**What is fragmentation and its uses in OS?**

Fragmentation refers to a process of information storage where the memory space of the system is used inadequately, thus reducing the overall efficiency or ability or both (sometimes). The implications of the process of fragmentation depend entirely on the specific allocation of storage space schemes in the operation along with the particular fragmentation types. In some instances, fragmentation leads to some unused storage capacity. This concept is also applicable to the generated unused space in this very situation.

The memory used for the preservation of the data set (like file formats) is very similar to the other systems (like the FAT file system), irrespective of the amount of fragmentation (it happens from null to the extreme).

**Types of Fragmentation:**

* External Fragmentation
* Internal Fragmentation
* Data Fragmentation (which exists beside or a combination)

Fragmentation can often be acknowledged when preferring enhancements, usability, or inefficiency. Similar things might also happen for the other tools, like processors.

**Internal Fragmentation**

Whenever a memory block gets allocated with a process, and in case the process happens to be smaller than the total amount of requested memory, a free space is ultimately created in this memory block. And due to this, the memory block’s free space is unused. This is what causes internal fragmentation.

**External Fragmentation**

External fragmentation occurs whenever a method of dynamic memory allocation happens to allocate some memory and leave a small amount of unusable memory. The total quantity of the memory available is reduced substantially in case there’s too much external fragmentation. So, there’s enough memory space in order to complete a request, and it is not contiguous. Thus, it is known as external fragmentation.

**Causes of Fragmentation**

The user processes are unloaded and loaded from the main memory. Also, all the processes are kept in the memory blocks in a system’s main memory. Various spaces are left after the loading and swapping of processes that other processes can’t load because of their sizes. The main memory is available, but the space isn’t sufficient in order to load other processes since the allocation of the main memory processes is dynamic.

**Pros and Cons of Fragmentation:**

Pros:

Storage optimization: The fragmented systems might make potentially better use of the storage devices by using every storage block available to us.

Fewer failures: If a system has insufficient sequential space, then it won’t support fragmentation and would ultimately fail. No issues would arise.

Fast data writes: The data written in a system supporting data fragmentation can be faster than the reorganizing data storage. It enables data writes in a contiguous manner.

Cons:

Need for regular defragmentation: The storage device with more fragmentation would ultimately degrade over time. Thus, there would be a necessary requirement for time-consuming operations of defragmentation.

Slower read times: The amount of time taken to read a non-sequential file may increase when a storage device happens to become more fragmented.