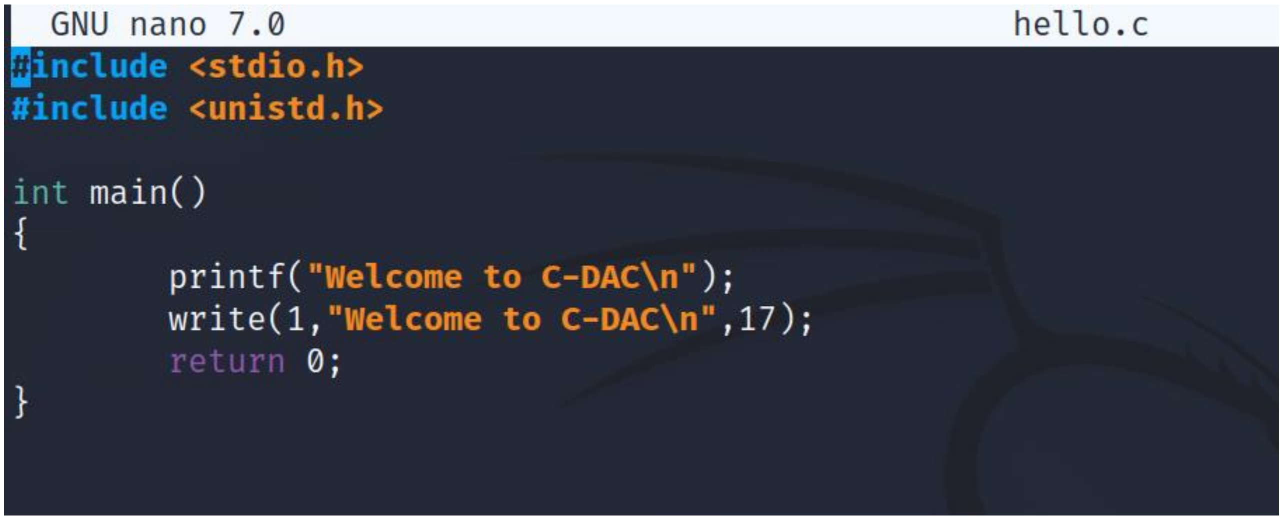
Sudarshan Ukharde

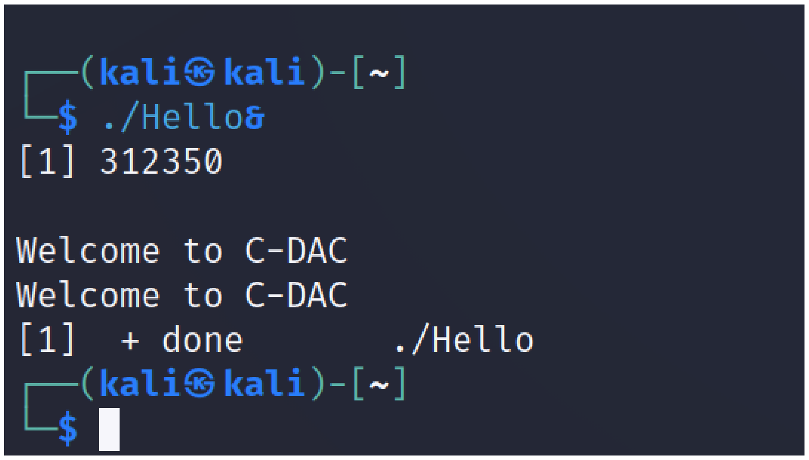
16-12-2022 Lab-Qns

1, Write a comment and display the process executed in background.



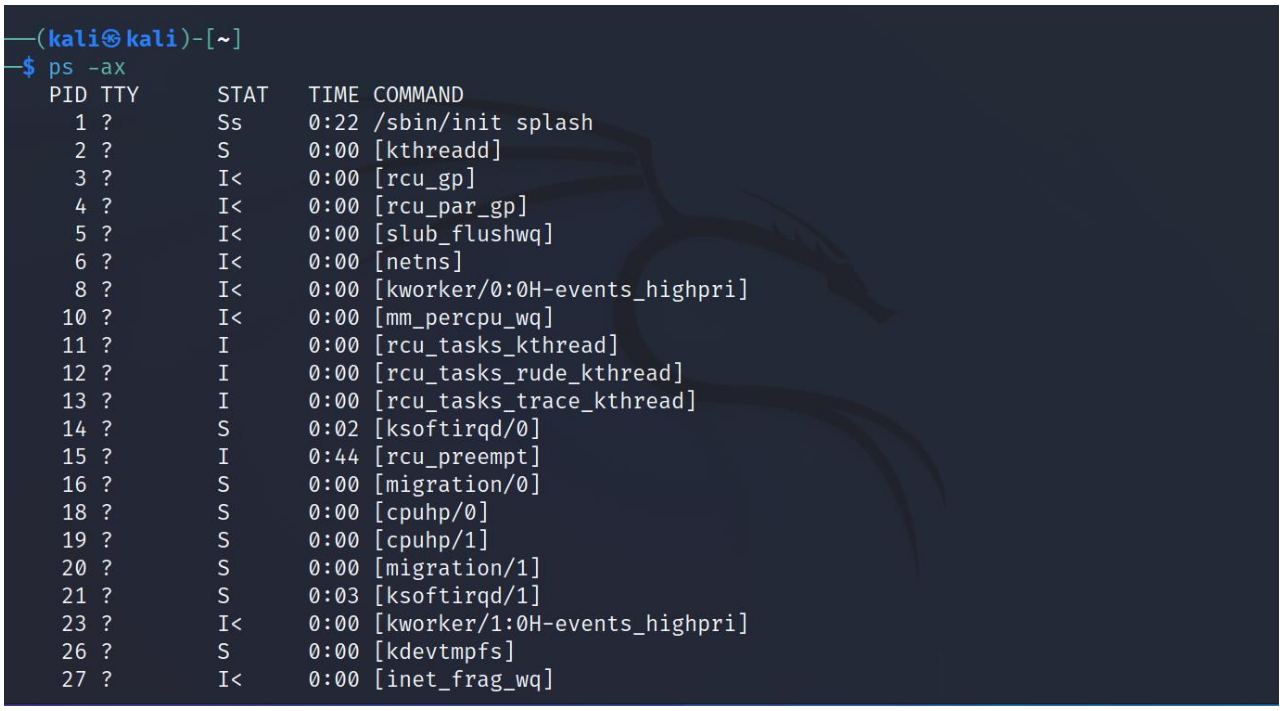
```
  GNU nano 7.0                                          hello.c
#include <stdio.h>
#include <unistd.h>

int main()
{
        printf("Welcome to C-DAC\n");
        write(1,"Welcome to C-DAC\n",17);
        return 0;
}
```

Output:



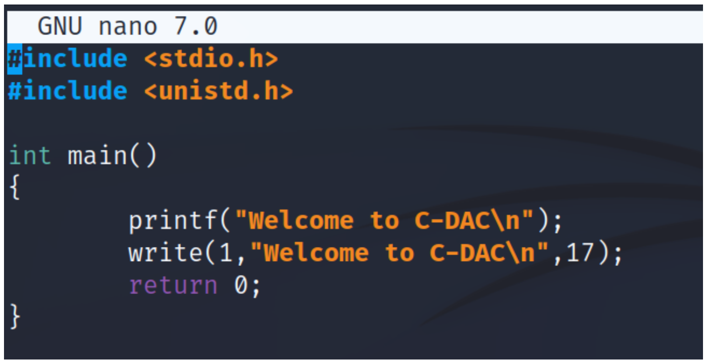
```
┌──(kali㉿kali)-[~]
└─$ ./Hello&
[1] 312350

Welcome to C-DAC
Welcome to C-DAC
[1]  + done       ./Hello
┌──(kali㉿kali)-[~]
└─$ 
```

2, Write a program to display list of all process listed



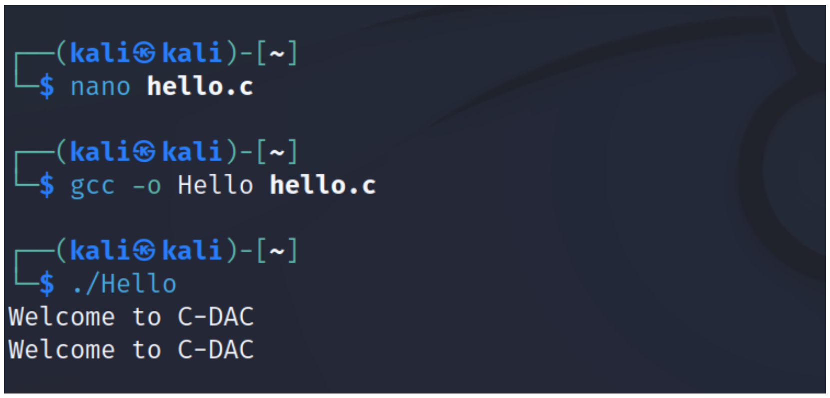
```
┌──(kali㉿kali)-[~]
└─$ ps -ax
    PID TTY      STAT   TIME COMMAND
      1 ?        Ss     0:22 /sbin/init splash
      2 ?        S      0:00 [kthreadd]
      3 ?        I<     0:00 [rcu_gp]
      4 ?        I<     0:00 [rcu_par_gp]
      5 ?        I<     0:00 [slub_flushwq]
      6 ?        I<     0:00 [netns]
      8 ?        I<     0:00 [kworker/0:0H-events_highpri]
     10 ?        I<     0:00 [mm_percpu_wq]
     11 ?        I      0:00 [rcu_tasks_kthread]
     12 ?        I      0:00 [rcu_tasks_rude_kthread]
     13 ?        I      0:00 [rcu_tasks_trace_kthread]
     14 ?        S      0:02 [ksoftirqd/0]
     15 ?        I      0:44 [rcu_preempt]
     16 ?        S      0:00 [migration/0]
     18 ?        S      0:00 [cpuhp/0]
     19 ?        S      0:00 [cpuhp/1]
     20 ?        S      0:00 [migration/1]
     21 ?        S      0:03 [ksoftirqd/1]
     23 ?        I<     0:00 [kworker/1:0H-events_highpri]
     26 ?        S      0:00 [kdevtmpfs]
     27 ?        I<     0:00 [inet_frag_wq]
```

