Experiment No.9
Implement page replacement policy FIFO.
Date of Performance:
Date of Submission:

Aim: To study and implement page replacement policy FIFO.

Objective: Page replacement algorithms are an important part of virtual memory management and it helps the OS to decide which memory page can be moved out, making space for the currently needed page. However, the ultimate objective of all page replacement algorithms is to reduce the number of page faults.

Theory:

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.

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.

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.

First In First Out (FIFO) algorithm

The page which has been loaded first in the main memory is selected first for replacement from the main memory.

Result:

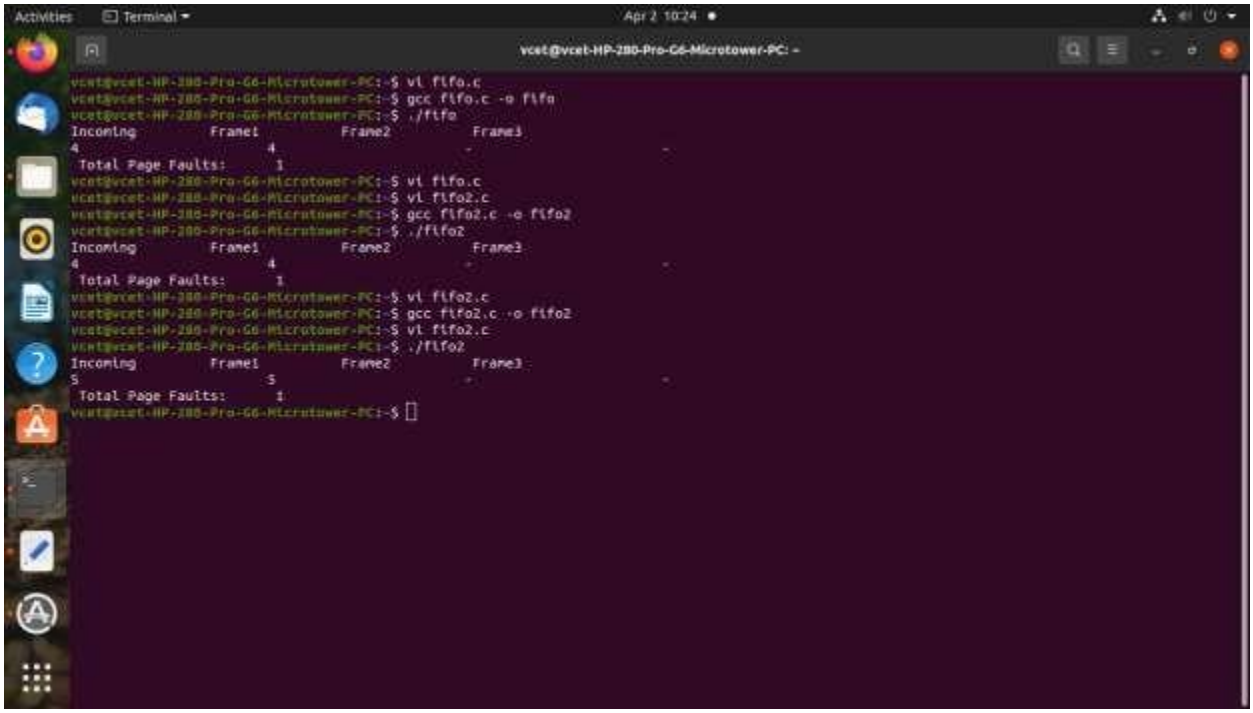
CODE

```
#include <stdio.h>

int main()

{
```

```
int incomingStream[]={3,1,4,5,2};
int pageFaults=0;
int frames=3;
int m,n,s,pages;
pages=sizeof(incomingStream)/sizeof(incomingStream[0]);
printf("Incoming\t Frame1 \t Frame2 \t Frame3");
int temp[frames];
for(m=0;m<frames;m++)
{
        temp[m]=-1;
}
for(n=0;n<frames;n++)
{
        s=0;
        for(n=0;n<frames;n++)
        {
              if(incomingStream[m]==temp[n])
              {
                    s++;
                    pageFaults;
              }
         }
        pageFaults++;
```

```
            if((pageFaults<-frames) && (s==0))
            {
            temp[m]=incomingStream[m];
}
            else if(s==0)
            {
                  temp[(pageFaults-1)%frames]=incomingStream[m];
            }
            printf("\n");
            printf("%d\t\t\t",incomingStream[m]);
            for(n=0;n<frames;n++)
            {
                  if(temp[n]!=-1)
                      printf("%d\t\t\t",temp[n]);
                  else
                    printf("-\t\t\t");
            }
            }
            printf("\n Total Page Faults:\t %d \n",pageFaults);
            return 0;
}
```

OUTPUT:



- **Conclusion:** FIFO page replacement algorithm in os is responsible for keeping track of all the pages in a queue**.** FIFO page replacement algorithm is a good choice When the number of incoming pages is few. Otherwise, It's not the best replacement algorithm to use practically.