GROUP – B

EXPERIMENT NO- 2

##### Title:

Write a program to simulate Page replacement algorithm.

##### Objectives :

* + To understand Page replacement policies
  + To understand paging concept
  + To understand Concept of page fault, page hit, miss, hit ratio etc

##### Problem Statement :

Write a java program to implement Page Replacement Policies LRU & OPT..

##### Outcomes:

After completion of this assignment students will be able to:

* + Knowledge of Page Replacement Policies in OS
  + Implemented LRU & OPT Page replacement Policies
  + Understood concept of paging.

##### Software Requirements:

Latest jdk., Eclipse

##### Hardware Requirement:

* + M/C Lenovo Think center M700 Ci3,6100,6th Gen. H81, 4GB RAM ,500GB HDD

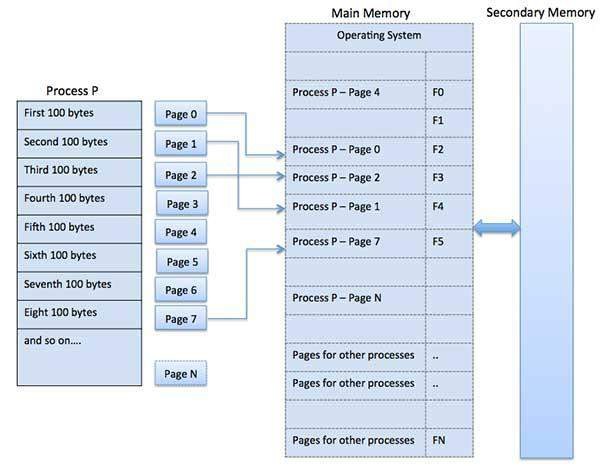
1. **Theory Concepts**:

**Paging :**

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard that's set up to emulate the computer's RAM. Paging technique plays an important role in implementing virtual memory.

Paging is a memory management technique in which process address space is broken into blocks of the same size called **pages** (size is power of 2, between 512 bytes and 8192 bytes). The size of the process is measured in the number of pages.

Similarly, main memory is divided into small fixed-sized blocks of (physical) memory called **frames** and the size of a frame is kept the same as that of a page to have optimum utilization of the main memory and to avoid external fragmentation.



Address Translation

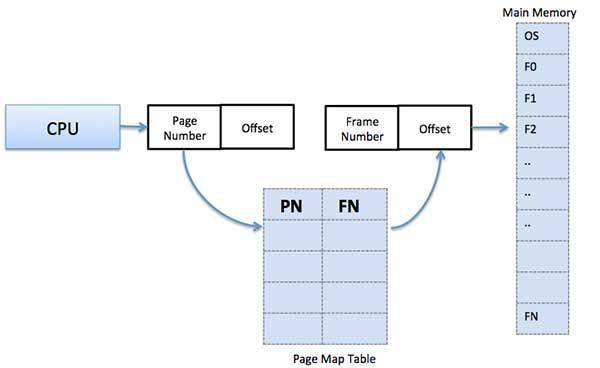
Page address is called **logical address** and represented by **page number**and the **offset**.

Logical Address = Page number + page offset

Frame address is called **physical address** and represented by a **frame number** and the **offset**.

Physical Address = Frame number + page offset

A data structure called **page map table** is used to keep track of the relation between a page of a process to a frame in physical memory



When the system allocates a frame to any page, it translates this logical address into a physical address and create entry into the page table to be used throughout execution of the program.

When a process is to be executed, its corresponding pages are loaded into any available memory frames. Suppose you have a program of 8Kb but your memory can accommodate only 5Kb at a given point in time, then the paging concept will come into picture. When a computer runs out of RAM, the operating system (OS) will move idle or unwanted pages of memory to secondary memory to free up RAM for other processes and brings them back when needed by the program.

This process continues during the whole execution of the program where the OS keeps removing idle pages from the main memory and write them onto the secondary memory and bring them back when required by the program.

##### Advantages and Disadvantages of Paging

Here is a list of advantages and disadvantages of paging −

* Paging reduces external fragmentation, but still suffer from internal fragmentation.
* Paging is simple to implement and assumed as an efficient memory management technique.
* Due to equal size of the pages and frames, swapping becomes very easy.
* Page table requires extra memory space, so may not be good for a system having small RAM.

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called **virtual memory** and it is a section of a hard disk that's set up to emulate the computer's RAM.