3 ,Write a c program to print "Welcome To CDAC "using Printf and write function (using strace)



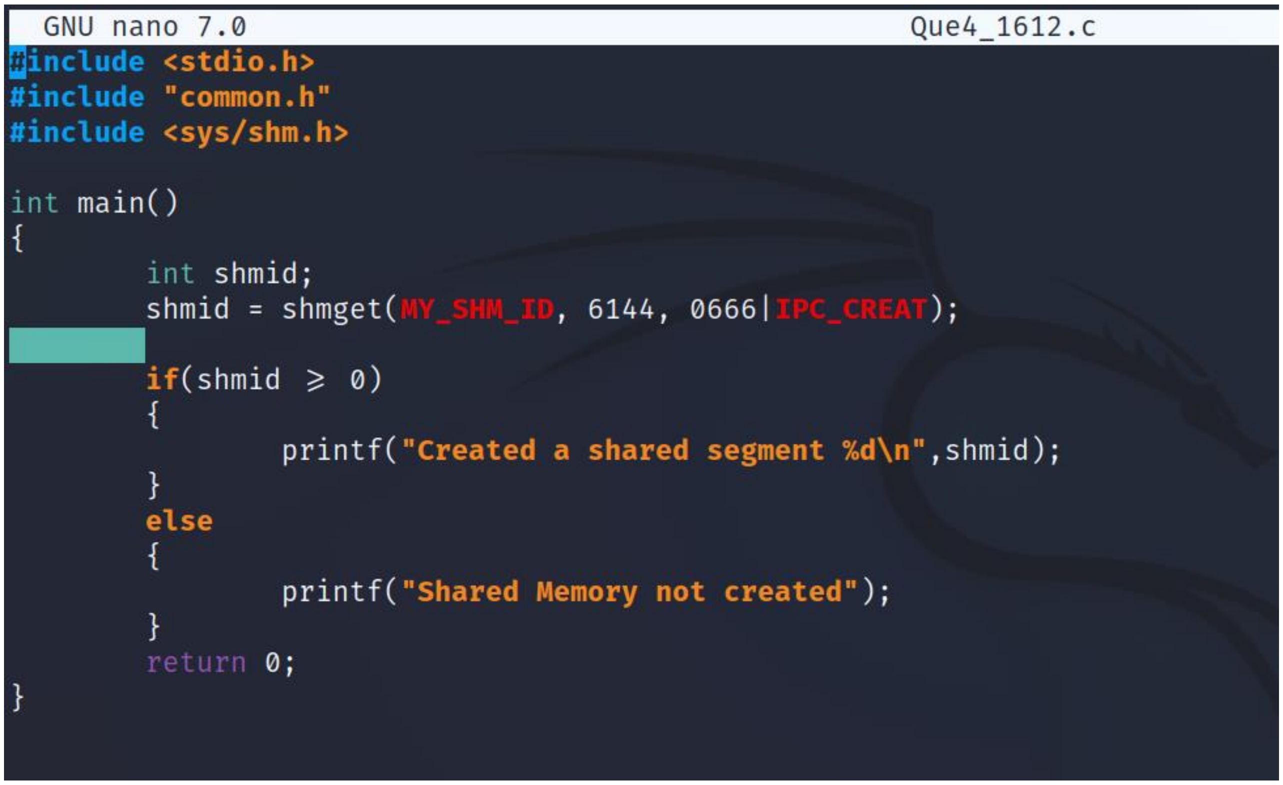
```
  GNU nano 7.0
#include <stdio.h>
#include <unistd.h>

int main()
{
        printf("Welcome to C-DAC\n");
        write(1,"Welcome to C-DAC\n",17);
        return 0;
}
```

OutPut:



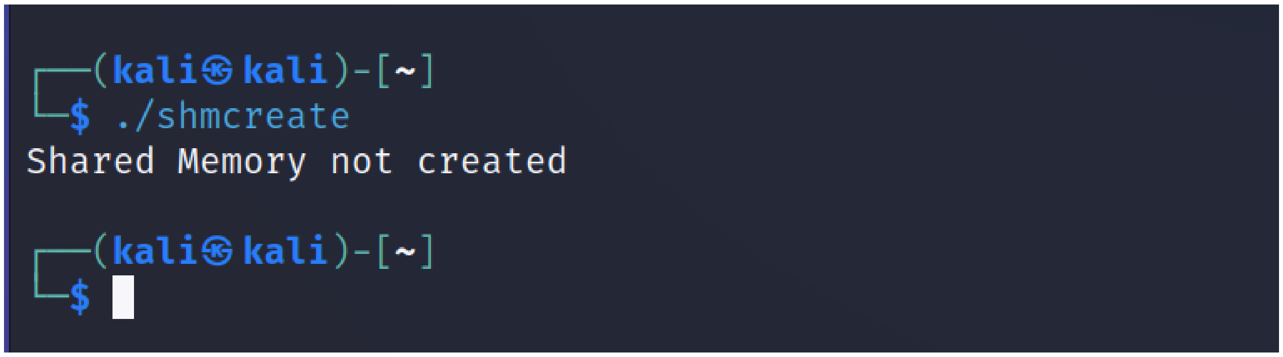
```
┌──(kali㉿kali)-[~]
└─$ nano hello.c

┌──(kali㉿kali)-[~]
└─$ gcc -o Hello hello.c

┌──(kali㉿kali)-[~]
└─$ ./Hello
Welcome to C-DAC
Welcome to C-DAC
```

4 Write a C program to create a shared memory with a size 6kb it should everyone can read and write And do the following operations



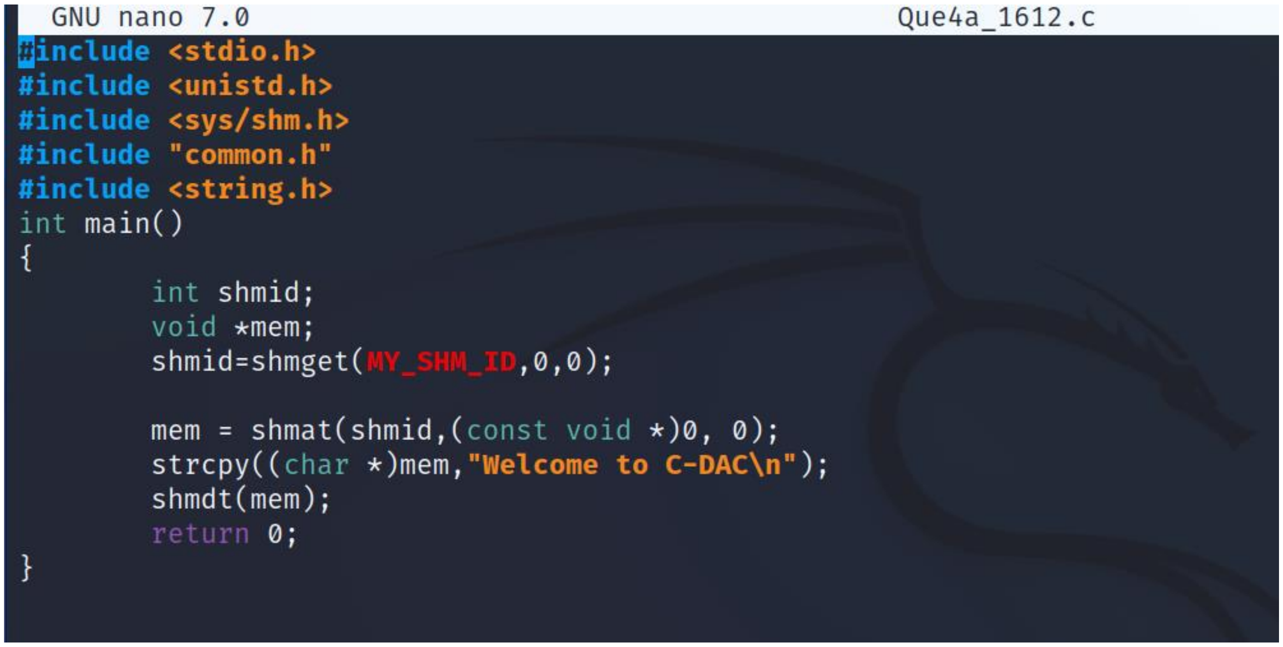
```
  GNU nano 7.0                                              Que4_1612.c
#include <stdio.h>
#include "common.h"
#include <sys/shm.h>

int main()
{
        int shmid;
        shmid = shmget(MY_SHM_ID, 6144, 0666|IPC_CREAT);

        if(shmid >= 0)
        {
                printf("Created a shared segment %d\n",shmid);
        }
        else
        {
                printf("Shared Memory not created");
        }
        return 0;
}
```

OutPut:



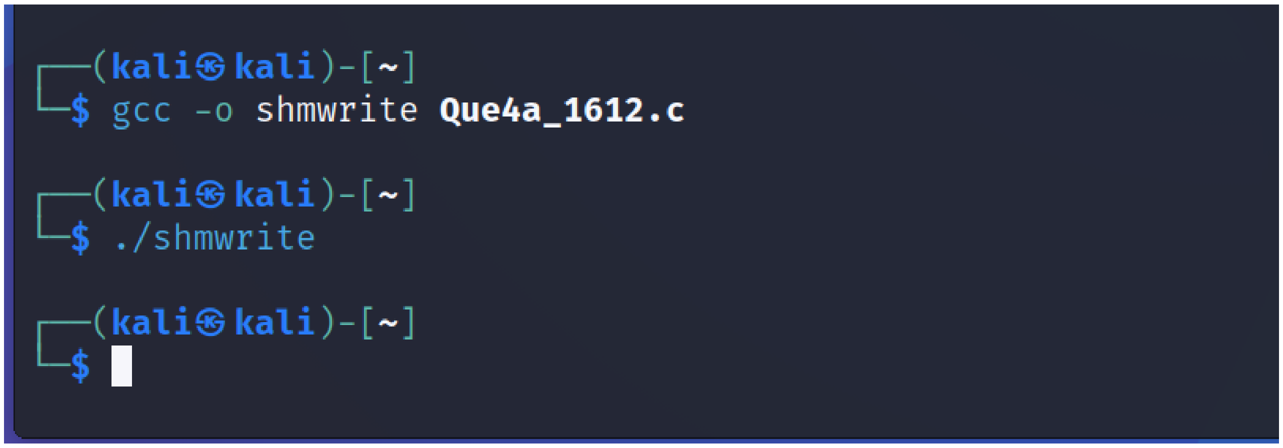
```
┌──(kali㉿kali)-[~]
└─$ ./shmcreate
Shared Memory not created

┌──(kali㉿kali)-[~]
└─$ 
```

