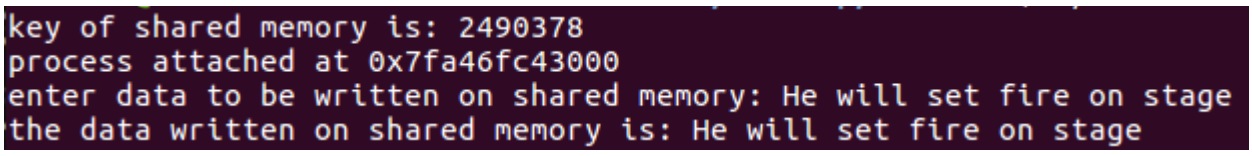
//program for writing in shared memory

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#include<string.h>
int main()
{
 int i;
 void *sm;
 char buff[100];
 int shmid;
 shmid=shmget((key_t)2345,1024,0666|IPC_CREAT);
 printf("key of shared memory is: %d\n",shmid);
 sm=shmat(shmid,NULL,0);
 printf("process attached at %p\n",sm);
 printf("enter data to be written on shared memory: ");
 scanf("%[^\n]s",buff);
 strcpy(sm,buff);
 printf("the data written on shared memory is: %s\n",(char*)sm);
 return 0;
}
```

output:



//Progarm for reading from shared memory

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#include<string.h>
int main()
{
 int i;
 void* sm;
 char buff[100];
 int shmid;
 shmid=shmget((key_t)2345,1024,0666);
 printf("key of shared memory is: %d",shmid);
 sm=shmat(shmid,NULL,0);
 printf("process attached at %p\n",sm);
 printf("data read from the shared memory is: %s\n",(char*)sm);
 return 0;
```

}

output:

```
key of shared memory is: 2490378process attached at 0x7f7f5e4b8000
data read from the shared memory is: He will set fire on stage
```

//Program for sending message to kernel

```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MAX 10

struct mesg_buffer {
    long mesg_type;
    char mesg_text[100];
} message;

int main()
{
    key_t key;
    int msgid;
    key = ftok("file", 64);
    msgid = msgget(key, 0666 | IPC_CREAT);
    message.mesg_type = 1;
    printf("Write Data : ");
    fgets(message.mesg_text,MAX,stdin);
    msgsnd(msgid, &message, sizeof(message), 0);
    printf("Data sent is : %s \n", message.mesg_text);

    return 0;
}
```

output:

```
Write Data : ram is playing.
Data send is : ram is playing.
```

//Program for displaying the message received by kernel

```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>

struct mesg_buffer {
    long mesg_type;
    char mesg_text[100];
} message;

int main()
```

```
{
    key_t key;
    int msgid;
    key = ftok("file", 64);
    msgid = msgget(key, 0666 | IPC_CREAT);
    msgrcv(msgid, &message, sizeof(message), 1, 0);
    printf("Data Received is : %s \n",
                    message.mesg_text);
    msgctl(msgid, IPC_RMID, NULL);

    return 0;
}
```

output: