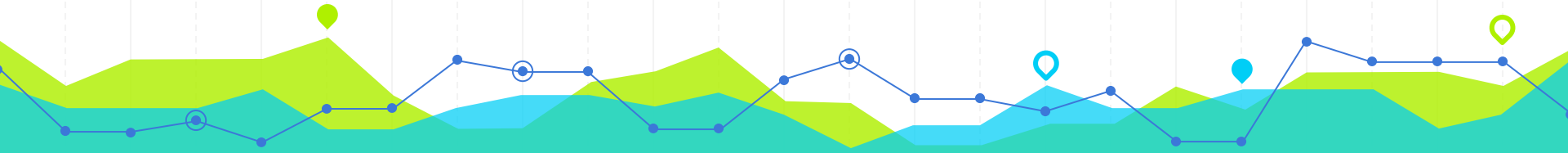


OS
Lab Assignment - 5

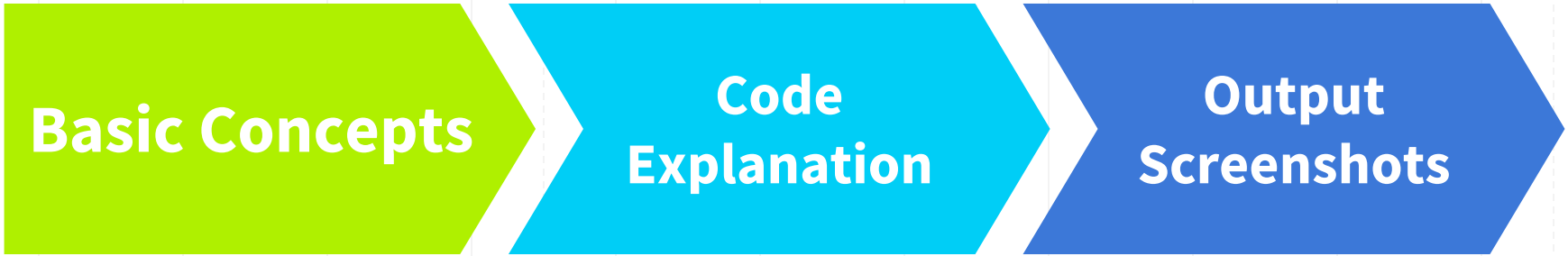
Group - 6

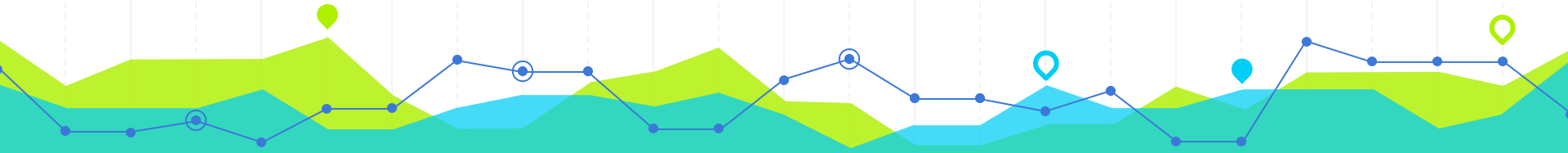
Abhijeet Sonkar	(IIB2019009)
Avneesh Kumar	(IIB2019010)
Ambika Singh	(IIB2019017)
Aditya Aggarwal	(IIT2019210)
Divy Agrawal	(IIT2019211)

Question Statement

Implement the code of the page replacement algorithm LRU Approximation: Enhanced Second Chance Algorithm inside the getFrameNo() function and print the value of total page-fault-count from the Finalize() function.

Flow of Content





1
Basic Concepts

Virtual Memory

- An imaginary memory area supported by some operating systems in conjunction with the hardware
- Programs use these virtual addresses rather than real addresses to store instructions and data. When the program is actually executed, the virtual addresses are converted into real memory addresses.
- The purpose of virtual memory is to enlarge the address space, the set of addresses a program can utilize. For example, virtual memory might contain twice as many addresses as main memory

Page Replacement Algorithm

- Page replacement algorithms decide which memory pages to page out, sometimes called swap out, or write to disk, when a page of memory needs to be allocated.
- Page replacement happens when a requested page is not in memory (page fault) and a free page cannot be used to satisfy the allocation, either because there are none, or because the number of free pages is lower than some threshold.

Paging

- Paging is a memory management scheme by which a computer stores and retrieves data from secondary storage for use in main memory. In this scheme, the operating system retrieves data from secondary storage in same-size blocks called pages.
- Paging is an important part of virtual memory implementations in modern operating systems, using secondary storage to let programs exceed the size of available physical memory

Least Recently Used (LRU)

- Least Recently Used page replacement algorithm keeps track of page usage over a short period of time. It works on the idea that the pages that have been most heavily used in the past are most likely to be used heavily in the future too.
- In LRU, whenever page replacement happens, the page which has not been used for the longest amount of time is replaced.