4.1 Attaching a Shared Memory Segment



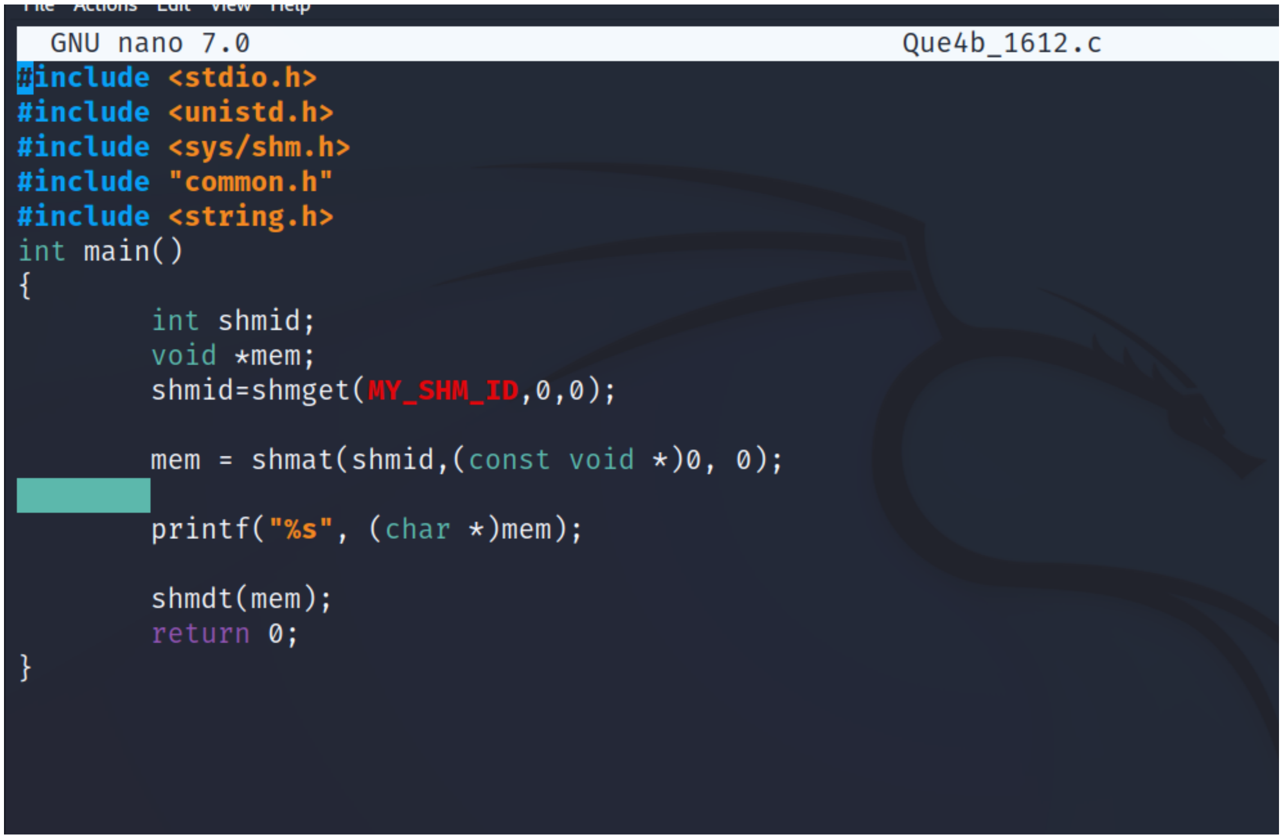
```
 GNU nano 7.0                                                Que4a_1612.c
#include <stdio.h>
#include <unistd.h>
#include <sys/shm.h>
#include "common.h"
#include <string.h>
int main()
{
        int shmid;
        void *mem;
        shmid=shmget(MY_SHM_ID,0,0);

        mem = shmat(shmid,(const void *)0, 0);
        strcpy((char *)mem,"Welcome to C-DAC\n");
        shmdt(mem);
        return 0;
}
```

OutPut:



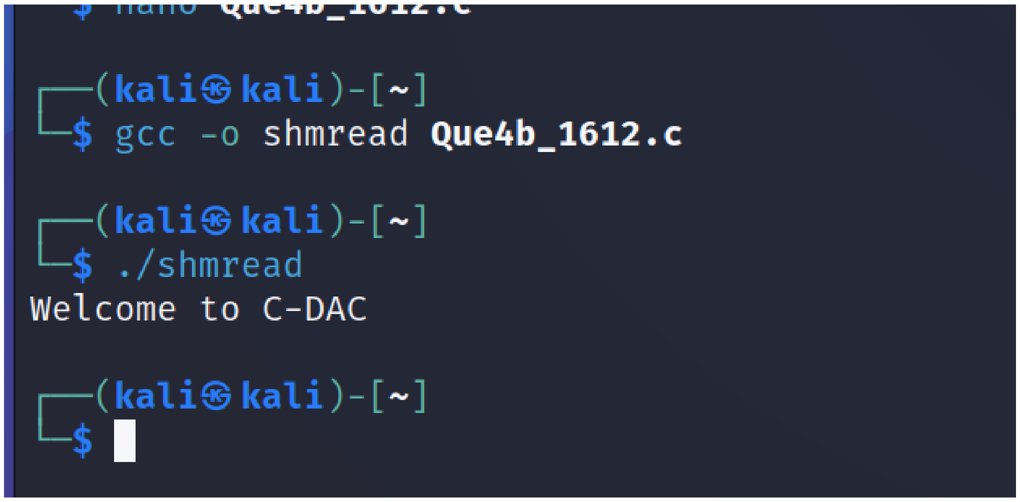
```
┌──(kali㉿kali)-[~]
└─$ gcc -o shmwrite Que4a_1612.c

┌──(kali㉿kali)-[~]
└─$ ./shmwrite

┌──(kali㉿kali)-[~]
└─$ 
```

4.2 Detaching the Shared Memory Segment



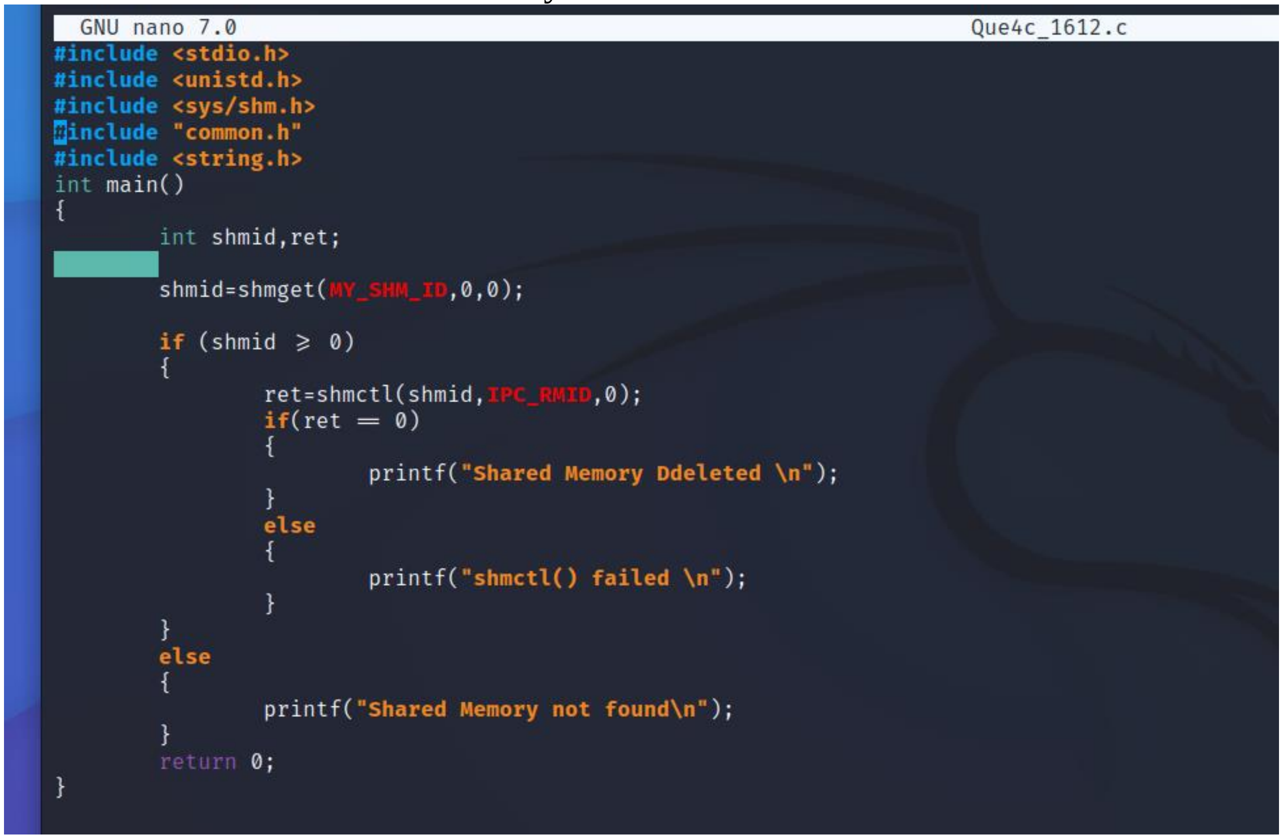
```
  GNU nano 7.0                                        Que4b_1612.c
#include <stdio.h>
#include <unistd.h>
#include <sys/shm.h>
#include "common.h"
#include <string.h>
int main()
{
        int shmid;
        void *mem;
        shmid=shmget(MY_SHM_ID,0,0);

        mem = shmat(shmid,(const void *)0, 0);

        printf("%s", (char *)mem);

        shmdt(mem);
        return 0;
}
```

