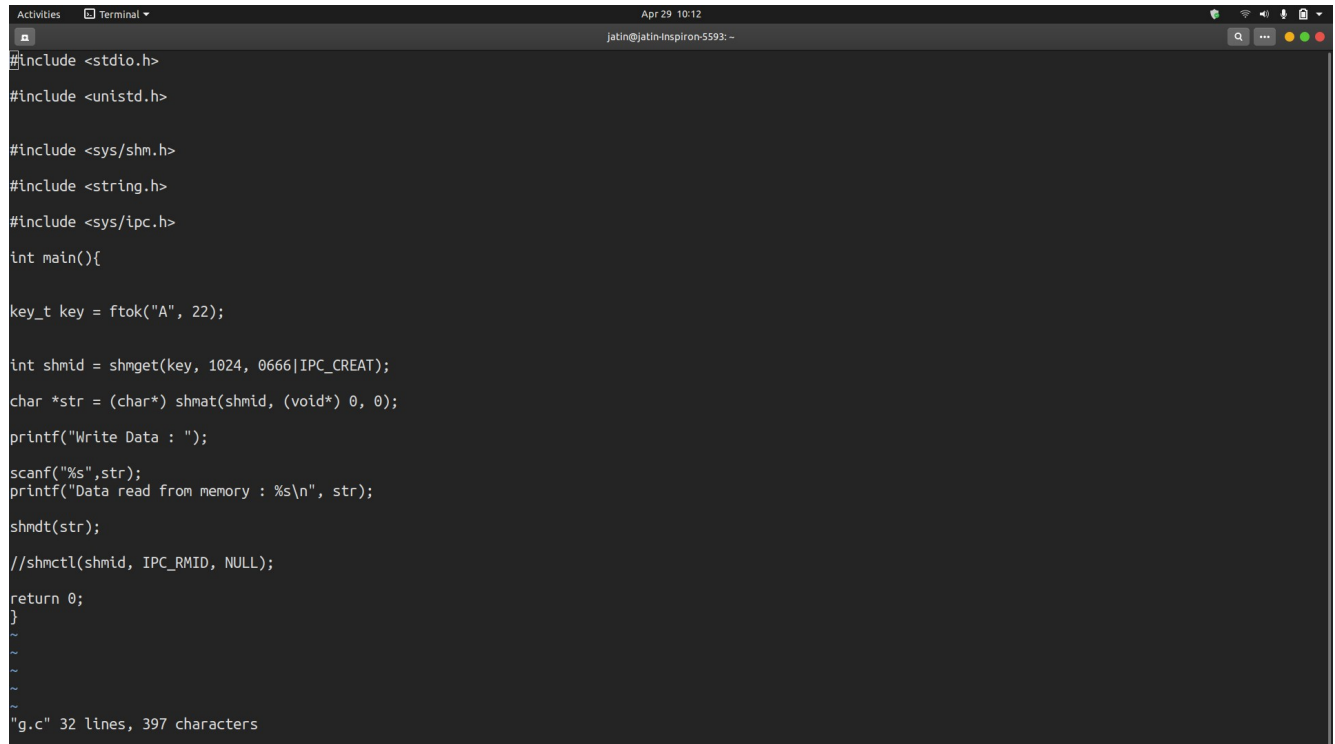


RA1911003010714

Jatin S Rastogi

OS EXP 12,13

1)Writer's code



```
#include <stdio.h>
#include <unistd.h>

#include <sys/shm.h>
#include <string.h>
#include <sys/ipc.h>

int main(){

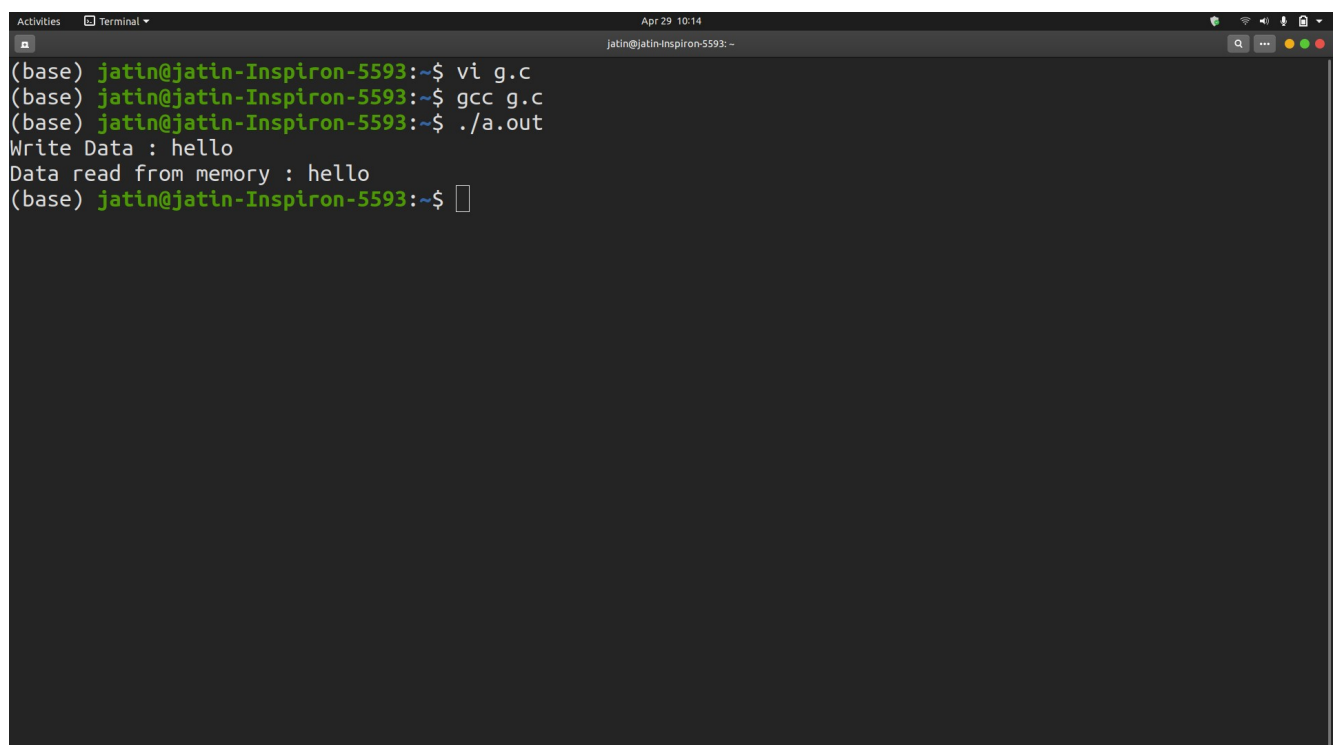
    key_t key = ftok("A", 22);

    int shmid = shmget(key, 1024, 0666|IPC_CREAT);
    char *str = (char*) shmat(shmid, (void*) 0, 0);
    printf("Write Data : ");
    scanf("%s",str);
    printf("Data read from memory : %s\n", str);

    shmdt(str);

    //shmctl(shmid, IPC_RMID, NULL);

    return 0;
}
~
~
~
~
~
"g.c" 32 lines, 397 characters
```



```
(base) jatin@jatin-Inspiron-5593:~$ vi g.c
(base) jatin@jatin-Inspiron-5593:~$ gcc g.c
(base) jatin@jatin-Inspiron-5593:~$ ./a.out
Write Data : hello
Data read from memory : hello
(base) jatin@jatin-Inspiron-5593:~$
```

Reader's code



A screenshot of a terminal window with a dark background. The window title bar shows 'Activities', 'Terminal', and the date 'Apr 29 10:17'. The terminal content displays C code for shared memory communication. The code includes headers for stdio, unistd, sys/shm, string, and sys/ipc. It defines a main function that sets a key, creates a shared memory segment, attaches it, and prints the data. The code is 32 lines long and 399 characters.

```
#include <stdio.h>

#include <unistd.h>

#include <sys/shm.h>
#include <string.h>
#include <sys/ipc.h>

int main(){

key_t key = ftok("A", 22);

int shmid = shmget(key, 1024, 0666|IPC_CREAT);
char *str = (char*) shmat(shmid, (void*) 0, 0);

//printf("Write Data : ");

//scanf("%s",str);
printf("Data read from memory : %s\n", str);

"g.c" 32 lines, 399 characters
```