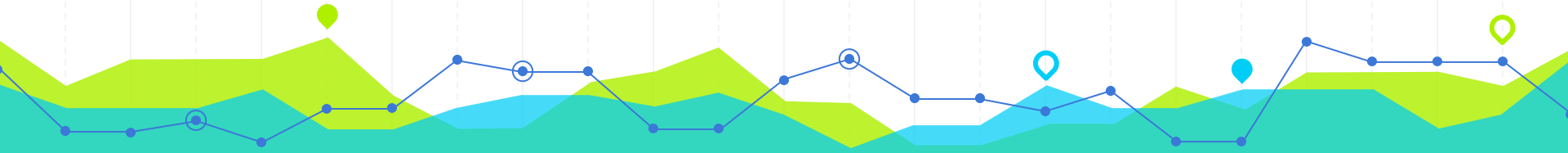
Enhanced Second Chance Algo

The reference bit and modify bit form a pair (r,m) where -

- ❑ (0,0) neither recently used nor modified - replace this page!
- ❑ (0,1) not recently used but modified - not as good to replace, since the OS must write out this page, but it might not be needed anymore.
- ❑ (1,0) recently used and unmodified - probably will be used again soon, but OS need not write it out before replacing it
- ❑ (1,1) recently used and modified - probably will be used again soon and the OS must write it out before replacing.
- ➢ On a page fault, the OS searches for the first page in the lowest nonempty class.

The OS goes around at most three times searching for the (0,0) class.

- ❑ Page with (0,0) => replace the page.
- ❑ Page with (0,1) => initiate an I/O to write out the page, locks the page in memory until the I/O completes, clears the modified bit, and continue the search.
- ❑ For pages with the reference bit set, the reference bit is cleared.
- ❑ If the hand goes completely around once, there was no (0,0) page. • On the second pass, a page that was originally (0,1) or (1,0) might have been changed to (0,0) => replace this page
- ➢ If the page is being written out, waits for the I/O to complete and then remove the page.
- ➢ A (0,1) page is treated as on the first pass.
- ➢ By the third pass, all the pages will be at (0,0).



2
Code Explanation

```
void create_queue(){
  queue=create_node();
  struct node *start=queue;
  for(int i=0;i<no_of_frames-1;i++){
    start->next=create_node();
    start=start->next;
  }
  start->next=queue;
}
```

This function is used to create the circular queue using linked list which is in turn used to implement page replacement algorithm.

This function is used to create a node for implementing a circular queue which is in turn used to implement page replacement algorithm.

```
struct node* create_node(){
  struct node *new=NULL;
  new=(struct node*)malloc(sizeof(struct node));
  new->page_no=-1;
  new->next=NULL;
  return new;
}
```

```
int SwapOut(int fno,int old_pno){
  if(PageTable[old_pno].modify_bit)
    writeFrame(fno,old_pno);
  PageTable[old_pno].modify_bit = 0;
  PageTable[old_pno].reference_bit = 0;
  return old_pno;
}
```

If the page which is to be replaced has a modify bit 1 then the page needs to be swapped out that means the data of the page has been modified and needs to be written back to file before bringing in a page which would be replacing it.

```
int getFrameNo(int pno){
  int fno;
  if(PageTable[pno].valid_bit){
    PageTable[pno].reference_bit=1;
    return PageTable[pno].frm_no;
  }
```

In this if condition we will check whether the page which is required is already present in one of the frames or not. If it is present in one of the frame then we will return that frame number.

If the page is not present in one of the frames then if there are some frames which are empty then we will bring the required page into this empty frame, valid bit corresponding to this page is set to 1 and return the number of this frame.

```
int i;
  for(i=0;i<no_of_frames;i++){
    if(FrameTable[i]==-1){
      fno=i;
      AddNewPage(queue,pno,fno);
      queue=queue->next;
      break;
    }
  }
```

```
if(i==no_of_frames){
  int victim_page=LRUEnchancedSecondChanceAlgorithm(pno);
  fno=PageTable[victim_page].frm_no;
  PageTable[victim_page].frm_no=-1;
}
printf("Alloted Frame Number=%d\n",fno);
readPage(pno, fno);
PageTable[pno].valid_bit=1;
return fno;
```

If no free frame is available then one of the pages stored in the frame needs to be replaced. The page(victim page) which needs to be replaced is determined by the function LRUEnchancedSecondChanceAlgorithm(pno). When the page which needs to be replaced is determined then the required page will be brought into the frame and the valid bit corresponding to it will be set to 1 and that of the victim page to 0 and its frame number to -1.

```
int LRUEnchancedSecondChanceAlgorithm(int pno){
  int victim_page=check00(pno);
  if(victim_page!=-1){
    return SwapOut(PageTable[victim_page].frm_no,victim_page);
  }
  victim_page=check01(pno);
  if(victim_page!=-1){
    return SwapOut(PageTable[victim_page].frm_no,victim_page);
  }
  LRUEnchancedSecondChanceAlgorithm(pno);
}
```

This function is used to find the appropriate page which will be replaced by the incoming page in one of the frame. Here we will

1. find a page having reference bit and modify bit as 0 for replacement
2. If above page not found then find a page having reference bit 0 and modify bit 1 for replacement.
3. We will repeat the two steps until one of the two situation meets.

This function will be used to bring the required page into the main memory by allotting one frame to it and storing the information in PageTable about the new page.