Output:



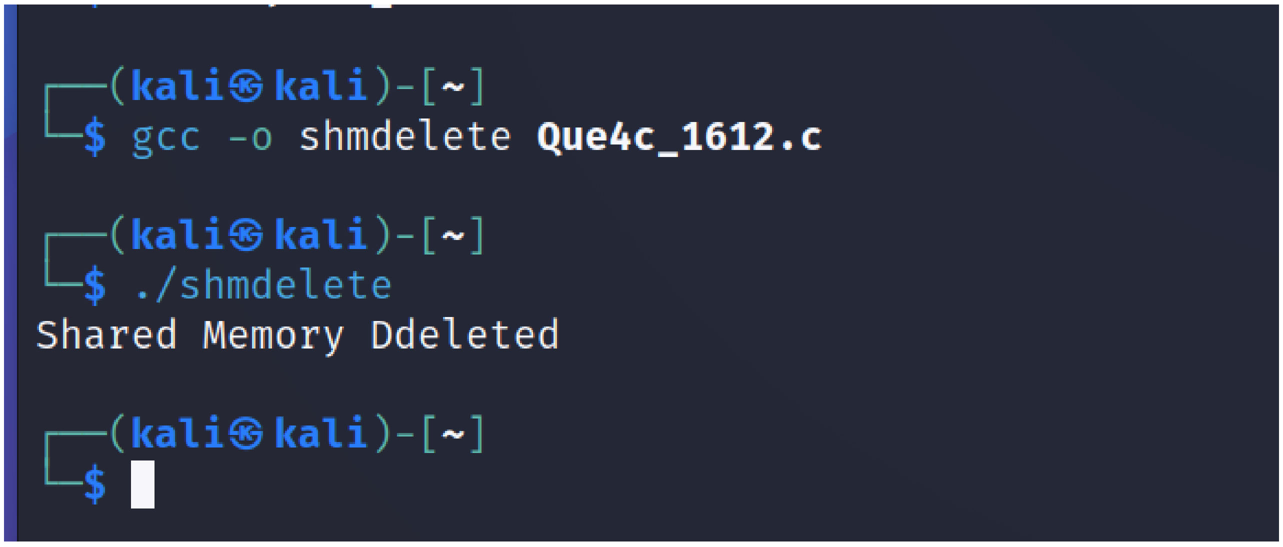
```
┌──(kali㉿kali)-[~]
└─$ gcc -o shmread Que4b_1612.c

┌──(kali㉿kali)-[~]
└─$ ./shmread
Welcome to C-DAC

┌──(kali㉿kali)-[~]
└─$ 
```

4.3 Delete the shared memory



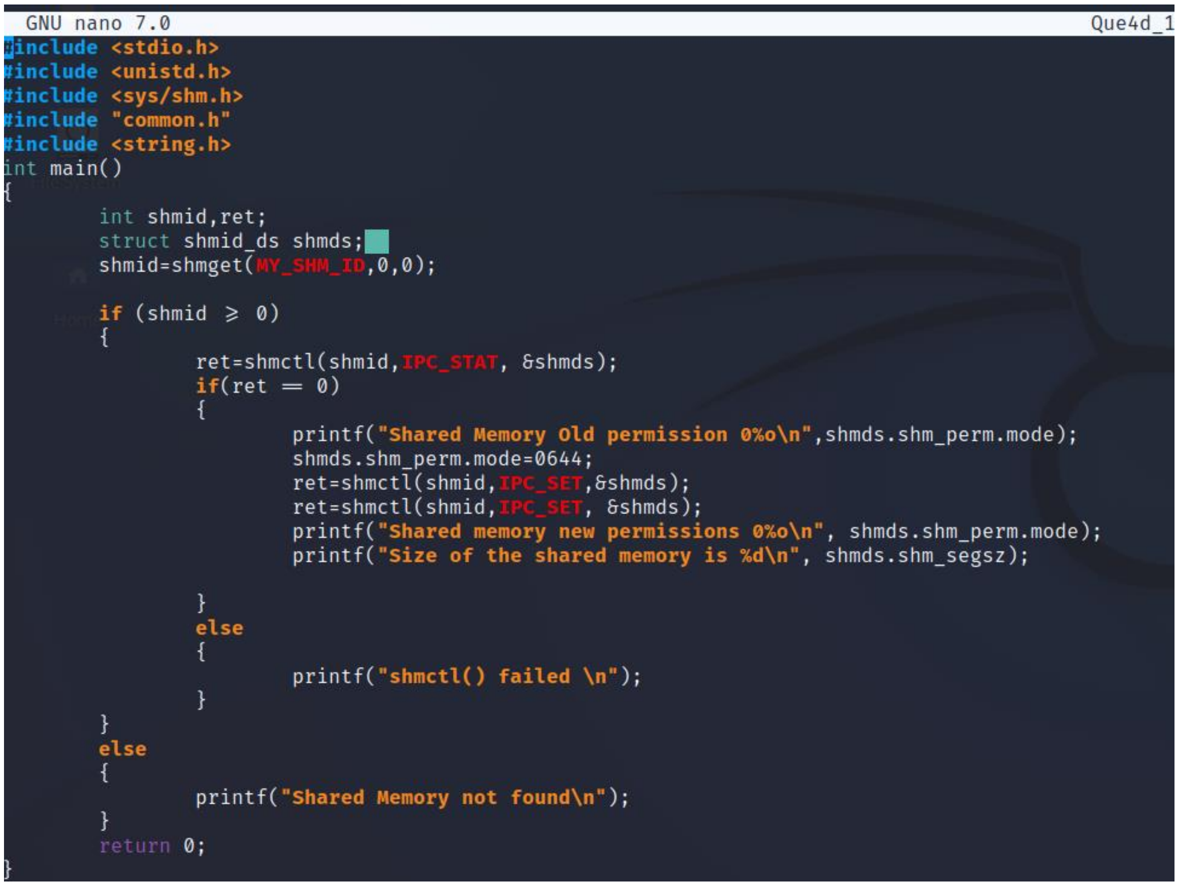
```
  GNU nano 7.0                                                           Que4c_1612.c
#include <stdio.h>
#include <unistd.h>
#include <sys/shm.h>
#include "common.h"
#include <string.h>
int main()
{
        int shmid,ret;

        shmid=shmget(MY_SHM_ID,0,0);

        if (shmid >= 0)
        {
                ret=shmctl(shmid,IPC_RMID,0);
                if(ret == 0)
                {
                        printf("Shared Memory Ddeleted \n");
                }
                else
                {
                        printf("shmctl() failed \n");
                }
        }
        else
        {
                printf("Shared Memory not found\n");
        }
        return 0;
}
```

OutPut:



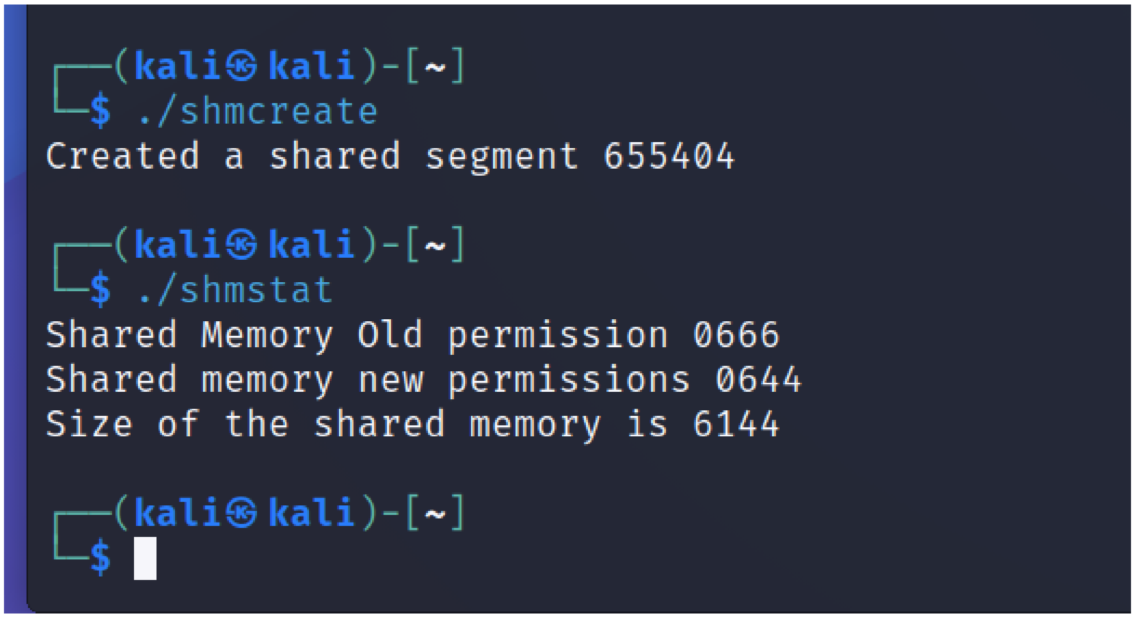
```
┌──(kali㉿kali)-[~]
└─$ gcc -o shmdelete Que4c_1612.c

┌──(kali㉿kali)-[~]
└─$ ./shmdelete
Shared Memory Ddeleted

┌──(kali㉿kali)-[~]
└─$ 
```