The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

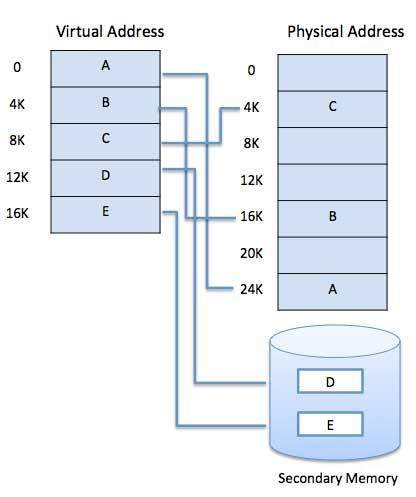
Following are the situations, when entire program is not required to be loaded fully in main memory.

* User written error handling routines are used only when an error occurred in the data or computation.
* Certain options and features of a program may be used rarely.
* Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.
* The ability to execute a program that is only partially in memory would counter many benefits.
* Less number of I/O would be needed to load or swap each user program into memory.

A program would no longer be constrained by the amount of physical memory that is available.

* Each user program could take less physical memory, more programs could be run the same time, with a corresponding increase in CPU utilization and throughput.

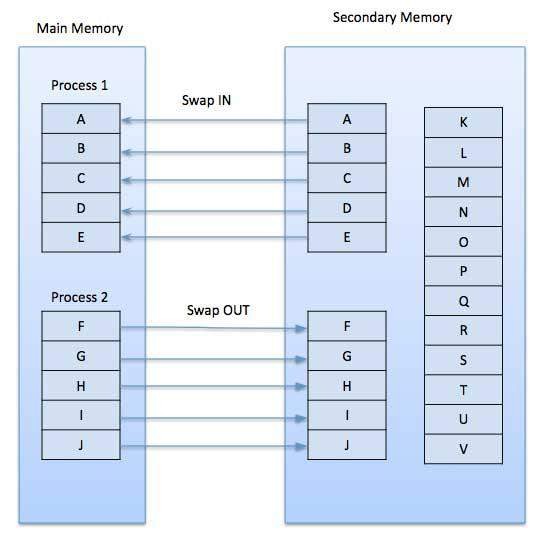
Modern microprocessors intended for general-purpose use, a memory management unit, or MMU, is built into the hardware. The MMU's job is to translate virtual addresses into physical addresses. A basic example is given below –



Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system. Demand segmentation can also be used to provide virtual memory.

##### Demand Paging

A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program’s pages out to the disk or any of the new program’s pages into the main memory Instead, it just begins executing the new program after loading the first page and fetches that program’s pages as they are referenced.



While executing a program, if the program references a page which is not available in the main memory because it was swapped out a little ago, the processor treats this invalid memory reference as a **page fault** and transfers control from the program to the operating system to demand the page back into the memory.

##### Advantages

Following are the advantages of Demand Paging −

* + Large virtual memory.
  + More efficient use of memory.
  + There is no limit on degree of multiprogramming.

##### Disadvantages

* + Number of tables and the amount of processor overhead for handling page interrupts are greater than in the case of the simple paged management techniques.

##### Page Replacement Algorithm:

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose accounting to reason that pages are not available or the number of free pages is lower than required pages.

When the page that was selected for replacement and was paged out, is referenced again, it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm: the lesser the time waiting for page-ins, the better is the algorithm.

A page replacement algorithm looks at the limited information about accessing the pages provided by hardware, and tries to select which pages should be replaced to minimize the total number of page misses, while balancing it with the costs of primary storage and processor time of the algorithm itself. There are many different page replacement algorithms. We evaluate an algorithm by running it on a particular string of memory reference and computing the number of page faults,

##### Page fault :

A **page fault** (sometimes called #PF, PF or hard **fault**) is a type of exception raised by computer hardware when a running program accesses a memory **page**that is not currently mapped by the memory management unit (MMU) into the virtual address space of a process.

##### Page hit :

A **hit** is a request to a web server for a file, like a web **page**, image, JavaScript, or Cascading Style Sheet. When a web **page** is downloaded from a server the number of "**hits**" or "**page hits**" is equal to the number of files requested.

##### Page frame :

The **page frame** is the storage unit (typically 4KB in size) whereas the **page** is the contents that you would store in the storage unit ie the **page frame**. For eg) the RAM is divided into fixed size blocks called **page frames** which is typically 4KB in size, and each **page frame** can store 4KB of data ie the **page**.

##### Page table :

A **page table** is the data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses.