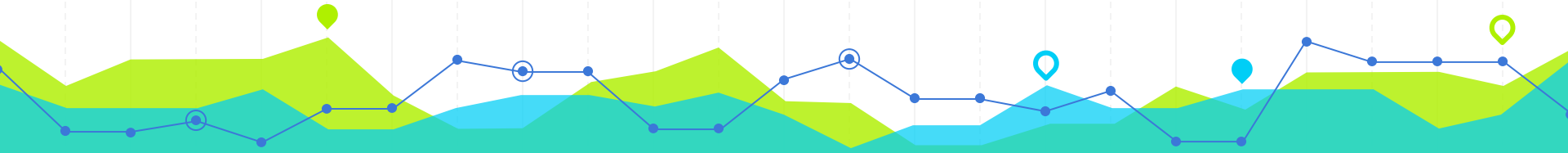
```
void AddNewPage(struct node *x,int new_page,int fram
e){
  x->page_no=new_page;
  PageTable[new_page].frm_no=frame;
  PageTable[new_page].reference_bit=1;
  PageTable[new_page].modify_bit=0;
}
```

```
int check00(int new_page){
  int reference=0,modify=0,page_no=-1;
  if(PageTable[queue->page_no].reference_bit==reference &&
            PageTable[queue->page_no].modify_bit==modify){
    page_no=queue->page_no;
    PageTable[page_no].valid_bit = 0;
    AddNewPage(queue,new_page,PageTable[page_no].frm_no);
    queue=queue->next;
    return page_no;
  }
  pointer=queue->next;
  int n=no_of_frames;
  for(int i=0;i<n-1;i++){
    int p=pointer->page_no;
    if(PageTable[p].reference_bit==reference &&
                        PageTable[p].modify_bit==modify){
      page_no=p;
      PageTable[page_no].valid_bit = 0;
      AddNewPage(pointer,new_page,PageTable[p].frm_no);
      break;
    }
    pointer=pointer->next;
  }
  return page_no;
}
```

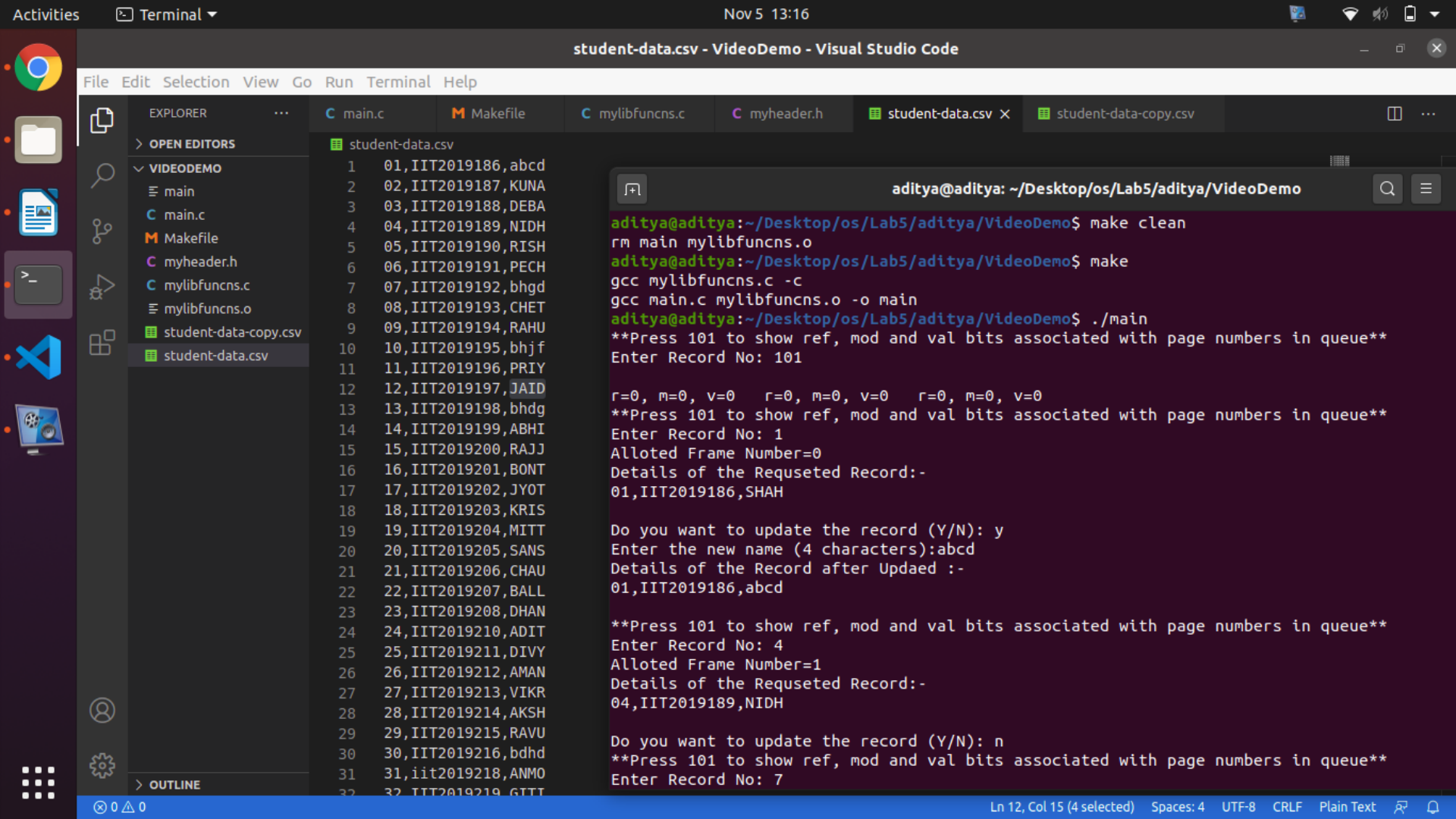
This function Is used to find a page in the circular queue having both reference bit and modify bit as 0. we will iterate until either we find the required page or reach the same position.