4.4 Modify the Existing Share memory 0666 to 0644



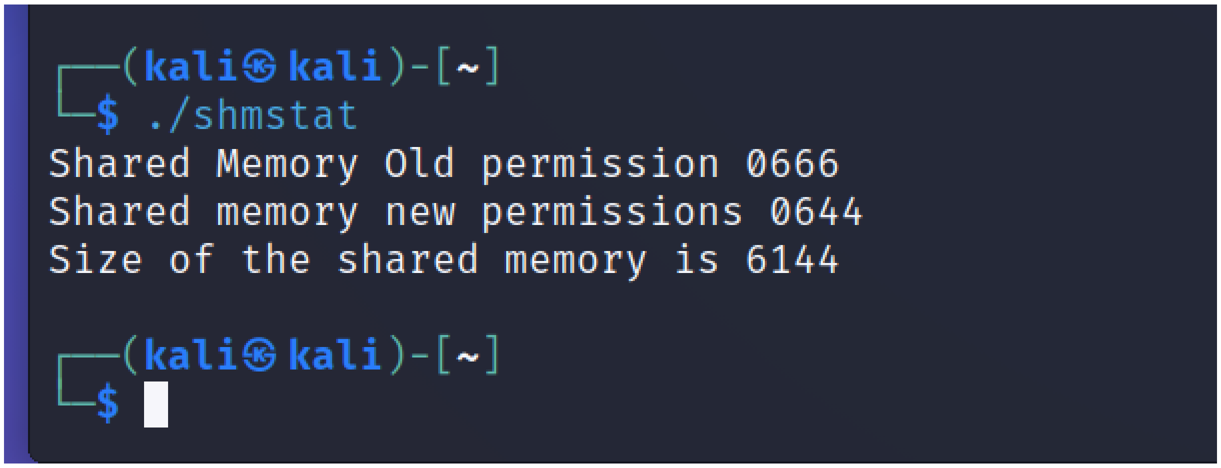
```
  GNU nano 7.0                                                              Que4d_1
#include <stdio.h>
#include <unistd.h>
#include <sys/shm.h>
#include "common.h"
#include <string.h>
int main()
{
        int shmid,ret;
        struct shmid_ds shmds;
        shmid=shmget(MY_SHM_ID,0,0);

        if (shmid >= 0)
        {
                ret=shmctl(shmid,IPC_STAT, &shmds);
                if(ret == 0)
                {
                        printf("Shared Memory Old permission 0%o\n",shmds.shm_perm.mode);
                        shmds.shm_perm.mode=0644;
                        ret=shmctl(shmid,IPC_SET,&shmds);
                        ret=shmctl(shmid,IPC_SET, &shmds);
                        printf("Shared memory new permissions 0%o\n", shmds.shm_perm.mode);
                        printf("Size of the shared memory is %d\n", shmds.shm_segsz);

                }
                else
                {
                        printf("shmctl() failed \n");
                }
        }
        else
        {
                printf("Shared Memory not found\n");
        }
        return 0;
}
```

OutPut:



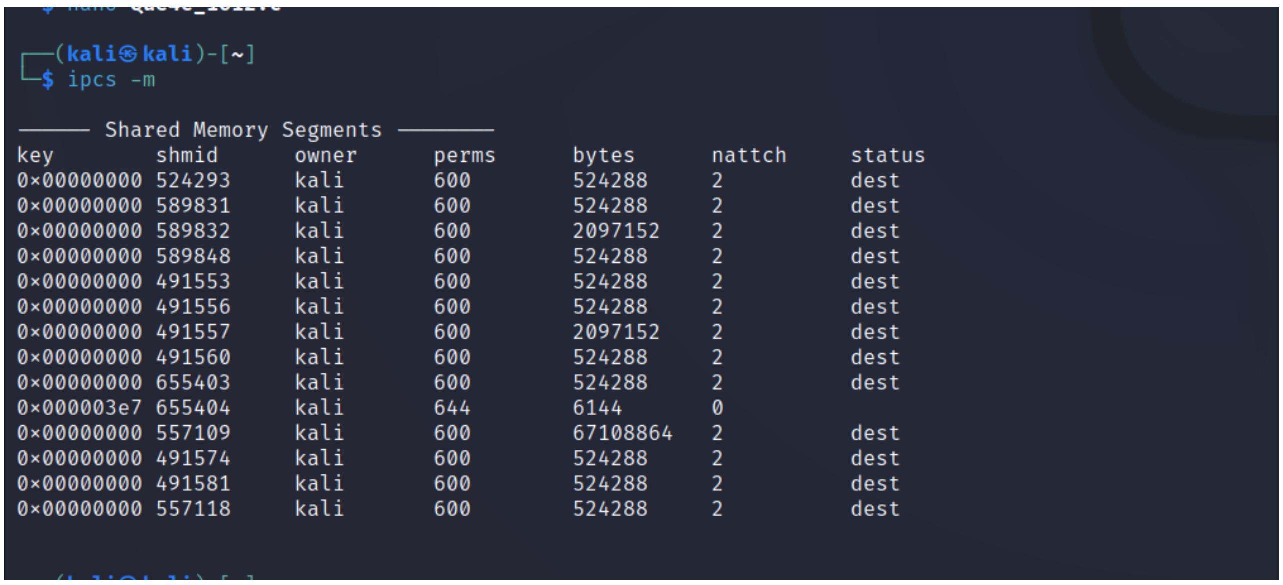
```
┌──(kali㉿kali)-[~]
└─$ ./shmcreate
Created a shared segment 655404

┌──(kali㉿kali)-[~]
└─$ ./shmstat
Shared Memory Old permission 0666
Shared memory new permissions 0644
Size of the shared memory is 6144

┌──(kali㉿kali)-[~]
└─$ 
```

4.5 Print he old and new permission values



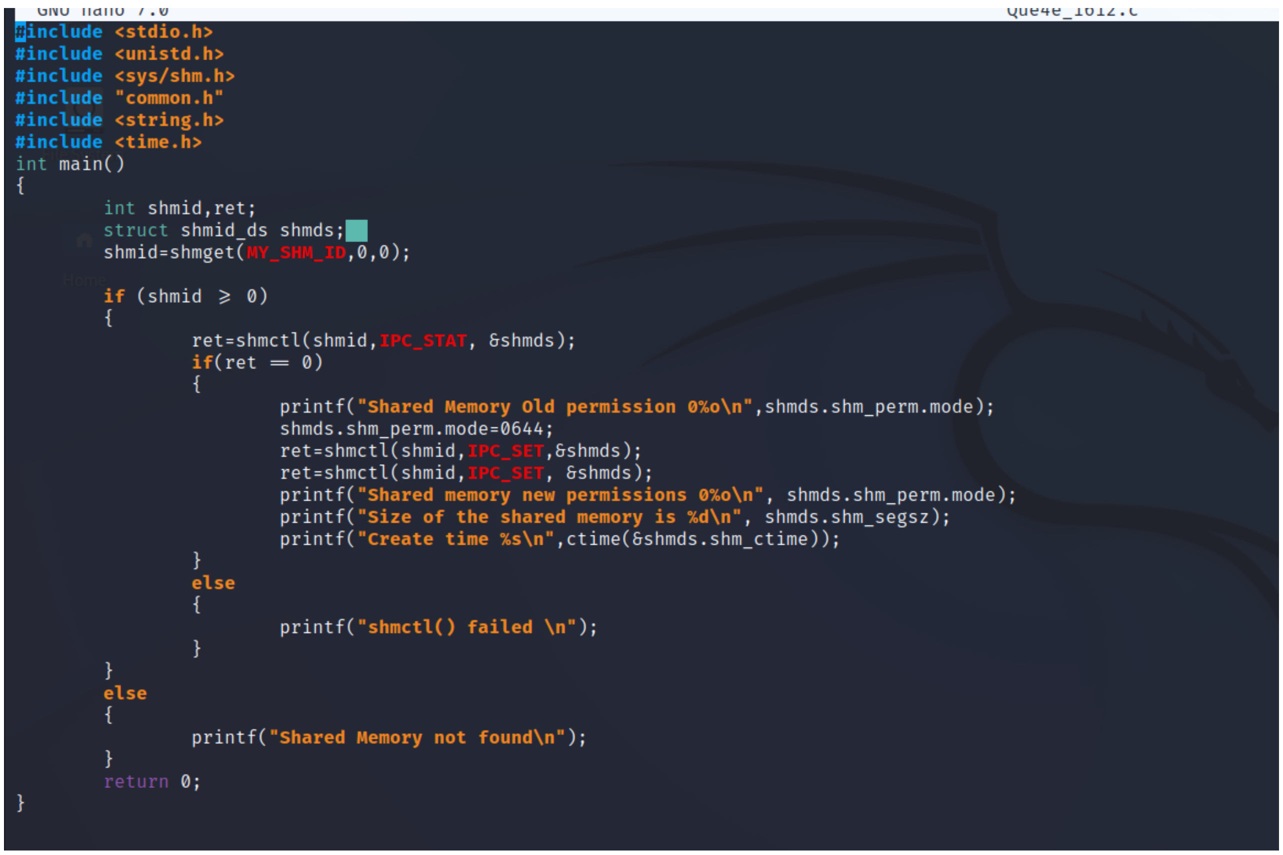
```
┌──(kali㉿kali)-[~]
└─$ ./shmstat
Shared Memory Old permission 0666
Shared memory new permissions 0644
Size of the shared memory is 6144

┌──(kali㉿kali)-[~]
└─$ 
```

