**Replacement Algorithms (Policies) :**

Basic to the implementation of virtual memory is the concept of demand paging. This means that the operating system, and not the programmer, controls the swapping of pages in and out of main memory as they are required by the active processes.

**replacement policy:**  A technique used in the virtual memory that makes a decision When a process needs a nonresident page, the operating system must decide which resident page is to be replaced by the requested page.

To illustrate the use of the **FIFO** mechanism, we offer the following example

**Example** :Consider the following reference string of pages made by a processor:

6, 7, 8, 9, 7, 8, 9, 10, 8, 9, 10. In particular, consider two cases: (a) the number of page frames allocated in the main memory is **TWO** and (b) the number of page frames allocated are **THREE**.

The figure below illustrates a trace of the reference string for the two cases. As can be seen from the figure, when the number of page frames is TWO, there were **11 page faults** (these are shown in bold in the figure).

When the number of page frames is increased to THREE, the number of page



faults was reduced to **five**. Since five pages are referenced, this is the optimum condition.

Least Recently Used (**LRU**) Replacement According to this technique, page

replacement is based on the pattern of usage of a given page residing in the main memory regardless of the time( spent )in the main memory. The page that has not been( referenced) for the longest time while residing in the main memory is selected for replacement. To illustrate the use of the LRU mechanism, we offer the following example.

**Example:** Consider the following reference string of pages made by a processor:

4, 7, 5, 7, 6, 7, 10, 4, 8, 5, 8, 6, 8, 11, 4, 9, 5, 9, 6, 9, 12, 4, 7, 5, 7. Assume that the number of page frames allocated in the main memory is **FOUR**. Compute the number of page faults generated. The trace of the main memory contents is shown in Figure below. Number of **page faults = 18.**

In presenting the **LRU**, we have a particular implementation, called stack-based LRU. In this implementation, the most recently accessed page is now represented by



the top page rectangle. The rectangles do not represent specific page frames as they did in the FIFO diagram. Thus, each reference generating a page fault is now on the top row.

**H.W:**