**OS ASSIGNMENT 7B**

**Name:Jayesh Surywanshi**

**Roll no:88**

**TE IT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Client side program-

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 27

void die(char \*s)

{

perror(s);

exit(1);

}

int main()

{

int shmid,retval;

key\_t key;

char \*shm, \*s;

key = 5678;

if ((shmid = shmget(key, MAXSIZE,0666)) < 0)

die("shmget");

if ((shm = shmat(shmid, NULL, 0)) == (char \*) -1)

die("shmat");

//Now read what the server put in the memory.

for (s = shm; \*s != '\0'; s++)

putchar(\*s);

putchar('\n');

/\*

\*Change the first character of the

\*segment to '\*', indicating we have read

\*the segment.

\*/

\*shm = '\*';

retval=shmdt(shm);

if(retval<0)

{

printf("detachment failure");

return 0;

}

retval=shmctl(shmid,IPC\_RMID,NULL);

exit(0);

}

Server side program-

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 27

void die(char \*s)

{

perror(s);

exit(1);

}

int main()

{

char c;

int shmid,retval;

key\_t key;

char \*shm, \*s;

key = 5678;

if ((shmid = shmget(key, MAXSIZE, IPC\_CREAT | 0666)) < 0)

die("shmget");

if ((shm = shmat(shmid,NULL,0)) == (char \*) -1)

die("shmat");

/\*

\* \* Put some things into the memory for the

\* other process to read.

\* \*/

s = shm;

for (c = 'a'; c <= 'z'; c++)

\*s++ = c;

/\*

\* Wait until the other process

\* changes the first character of our memory

\* to '\*', indicating that it has read what

\* we put there.

\*/

while (\*shm != '\*')

sleep(1);

retval=shmdt(shm);

if(retval<0)

{

printf("detachment failure");

return 0;

}

exit(0);

}

Output=

Page Replacement Algorithms

1.Enter data

2.FIFO

3.Optimal

4.LRU

5.LFU

6.Second Chance

7.Exit

Enter your choice:1

Enter length of page reference sequence:8

Enter the page reference sequence:2

3

4

2

3

5

6

2

Enter no of frames:3

Page Replacement Algorithms

1.Enter data

2.FIFO

3.Optimal

4.LRU

5.LFU

6.Second Chance

7.Exit

Enter your choice:2

For 2 : 2

For 3 : 2 3

For 4 : 2 3 4

For 2 :No page fault

For 3 :No page fault

For 5 : 3 4 5

For 6 : 4 5 6

For 2 : 5 6 2

Total no of page faults:6

Page Replacement Algorithms

1.Enter data

2.FIFO

3.Optimal

4.LRU

5.LFU

6.Second Chance

7.Exit

Enter your choice:3

For 2 : 2

For 3 : 2 3

For 4 : 2 3 4

For 2 :No page fault

For 3 :No page fault

For 5 : 2 5 4

For 6 : 2 6 4

For 2 :No page fault

Total no of page faults:5

Page Replacement Algorithms

1.Enter data

2.FIFO

3.Optimal

4.LRU

5.LFU

6.Second Chance

7.Exit

Enter your choice:4

For 2 : 2

For 3 : 2 3

For 4 : 2 3 4

For 2 :No page fault!

For 3 :No page fault!

For 5 : 2 3 5

For 6 : 6 3 5

For 2 : 6 2 5

Total no of page faults:6

Page Replacement Algorithms

1.Enter data

2.FIFO

3.Optimal

4.LRU

5.LFU

6.Second Chance

7.Exit

Enter your choice:5

For 2 : 2

For 3 : 2 3

For 4 : 2 3 4

For 2 :No page fault!

For 3 :No page fault!

For 5 : 2 3 5

For 6 : 2 3 6

For 2 :No page fault!

Total no of page faults:5

Page Replacement Algorithms

1.Enter data

2.FIFO

3.Optimal

4.LRU

5.LFU

6.Second Chance

7.Exit

Enter your choice:7