4.6 Print the size of shared memory



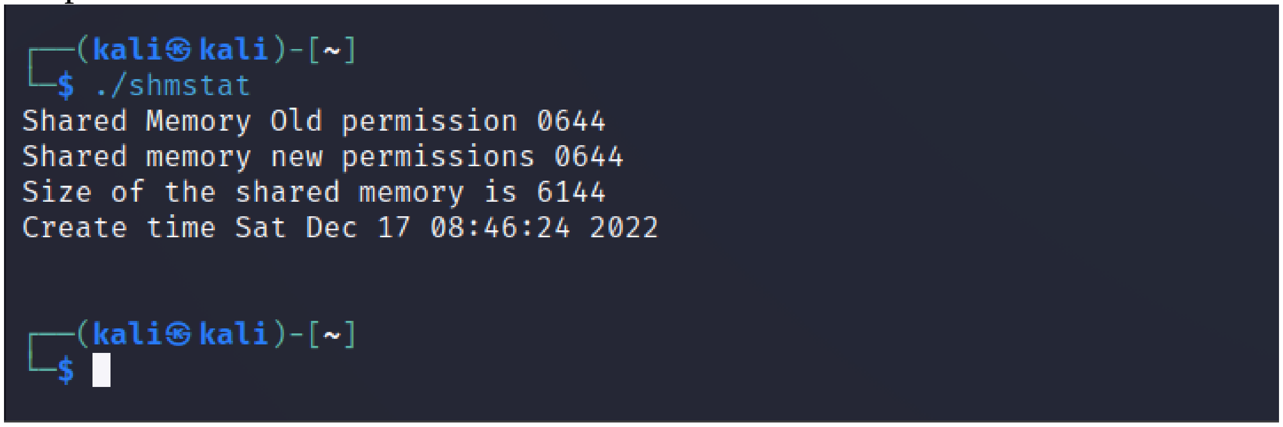
```
┌──(kali㉿kali)-[~]
└─$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 524293     kali       600        524288     2          dest
0x00000000 589831     kali       600        524288     2          dest
0x00000000 589832     kali       600        2097152    2          dest
0x00000000 589848     kali       600        524288     2          dest
0x00000000 491553     kali       600        524288     2          dest
0x00000000 491556     kali       600        524288     2          dest
0x00000000 491557     kali       600        2097152    2          dest
0x00000000 491560     kali       600        524288     2          dest
0x00000000 655403     kali       600        524288     2          dest
0x000003e7 655404     kali       644        6144       0
0x00000000 557109     kali       600        67108864   2          dest
0x00000000 491574     kali       600        524288     2          dest
0x00000000 491581     kali       600        524288     2          dest
0x00000000 557118     kali       600        524288     2          dest
```

4.7 Print the time of shared memory segment created



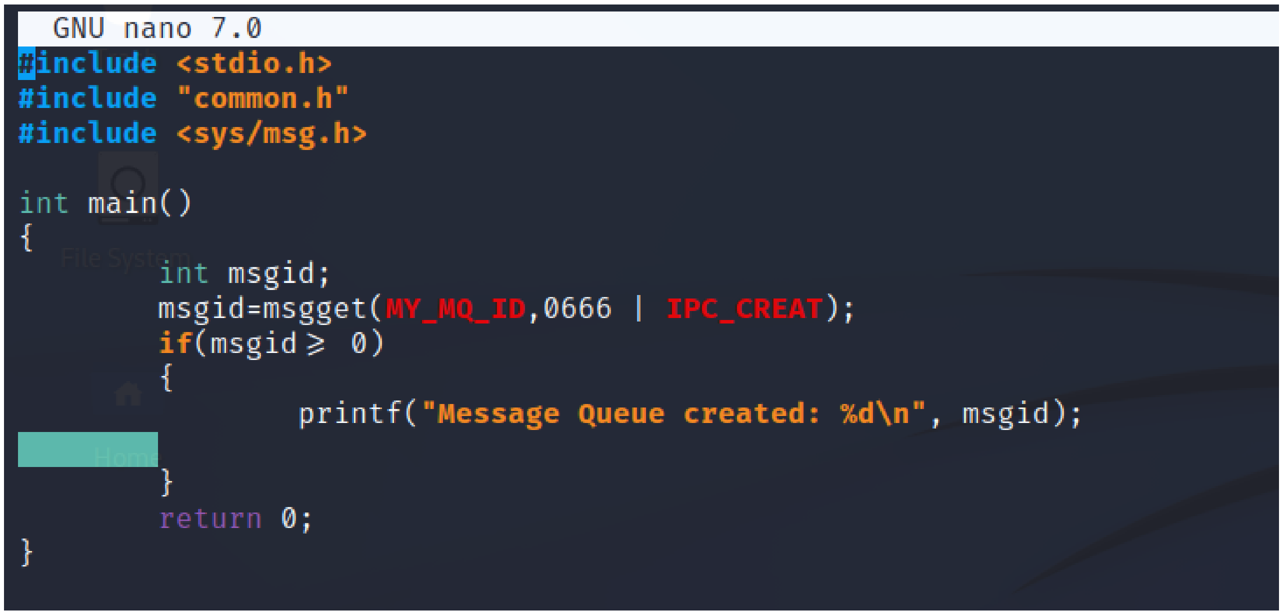
```c
#include <stdio.h>
#include <unistd.h>
#include <sys/shm.h>
#include "common.h"
#include <string.h>
#include <time.h>
int main()
{
        int shmid,ret;
        struct shmid_ds shmds;
        shmid=shmget(MY_SHM_ID,0,0);

        if (shmid >= 0)
        {
                ret=shmctl(shmid,IPC_STAT, &shmds);
                if(ret == 0)
                {
                        printf("Shared Memory Old permission 0%o\n",shmds.shm_perm.mode);
                        shmds.shm_perm.mode=0644;
                        ret=shmctl(shmid,IPC_SET,&shmds);
                        ret=shmctl(shmid,IPC_SET, &shmds);
                        printf("Shared memory new permissions 0%o\n", shmds.shm_perm.mode);
                        printf("Size of the shared memory is %d\n", shmds.shm_segsz);
                        printf("Create time %s\n",ctime(&shmds.shm_ctime));
                }
                else
                {
                        printf("shmctl() failed \n");
                }
        }
        else
        {
                printf("Shared Memory not found\n");
        }
        return 0;
}
```

Output:



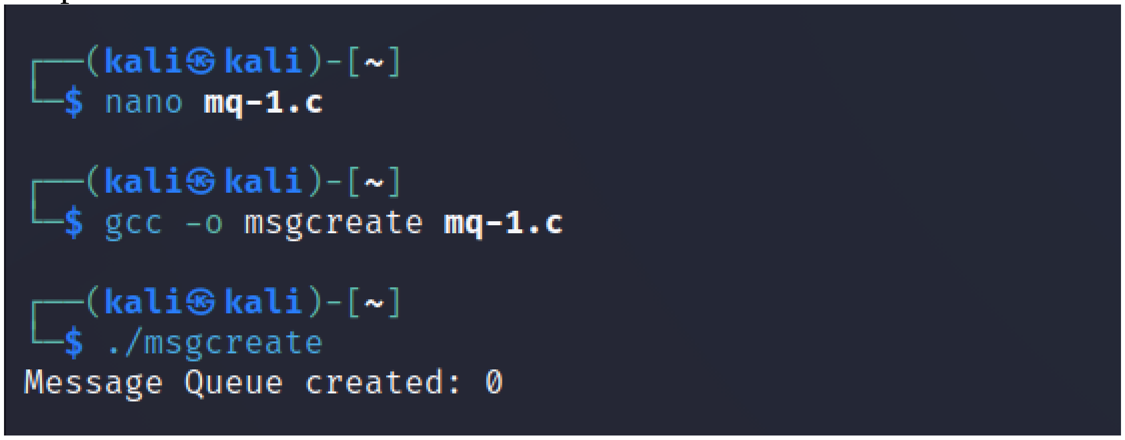
```
┌──(kali㉿kali)-[~]
└─$ ./shmstat
Shared Memory Old permission 0644
Shared memory new permissions 0644
Size of the shared memory is 6144
Create time Sat Dec 17 08:46:24 2022


┌──(kali㉿kali)-[~]
└─$ 
```

5 Write a program to create a message Queue



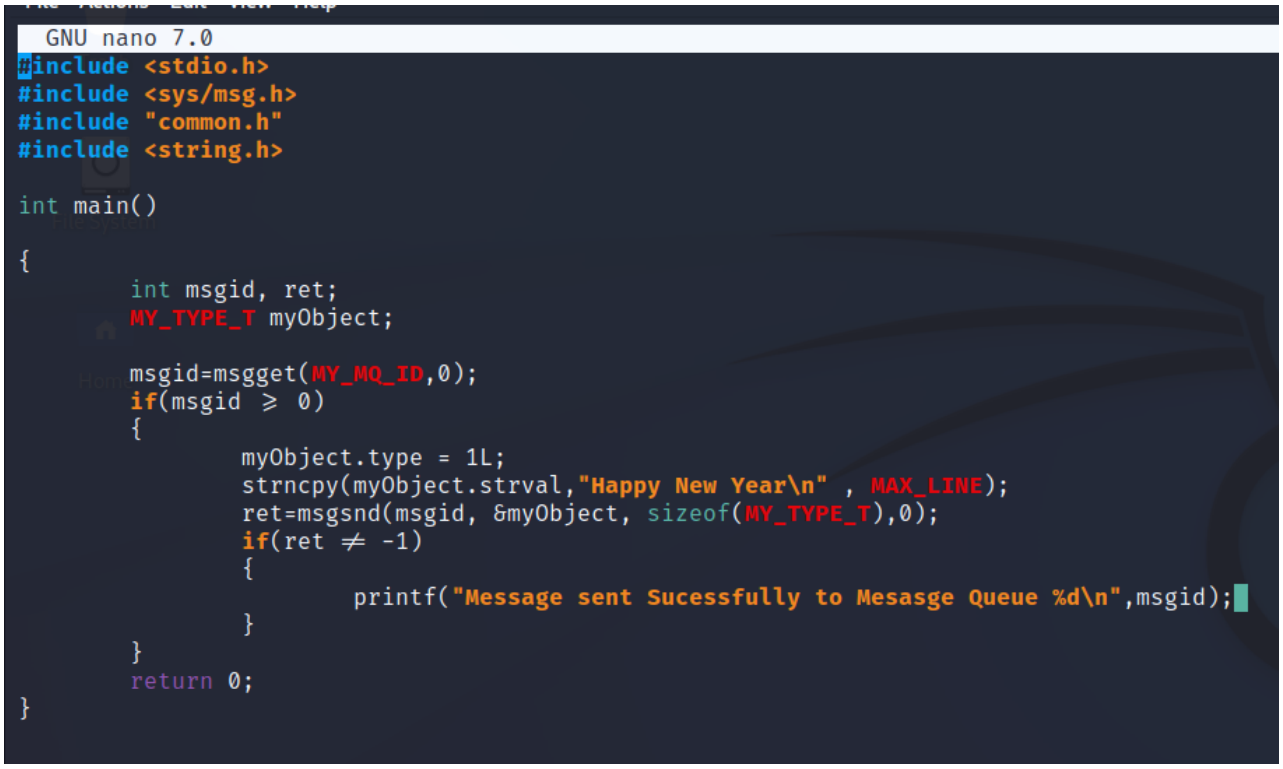
```
  GNU nano 7.0
#include <stdio.h>
#include "common.h"
#include <sys/msg.h>

int main()
{
        int msgid;
        msgid=msgget(MY_MQ_ID,0666 | IPC_CREAT);
        if(msgid >= 0)
        {
                printf("Message Queue created: %d\n", msgid);

        }
        return 0;
}
```