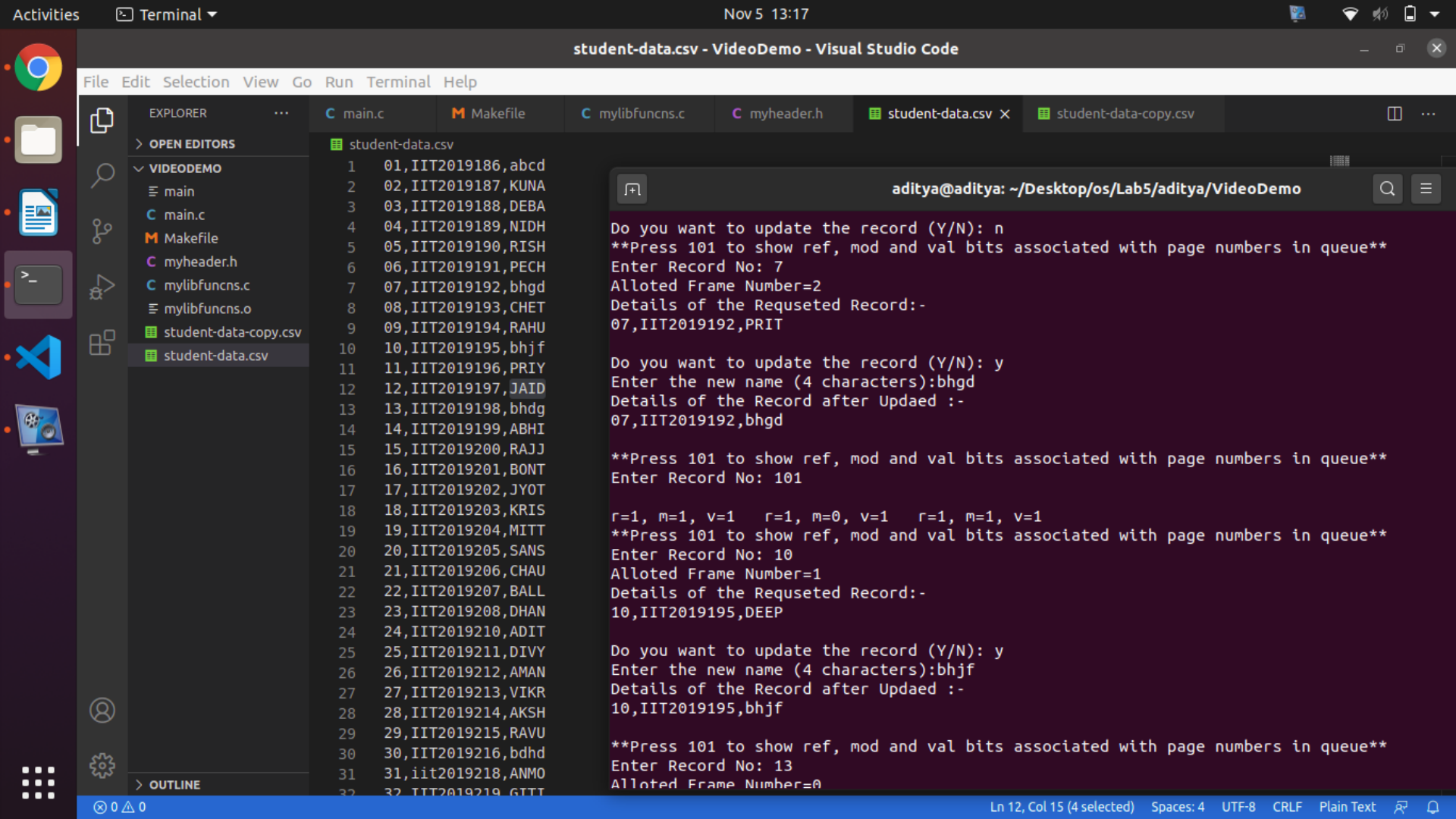
```
int check01(int new_page){
  int reference=0,modify=1,page_no=-1;
  int n=no_of_frames;
  for(int i=0;i<n;i++){
    int p=queue->page_no;
    if(PageTable[p].reference_bit==reference && PageTable[p].modify_bit==modify){
      page_no=p;
      PageTable[page_no].valid_bit = 0;
      AddNewPage(queue,new_page,PageTable[p].frm_no);
      queue=queue->next;
      break;
    }
    PageTable[p].reference_bit=0;
    queue=queue->next;
  }
  return page_no;
}
```

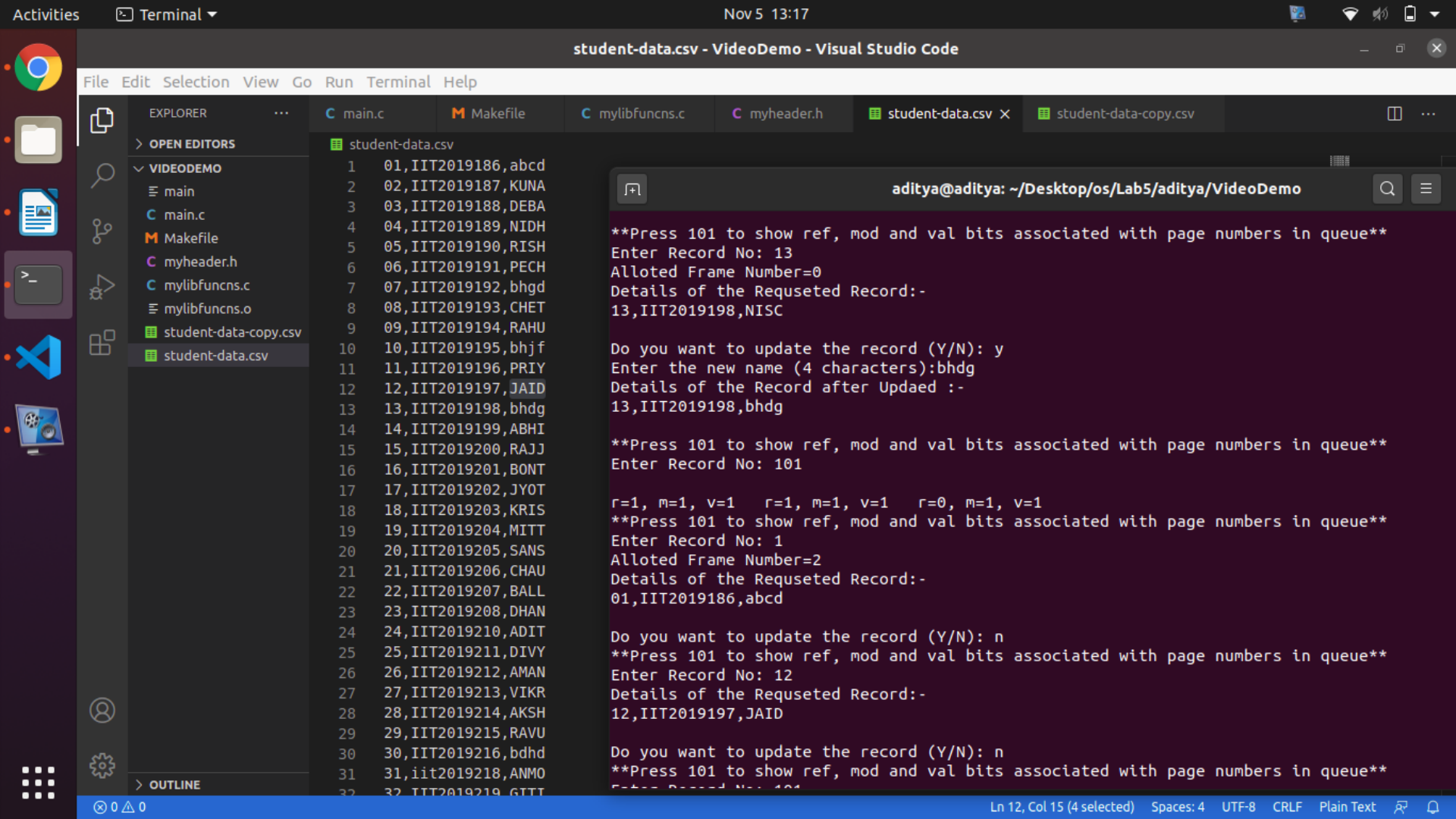
This function will be called when no page is found to be replaced from frames having both modify bit and reference bit as 0. so now we will find a page having reference bit 0 and modify bit 1 in the circular queue. And while iterating if a corresponding page does not has reference bit 0 then we would set its reference bit to 0 and move further until we find the required page.