Output:

```
Activities Terminal Apr 29 10:16
jatin@jatin-Inspiron-5593: ~
(base) jatin@jatin-Inspiron-5593:~$ vi g.c
(base) jatin@jatin-Inspiron-5593:~$ gcc g.c
(base) jatin@jatin-Inspiron-5593:~$ ./a.out
Data read from memory : hello
(base) jatin@jatin-Inspiron-5593:~$
```

```
Activities Terminal Apr 29 10:25
jatin@jatin-Inspiron-5593: ~
#include <stdio.h>

#include <sys/ipc.h>

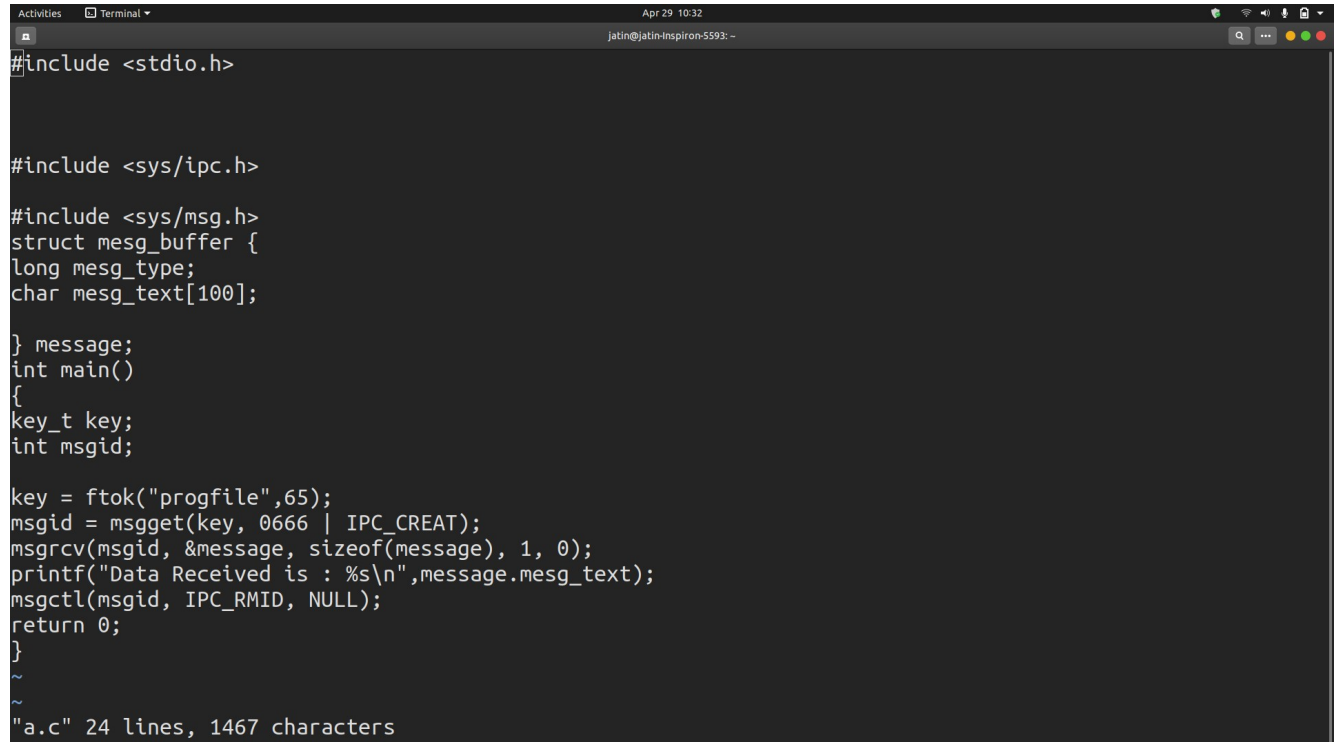
#include <sys/msg.h>
struct mesg_buffer {
long mesg_type;
char mesg_text[100];
} message;
int main()
{
key_t key;
int msgid;

key = ftok("progfile",65);
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:

```
shivi21@LAPTOP-EAV0Q77J: ~
shivi21@LAPTOP-EAV0Q77J:~$ vi a.c
shivi21@LAPTOP-EAV0Q77J:~$ gcc a.c
shivi21@LAPTOP-EAV0Q77J:~$ ./a.out
Write Data : hey
Data send is : hey
shivi21@LAPTOP-EAV0Q77J:~$
```

Reader's code



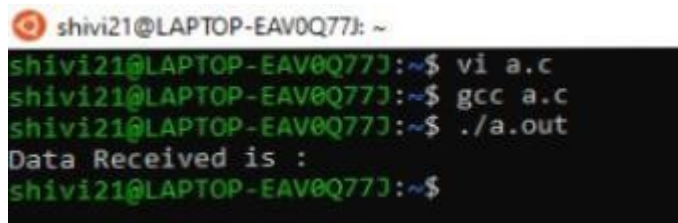
```
#include <stdio.h>

#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("progfile",65);
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;
}
~
~
"a.c" 24 lines, 1467 characters
```

Output:



```
shivi21@LAPTOP-EAV0Q77J: ~
shivi21@LAPTOP-EAV0Q77J:~$ vi a.c
shivi21@LAPTOP-EAV0Q77J:~$ gcc a.c
shivi21@LAPTOP-EAV0Q77J:~$ ./a.out
Data Received is :
shivi21@LAPTOP-EAV0Q77J:~$
```