Output:



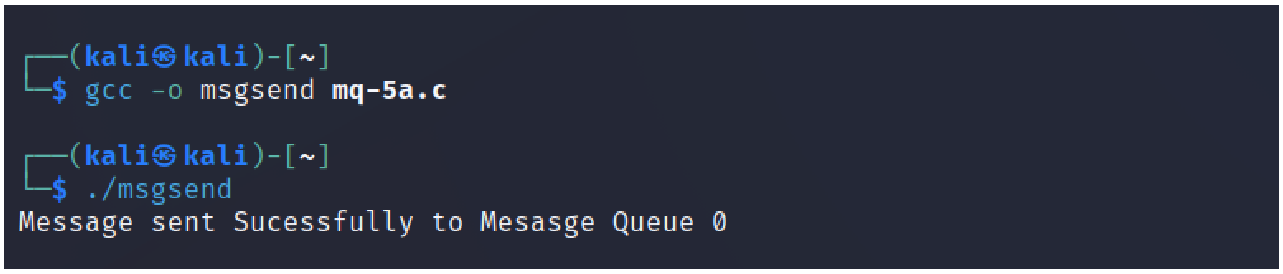
```
┌──(kali㉿kali)-[~]
└─$ nano mq-1.c

┌──(kali㉿kali)-[~]
└─$ gcc -o msgcreate mq-1.c

┌──(kali㉿kali)-[~]
└─$ ./msgcreate
Message Queue created: 0
```

5.1 Send a data “Happy new year” to that above Message Queue



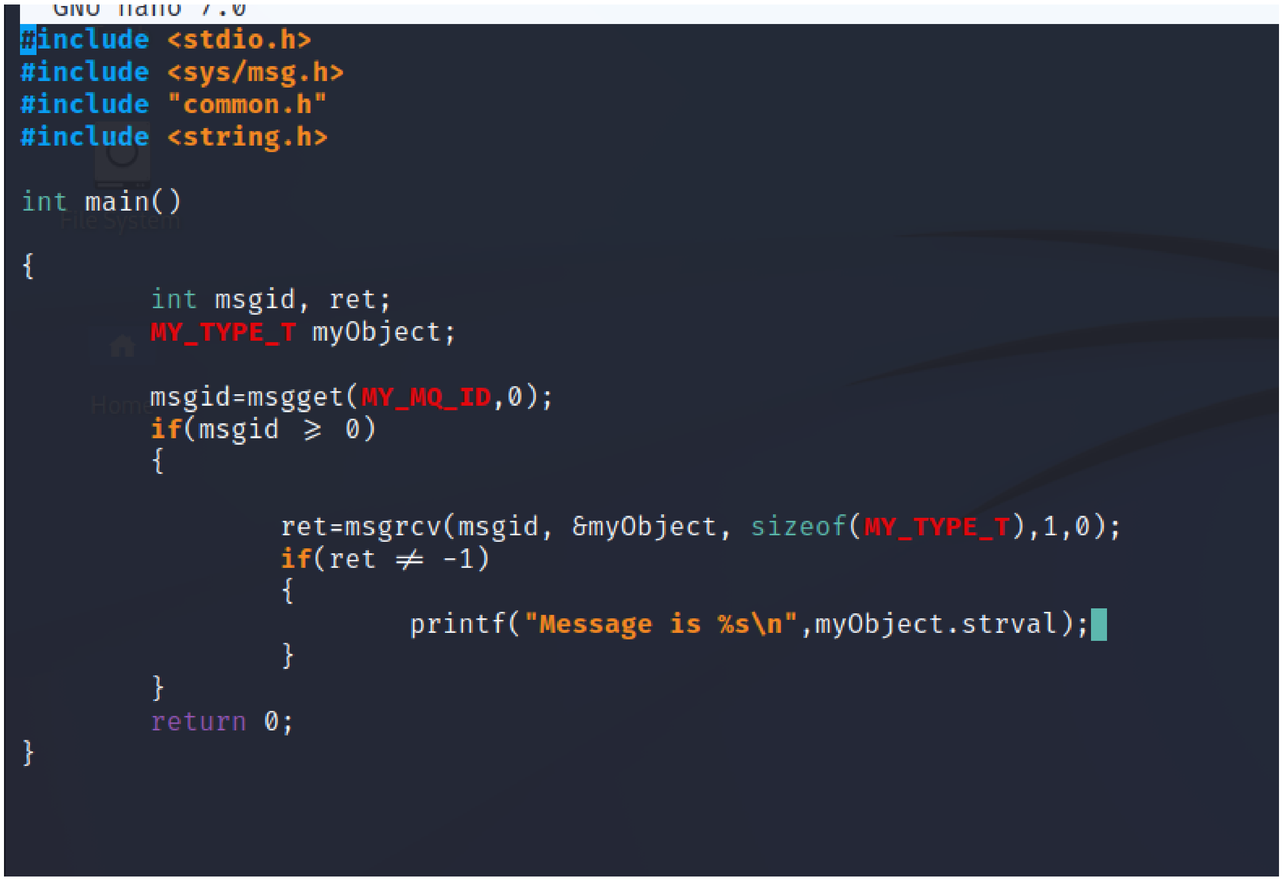
```
  GNU nano 7.0
#include <stdio.h>
#include <sys/msg.h>
#include "common.h"
#include <string.h>

int main()

{
        int msgid, ret;
        MY_TYPE_T myObject;

        msgid=msgget(MY_MQ_ID,0);
        if(msgid ⩾ 0)
        {
                myObject.type = 1L;
                strncpy(myObject.strval,"Happy New Year\n" , MAX_LINE);
                ret=msgsnd(msgid, &myObject, sizeof(MY_TYPE_T),0);
                if(ret ≠ -1)
                {
                        printf("Message sent Sucessfully to Mesasge Queue %d\n",msgid);
                }
        }
        return 0;
}
```

Output:



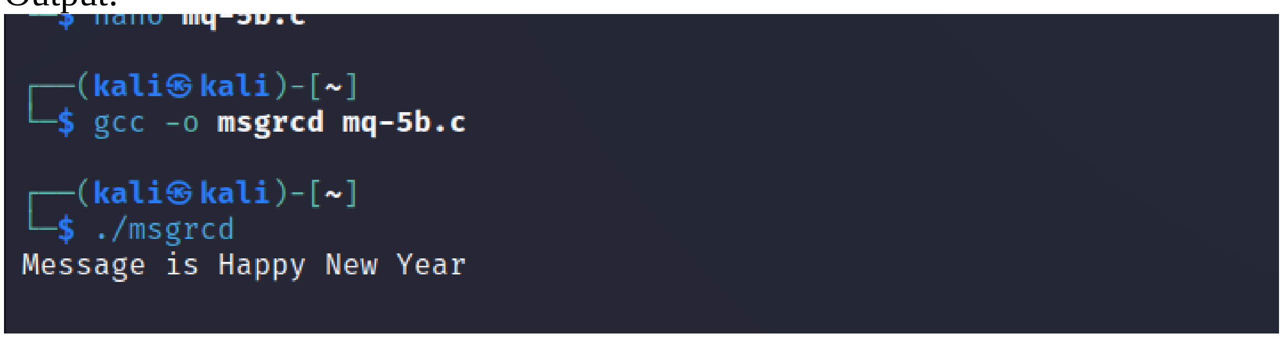
```
┌──(kali㉿kali)-[~]
└─$ gcc -o msgsend mq-5a.c

┌──(kali㉿kali)-[~]
└─$ ./msgsend
Message sent Sucessfully to Mesasge Queue 0
```

