

### **Memory Management**

#### **1. Implement Following Page Replacement Algorithm:**

- a. FIFO
- b. LRU
- c. OPTIMAL
- d. SECOND CHANCE
- e. CLOCK
- f. NFU
- g. MFU

#### **2. Create a simulator for memory management algorithm.**

The main memory must accommodate both operating system and various user processes. the memory is usually divided into 2 partitions, one for resident operating system and one for user process. one of the simplest scheme for memory allocation is to divide memory into a number of fixed size partitions. In general there is at any time a set of holes of various sizes scattered throughout the memory. The set of holes is searched to determine which hole is best to allocate. First Fit Allocates the first hole that is big enough. Searching can be started either at the beginning or where the previous first fit search end read. Best Fit allocates the smallest hole that is big enough. We must search the entire list , unless the list is kept ordered by size. Worst Fit Allocates to the largest hole.

Read the memory partitions number and size of each partition. Read the size of memory needed by the process. If now memory partition is greater than size , print no memory free. Else read the choice. In the First Fit , the first memory partition which is greater than size is allocated. In Best Fit, the memory partition with minimum arr[] size is allocated. In Worst fit , the largest memory partition is allocated.

#### **3. Simulate Paging technique of Memory Management.**

Enter no. of pages and the page size for each page, use malloc() function for memory allocation and print address of each page.