| **First in First out (FIFO)** | **Optimal Page Replacement** | **Least Recently Used (LRU)** |
| --- | --- | --- |
| The operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced, the page in the front of the queue is selected for removal. | The pages are replaced with the ones that will not be used for the longest duration of time in the future. | It removes the page that has not been utilized in the memory for the longest period of time. It replaces the least frequently used pages. |
| **Advantages**:   * It is simple and easy to understand and implement. * It is efficiently used for small systems. * It does not cause more overheads. | **Advantages:**     * Reduce the no of page faults. * This algorithm is easy to use. * This algorithm provides excellent efficiency and is less complex. * For the best result, the implementation of data structures is very easy. | **Advantages:**   * It takes into account the usage history of pages, and it can lead to fewer page faults and faster application response times. * Less susceptible to thrashing. * efficient use of memory because it replaces the page that has not been used for the longest time. |
| **Disadvantages:**   * The process effectiveness is low. * When we increase the number of frames while using FIFO, we are giving more memory to processes. * There is an increase in page faults as page frames increase. | **Disadvantages:**   * In this algorithm future awareness of the program is needed. * Practical Implementation is not possible because the operating system is unable to track the future request. | **Disadvantages:**   * It requires additional Data Structure to be implemented. * Hardware assistance is high. * Error detection is difficult as compared to other algorithms. * It has limited acceptability. * Very costly to operate. |