5.2 Receive that date from the queue and display it



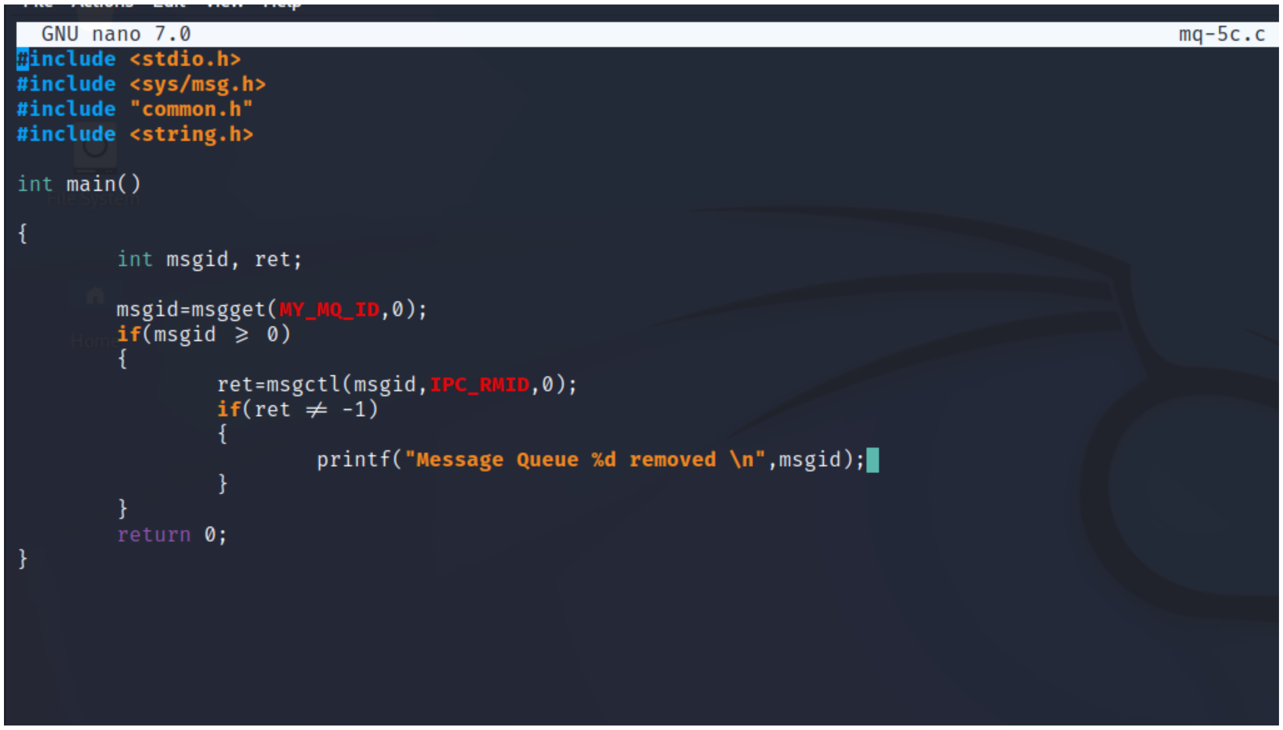
```
  GNU nano 7.0
#include <stdio.h>
#include <sys/msg.h>
#include "common.h"
#include <string.h>

int main()

{
        int msgid, ret;
        MY_TYPE_T myObject;

        msgid=msgget(MY_MQ_ID,0);
        if(msgid  ⩾  0)
        {

                ret=msgrcv(msgid, &myObject, sizeof(MY_TYPE_T),1,0);
                if(ret  ≠  -1)
                {
                        printf("Message is %s\n",myObject.strval);
                }
        }
        return 0;
}
```

Output:



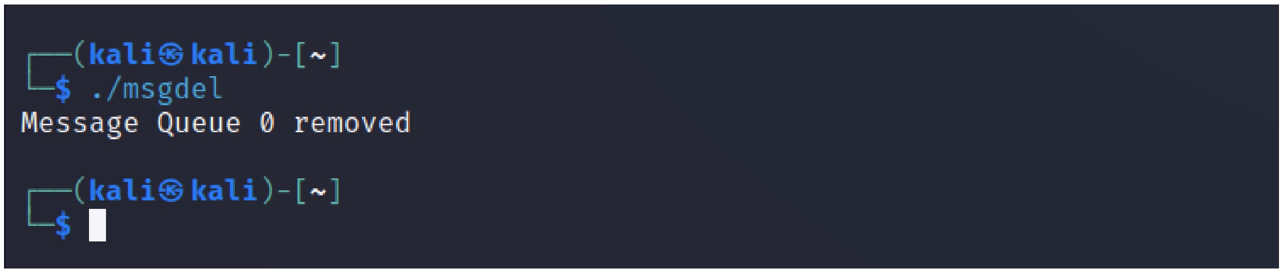
```
└─$ nano mq-5b.c

┌──(kali㉿kali)-[~]
└─$ gcc -o msgrcd mq-5b.c

┌──(kali㉿kali)-[~]
└─$ ./msgrcd
Message is Happy New Year
```

5.3 Delete the message Queue



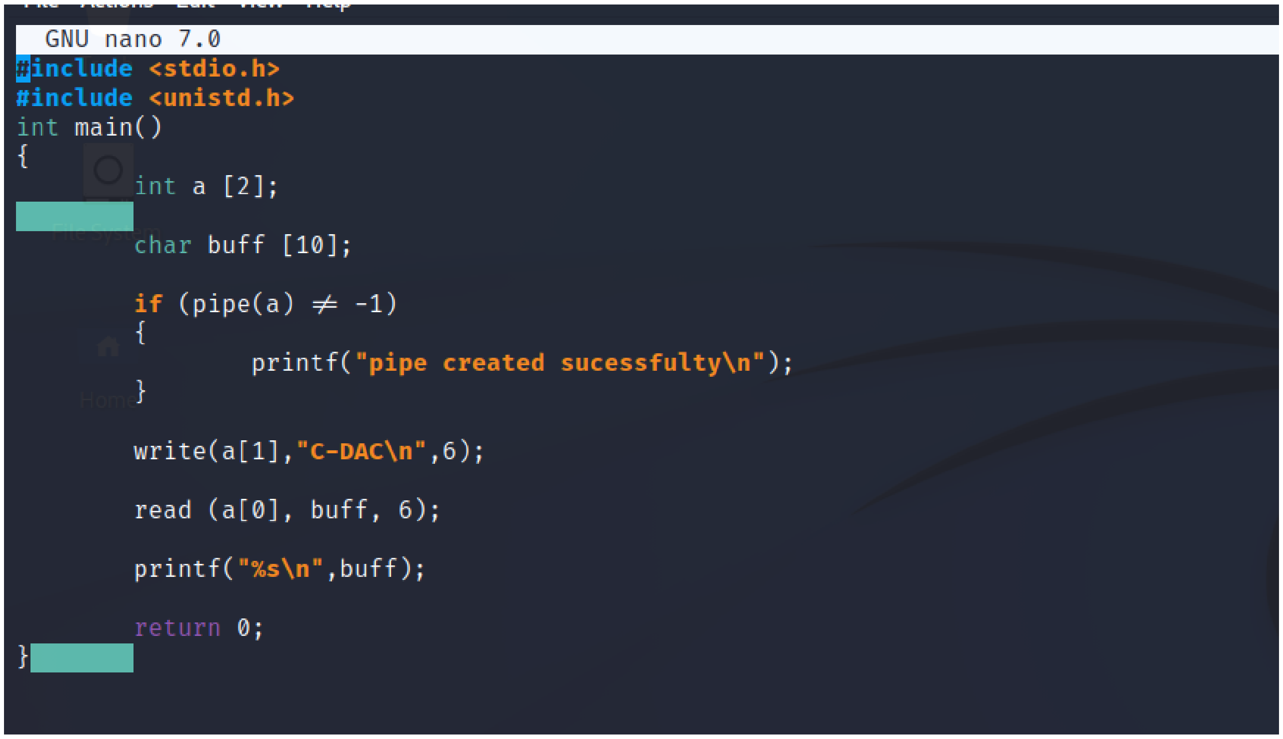
```
  GNU nano 7.0                                                                                    mq-5c.c
#include <stdio.h>
#include <sys/msg.h>
#include "common.h"
#include <string.h>

int main()

{
        int msgid, ret;

        msgid=msgget(MY_MQ_ID,0);
        if(msgid >= 0)
        {
                ret=msgctl(msgid,IPC_RMID,0);
                if(ret != -1)
                {
                        printf("Message Queue %d removed \n",msgid);
                }
        }
        return 0;
}
```

Output:



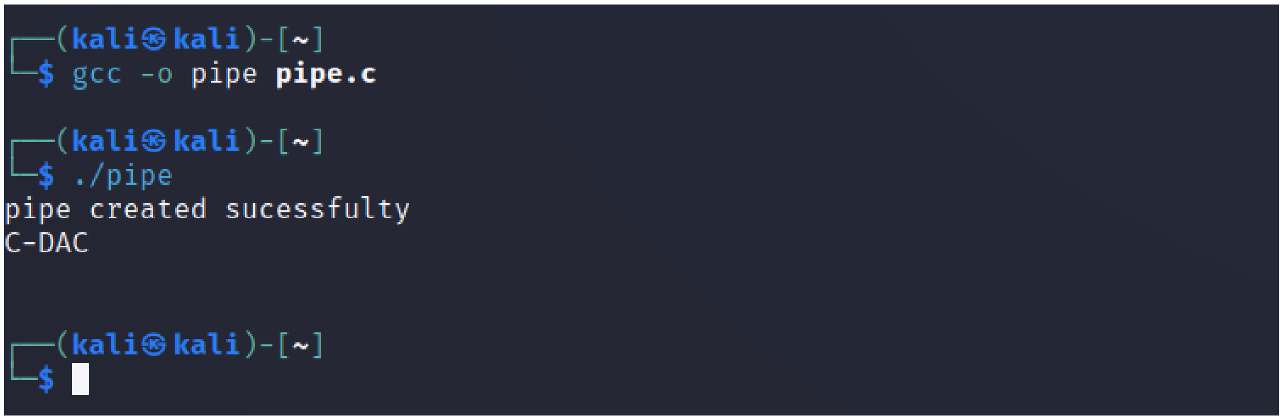
```
┌──(kali㉿kali)-[~]
└─$ ./msgdel
Message Queue 0 removed

┌──(kali㉿kali)-[~]
└─$ 
```

6 .Write a program to create an unnamed pipe.



```
GNU nano 7.0
#include <stdio.h>
#include <unistd.h>
int main()
{
        int a [2];

        char buff [10];

        if (pipe(a) ≠ -1)
        {
                printf("pipe created sucessfulty\n");
        }

        write(a[1],"C-DAC\n",6);

        read (a[0], buff, 6);

        printf("%s\n",buff);

        return 0;
}
```

Output:



```
┌──(kali㉿kali)-[~]
└─$ gcc -o pipe pipe.c

┌──(kali㉿kali)-[~]
└─$ ./pipe
pipe created sucessfulty
C-DAC


┌──(kali㉿kali)-[~]
└─$
```