory already available is 4Kb.

This causes external fragmentation. How does OS know about the free space recovered after the P1 and P3 leaving. a. Free holes in the memory are represented by a free list (Linked-List data structure

We can solve this using Defragmentation/ Compaction.

We just sort out for the free space and used space



a. Dynamic partitioning suffers from external fragmentation.

b. Compaction to minimize the probability of external fragmentation.

c. All the free partitions are made contiguous, and all the loaded partitions are brought together.

d. By applying this technique, we can store the bigger processes in the memory. The free partitions are merged which can now be allocated according to the needs of new processes. This technique is also called defragmentation.

e. **The efficiency of the system is decreased in the case of compaction since all the free spaces will be transferred from several places to a single place**.

After compaction, the physical address may get changes in relocation register. This will make the memory space needed available

* Algorithms for Process allocations in free hole located using free lost
* Various algorithms which are implemented by the Operating System in order to find out the holes in the linked list and allocate them to the processes.
  + First Fit
    - Allocate the first hole that is big enough.
    - Simple and easy to implement.
    - Fast/Less time complexity
  + Next Fit
    - Enhancement/tweek on First fit but starts search always from last allocated hole.
    - Same advantages of First Fit
  + Best Fit
    - Allocate smallest hole that is big enough.
    - Lesser internal fragmentation.
    - May create many small holes and cause major external fragmentation(**drawback**)
    - Slow, as required to iterate whole free holes list.
  + Worst Fit
    - Allocate the largest hole that is big enough.
    - Slow, as required to iterate whole free holes list.
    - Leaves larger holes that may accommodate other processes.