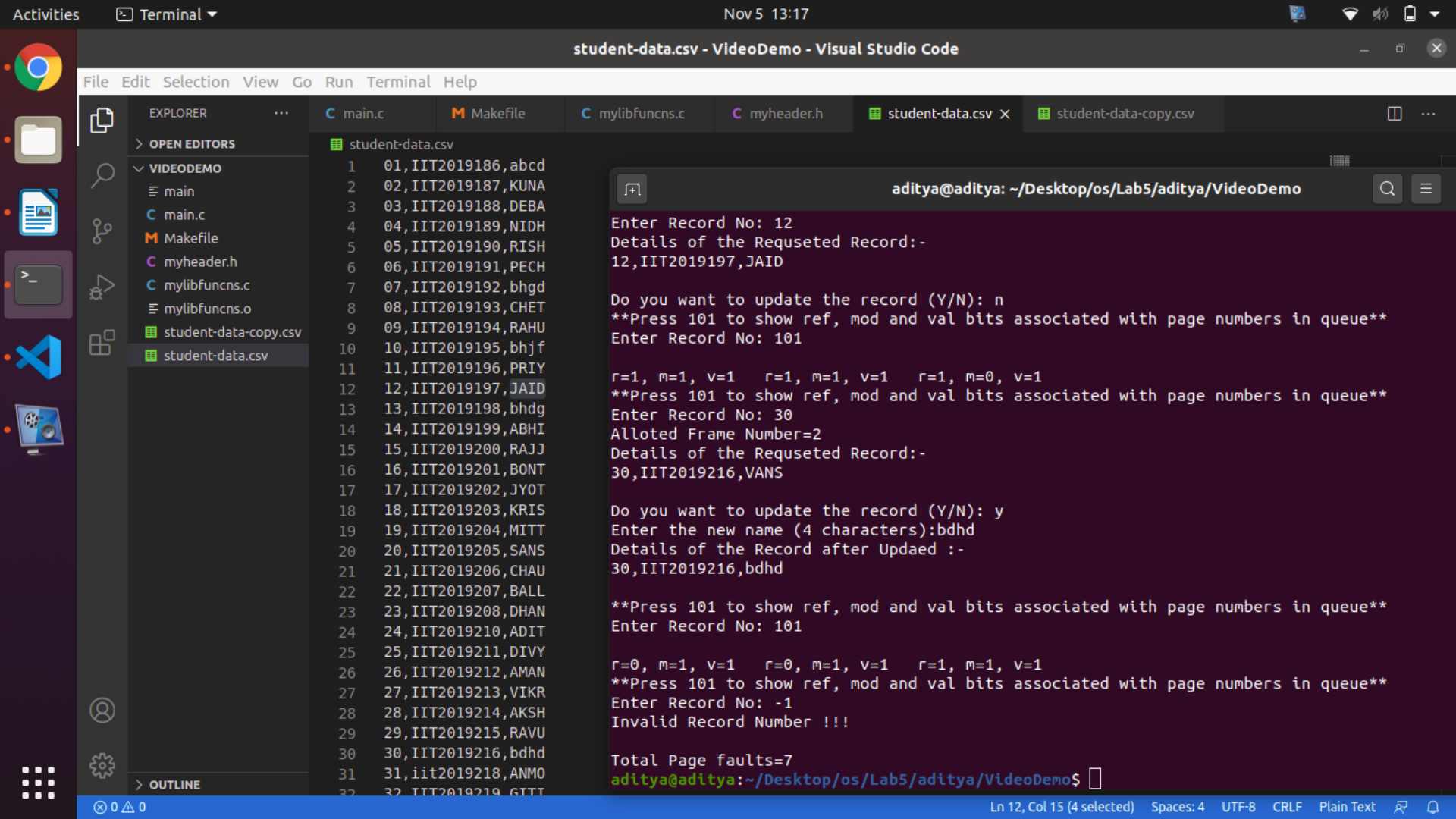


3
Output Screenshot


Activities
Terminal
Nov 5 13:16
student-data.csv - VideoDemo - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
VIDEODEMO
main
main.c
Makefile
myheader.h
mylibfuncns.c
mylibfuncns.o
student-data-copy.csv
student-data.csv
OUTLINE
main.c
Makefile
mylibfuncns.c
myheader.h
student-data.csv
student-data-copy.csv
student-data.csv
1 01,IIT2019186,abcd
2 02,IIT2019187,KUNA
3 03,IIT2019188,DEBA
4 04,IIT2019189,NIDH
5 05,IIT2019190,RISH
6 06,IIT2019191,PECH
7 07,IIT2019192,bhgd
8 08,IIT2019193,CHET
9 09,IIT2019194,RAHU
10 10,IIT2019195,bhjf
11 11,IIT2019196,PRIY
12 12,IIT2019197,JAID
13 13,IIT2019198,bhdg
14 14,IIT2019199,ABHI
15 15,IIT2019200,RAJJ
16 16,IIT2019201,BONT
17 17,IIT2019202,JYOT
18 18,IIT2019203,KRIS
19 19,IIT2019204,MITT
20 20,IIT2019205,SANS
21 21,IIT2019206,CHAU
22 22,IIT2019207,BALL
23 23,IIT2019208,DHAN
24 24,IIT2019210,ADIT
25 25,IIT2019211,DIVY
26 26,IIT2019212,AMAN
27 27,IIT2019213,VIKR
28 28,IIT2019214,AKSH
29 29,IIT2019215,RAVU
30 30,IIT2019216,bdhd
31 31,iit2019218,ANMO
aditya@aditya: ~/Desktop/os/Lab5/aditya/VideoDemo
aditya@aditya:~/Desktop/os/Lab5/aditya/VideoDemo$ make clean
rm main mylibfuncns.o
aditya@aditya:~/Desktop/os/Lab5/aditya/VideoDemo$ make
gcc mylibfuncns.c -c
gcc main.c mylibfuncns.o -o main
aditya@aditya:~/Desktop/os/Lab5/aditya/VideoDemo$ ./main
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 101
r=0, m=0, v=0 r=0, m=0, v=0 r=0, m=0, v=0
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 1
Alloted Frame Number=0
Details of the Requseted Record:-
01,IIT2019186,SHAH
Do you want to update the record (Y/N): y
Enter the new name (4 characters):abcd
Details of the Record after Updaed :-
01,IIT2019186,abcd
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 4
Alloted Frame Number=1
Details of the Requseted Record:-
04,IIT2019189,NIDH
Do you want to update the record (Y/N): n
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 7
Ln 12, Col 15 (4 selected)
Spaces: 4
UTF-8
CRLF
Plain Text


Activities
Terminal
Nov 5 13:17
student-data.csv - VideoDemo - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
VIDEODEMO
main
main.c
Makefile
myheader.h
mylibfuncns.c
mylibfuncns.o
student-data-copy.csv
student-data.csv
OUTLINE
main.c
Makefile
mylibfuncns.c
myheader.h
student-data.csv
student-data-copy.csv
student-data.csv
1 01,IIT2019186,abcd
2 02,IIT2019187,KUNA
3 03,IIT2019188,DEBA
4 04,IIT2019189,NIDH
5 05,IIT2019190,RISH
6 06,IIT2019191,PECH
7 07,IIT2019192,bhgd
8 08,IIT2019193,CHET
9 09,IIT2019194,RAHU
10 10,IIT2019195,bhjf
11 11,IIT2019196,PRIY
12 12,IIT2019197,JAID
13 13,IIT2019198,bhdg
14 14,IIT2019199,ABHI
15 15,IIT2019200,RAJJ
16 16,IIT2019201,BONT
17 17,IIT2019202,JYOT
18 18,IIT2019203,KRIS
19 19,IIT2019204,MITT
20 20,IIT2019205,SANS
21 21,IIT2019206,CHAU
22 22,IIT2019207,BALL
23 23,IIT2019208,DHAN
24 24,IIT2019210,ADIT
25 25,IIT2019211,DIVY
26 26,IIT2019212,AMAN
27 27,IIT2019213,VIKR
28 28,IIT2019214,AKSH
29 29,IIT2019215,RAVU
30 30,IIT2019216,bdhd
31 31,iit2019218,ANMO
aditya@aditya: ~/Desktop/os/Lab5/aditya/VideoDemo
Do you want to update the record (Y/N): n
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 7
Alloted Frame Number=2
Details of the Requseted Record:-
07,IIT2019192,PRIT
Do you want to update the record (Y/N): y
Enter the new name (4 characters):bhgd
Details of the Record after Updaed :-
07,IIT2019192,bhgd
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 101
r=1, m=1, v=1 r=1, m=0, v=1 r=1, m=1, v=1
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 10
Alloted Frame Number=1
Details of the Requseted Record:-
10,IIT2019195,DEEP
Do you want to update the record (Y/N): y
Enter the new name (4 characters):bhjf
Details of the Record after Updaed :-
10,IIT2019195,bhjf
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 13
Alloted Frame Number=0
0 0
Ln 12, Col 15 (4 selected)
Spaces: 4
UTF-8
CRLF
Plain Text


