**ALLOCATION OF FRAMES**

The main memory of the operating system is divided into various frames. The pages of process is stored in these frames, and once the process is saved as a frame, the CPU may run it. When many processes are kept in the physical memory, the way in which the physical frames are allocated to the different processes is called allocation of frames.

**Minimum number of Frames:**

There are various constraints to the strategies for the allocation of frames:

* You cannot allocate more than the total number of available frames.
* At least a minimum number of frames should be allocated to each process. This limitation is due to two factors. The first is that when the number of frames assigned drops, the page fault ratio grows, decreasing the process's execution performance. Second, there should be sufficient frames to hold all the multiple pages that any instruction may reference.

Frame allocation algorithms are used if you have multiple processes; it helps decide how many frames to allocate to each process.

**Frame allocation algorithms:**

There are mainly five ways of frame allocation algorithms in the OS. These are as follows:

1. Equal Frame Allocation
2. Proportional Frame Allocation
3. Priority Frame Allocation
4. Global Replacement Allocation
5. Local Replacement Allocation

**1)Equal Frame Allocation:**

The easiest way to split m frames among n processes is to give everyone an equal share, m/n frames. For instance, if there are 93 frames and five processes, each process will get 18 frames. The three leftover frames can be used as a free-frame buffer pool. This scheme is called equal frame allocation.

Consider a system with a 1-KB frame size. If a small student process of 10 KB and an interactive database of 127 KB are the only two processes running in a system with 62 free frames, it does not make much sense to give each process 31 frames. The student process does not need more than 10 frames, so the other 21 are, strictly speaking, wasted.

**Disadvantage:** In systems with processes of varying sizes, it does not make much sense to give each process equal frames. Allocation of a large number of frames to a small process will eventually lead to the wastage of a large number of allocated unused frames.

**2)Proportional Frame Allocation:**

To solve this problem, we can use proportional allocation. In proportional allocation, the number of frames allocated to a process is based on the size of the process.

For a process pi of size si, the number of allocated frames is **ai = (si/S)\*m**, where S is the sum of the sizes of all the processes and m is the number of frames in the system.

**Example:** With proportional allocation, we would split 62 frames between two processes, one of 10 pages and one of 127 pages, by allocating 4 frames and 57 frames, respectively, since

10/137 × 62 ≈ 4, and

127/137 × 62 ≈ 57.

In this way, both processes share the available frames according to their “needs,” rather than equally.

**3) Priority Frame Allocation:**

with either equal or proportional allocation, a high-priority process is treated the same as a low-priority process. In this method, the ratio of frames depends not on the relative sizes of processes but rather on the priorities of processes or on a combination of size and priority. Suppose a process has a high priority and requires more frames that many frames will be allocated to it. Following that, lesser priority processes are allocated.

**4) Global Replacement Allocation:**

Another important factor in the way frames are allocated to the various processes is page replacement. With multiple processes competing for frames, we can classify page-replacement algorithms into two broad categories: global replacement and local replacement. Global replacement allows a process to select a replacement frame from the set of all frames, even if that frame is currently allocated to some other process; that is, one process can take a frame from another.

**Advantage:** Does not hinder the performance of processes and hence results in greater system throughput.

**Disadvantage:** This allows a high priority process to select frames from a low priority process. This may increase the number of frames allocated to a high priority process. However, the low priority processes will still suffer from page faults, since the high priority process takes frames from the low priority process. Therefore, the set of pages in memory for a process may depend on the paging behaviour of other processes.

**5) Local Replacement Allocation:**

When a process needs a page which is not in the memory, it can bring in the new page and allocate it a frame from its own set of allocated frames only.

**Advantage:**

The pages in memory for a particular process and the page fault ratio is affected by the paging behaviour of only that process.

**Disadvantage:** A low priority process may hinder a high priority process by not making its frames available to the high priority process.

Global replacement gives better throughput and hence, global replacement is commonly used.