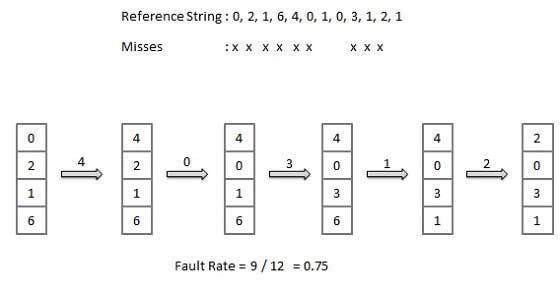
##### Reference String :

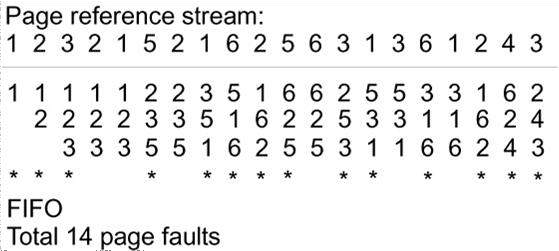
The string of memory references is called reference string. Reference strings are generated artificially or by tracing a given system and recording the address of each memory reference. The latter choice produces a large number of data, where we note two things.

* For a given page size, we need to consider only the page number, not the entire address.
* If we have a reference to a page **p**, then any immediately following references to page **p** will never cause a page fault. Page p will be in memory after the first reference; the immediately following references will not fault.
* For example, consider the following sequence of addresses − 123,215,600,1234,76,96
* If page size is 100, then the reference string is 1,2,6,12,0,0

##### First In First Out (FIFO) algorithm:

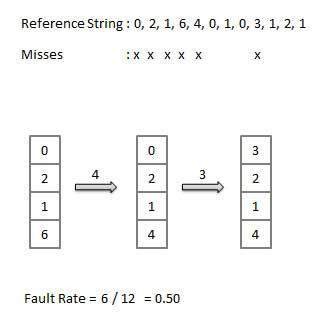
* Oldest page in main memory is the one which will be selected for replacement.
* Easy to implement, keep a list, replace pages from the tail and add new pages at the head.

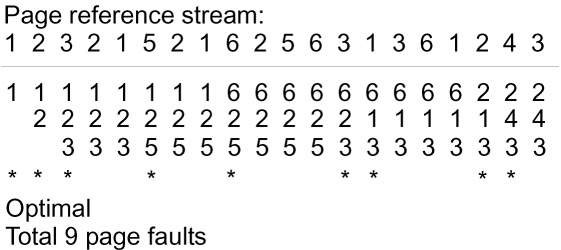




##### Optimal Page algorithm:

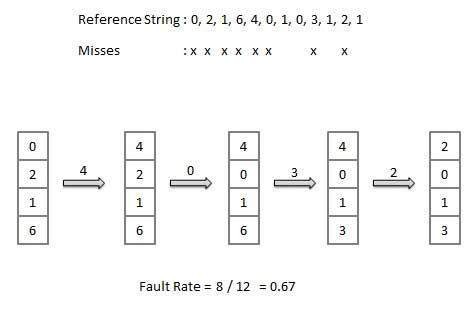
* An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms. An optimal page-replacement algorithm exists, and has been called OPT or MIN.
* Replace the page that will not be used for the longest period of time. Use the time when a page is to be used.

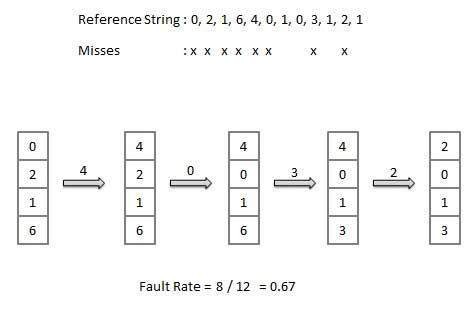


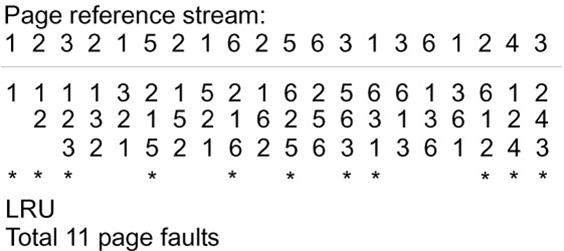


**Least Recently Used (LRU) algorithm:**

* Page which has not been used for the longest time in main memory is the one which will be selected for replacement.
* Easy to implement, keep a list, replace pages by looking back into time.







**Page Buffering algorithm**

* To get a process start quickly, keep a pool of free frames.
* On page fault, select a page to be replaced.
* Write the new page in the frame of free pool, mark the page table and restart the process.
* Now write the dirty page out of disk and place the frame holding replaced page in free pool.

##### Least frequently Used(LFU) algorithm

* The page with the smallest count is the one which will be selected for replacement.
* This algorithm suffers from the situation in which a page is used heavily during the initial phase of a process, but then is never used again.

##### Most frequently Used(MFU) algorithm

* This algorithm is based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

##### Conclusion :

Thus , I have implemented page replacement policies `LRU and OPT.