Activities
Terminal
Nov 5 13:17
student-data.csv - VideoDemo - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
VIDEODEMO
main
main.c
Makefile
myheader.h
mylibfuncns.c
mylibfuncns.o
student-data-copy.csv
student-data.csv
OUTLINE
main.c
Makefile
mylibfuncns.c
myheader.h
student-data.csv
student-data-copy.csv
student-data.csv
01,IIT2019186,abcd
02,IIT2019187,KUNA
03,IIT2019188,DEBA
04,IIT2019189,NIDH
05,IIT2019190,RISH
06,IIT2019191,PECH
07,IIT2019192,bhgd
08,IIT2019193,CHET
09,IIT2019194,RAHU
10,IIT2019195,bhjf
11,IIT2019196,PRIY
12,IIT2019197,JAID
13,IIT2019198,bhdg
14,IIT2019199,ABHI
15,IIT2019200,RAJJ
16,IIT2019201,BONT
17,IIT2019202,JYOT
18,IIT2019203,KRIS
19,IIT2019204,MITT
20,IIT2019205,SANS
21,IIT2019206,CHAU
22,IIT2019207,BALL
23,IIT2019208,DHAN
24,IIT2019210,ADIT
25,IIT2019211,DIVY
26,IIT2019212,AMAN
27,IIT2019213,VIKR
28,IIT2019214,AKSH
29,IIT2019215,RAVU
30,IIT2019216,bdhd
31,iit2019218,ANMO
aditya@aditya: ~/Desktop/os/Lab5/aditya/VideoDemo
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 13
Alloted Frame Number=0
Details of the Requseted Record:-
13,IIT2019198,NISC
Do you want to update the record (Y/N): y
Enter the new name (4 characters):bhdg
Details of the Record after Updaed :-
13,IIT2019198,bhdg
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 101
r=1, m=1, v=1 r=1, m=1, v=1 r=0, m=1, v=1
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 1
Alloted Frame Number=2
Details of the Requseted Record:-
01,IIT2019186,abcd
Do you want to update the record (Y/N): n
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 12
Details of the Requseted Record:-
12,IIT2019197,JAID
Do you want to update the record (Y/N): n
Press 101 to show ref, mod and val bits associated with page numbers in queue
Ln 12, Col 15 (4 selected)
Spaces: 4
UTF-8
CRLF
Plain Text


Activities
Terminal
Nov 5 13:17
student-data.csv - VideoDemo - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
VIDEODEMO
main
main.c
Makefile
myheader.h
mylibfuncns.c
mylibfuncns.o
student-data-copy.csv
student-data.csv
OUTLINE
main.c
Makefile
mylibfuncns.c
myheader.h
student-data.csv
student-data-copy.csv
student-data.csv
01,IIT2019186,abcd
02,IIT2019187,KUNA
03,IIT2019188,DEBA
04,IIT2019189,NIDH
05,IIT2019190,RISH
06,IIT2019191,PECH
07,IIT2019192,bhgd
08,IIT2019193,CHET
09,IIT2019194,RAHU
10,IIT2019195,bhjf
11,IIT2019196,PRIY
12,IIT2019197,JAID
13,IIT2019198,bhdg
14,IIT2019199,ABHI
15,IIT2019200,RAJJ
16,IIT2019201,BONT
17,IIT2019202,JYOT
18,IIT2019203,KRIS
19,IIT2019204,MITT
20,IIT2019205,SANS
21,IIT2019206,CHAU
22,IIT2019207,BALL
23,IIT2019208,DHAN
24,IIT2019210,ADIT
25,IIT2019211,DIVY
26,IIT2019212,AMAN
27,IIT2019213,VIKR
28,IIT2019214,AKSH
29,IIT2019215,RAVU
30,IIT2019216,bdhd
31,iit2019218,ANMO
aditya@aditya: ~/Desktop/os/Lab5/aditya/VideoDemo
Enter Record No: 12
Details of the Requseted Record:-
12,IIT2019197,JAID
Do you want to update the record (Y/N): n
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 101
r=1, m=1, v=1 r=1, m=1, v=1 r=1, m=0, v=1
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 30
Alloted Frame Number=2
Details of the Requseted Record:-
30,IIT2019216,VANS
Do you want to update the record (Y/N): y
Enter the new name (4 characters):bdhd
Details of the Record after Updaed :-
30,IIT2019216,bdhd
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: 101
r=0, m=1, v=1 r=0, m=1, v=1 r=1, m=1, v=1
Press 101 to show ref, mod and val bits associated with page numbers in queue
Enter Record No: -1
Invalid Record Number !!!
Total Page faults=7
aditya@aditya:~/Desktop/os/Lab5/aditya/VideoDemo$
Ln 12, Col 15 (4 selected)
Spaces: 4
UTF-8
CRLF
Plain Text

Basically for this video, we have added an extra command that since we have 94 records so, **if user gives 101 as record number then the ref, mod and val bits would be shown** for all the page numbers in queue.

Now let us start

make clean will remove all the executables

initially all bits are 0

Now all frames are occupied

for record 10, frame 1 is replaced

frame 0 is replaced

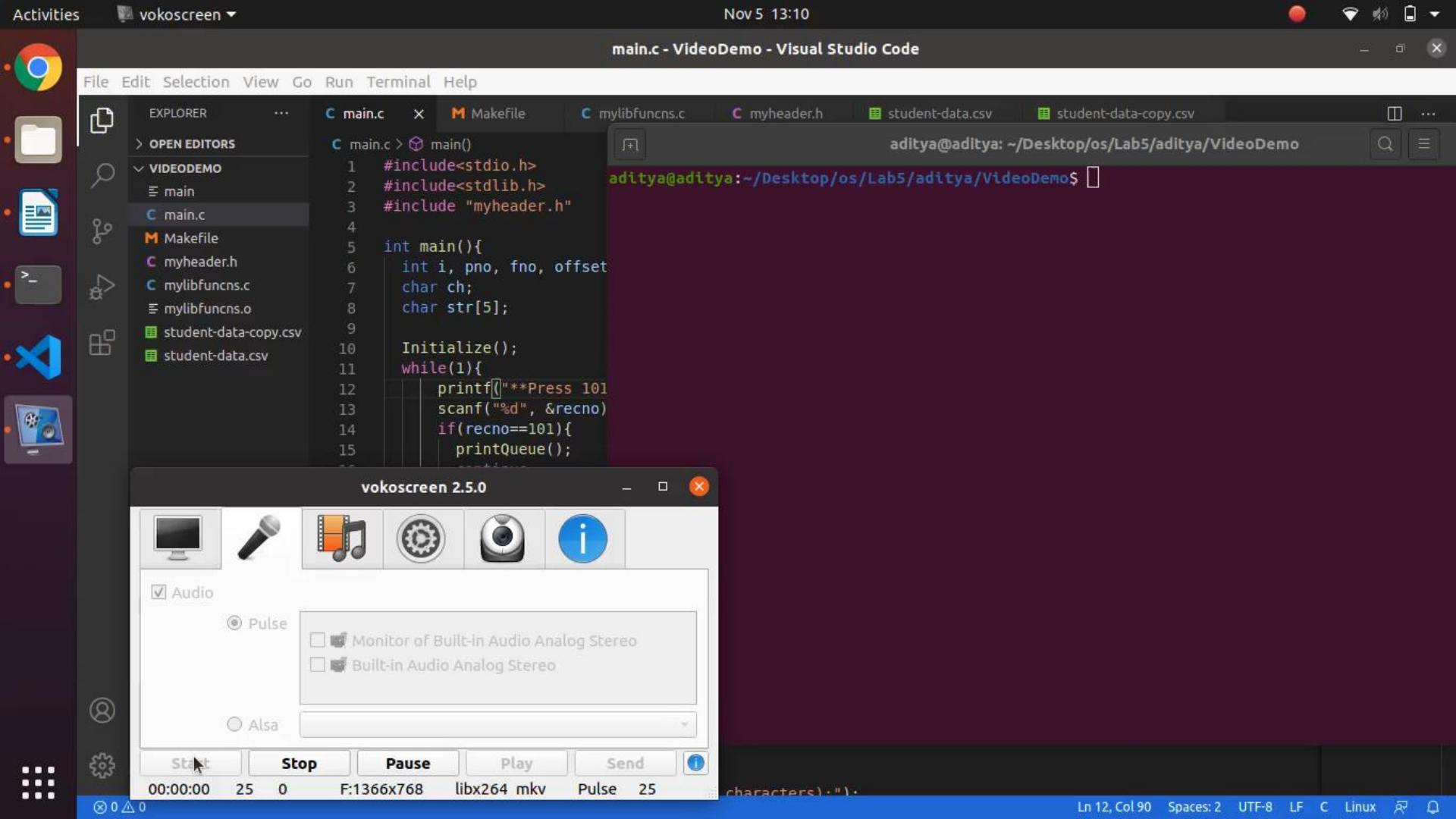
for same page no, frame is not replaced

we can exit by giving record -1

let's count them manually, it is coming 7 manually too

let us see csv file , all changes are same as given by user

Thanks


Activities
vokoscreen
Nov 5 13:10
main.c - VideoDemo - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
VIDEODEMO
main
main.c
Makefile
myheader.h
mylibfuncns.c
mylibfuncns.o
student-data-copy.csv
student-data.csv
main.c
Makefile
mylibfuncns.c
myheader.h
student-data.csv
student-data-copy.csv
main.c > main()
#include<stdio.h>
#include<stdlib.h>
#include "myheader.h"
int main(){
int i, pno, fno, offset
char ch;
char str[5];
Initialize();
while(1){
printf("**Press 101
scanf("%d", &recno)
if(recno==101){
printQueue();
aditya@aditya: ~/Desktop/os/Lab5/aditya/VideoDemo
aditya@aditya:~/Desktop/os/Lab5/aditya/VideoDemo$
vokoscreen 2.5.0
Audio
Pulse
Monitor of Built-in Audio Analog Stereo
Built-in Audio Analog Stereo
Alsa
Start
Stop
Pause
Play
Send
00:00:00 25 0 F:1366x768 libx264 mkv Pulse 25
characters):");
Ln 12, Col 90 Spaces: 2 UTF-8 LF C Linux

THANKS