OPERATING SYSTEM
LAB ASSIGNMENT-8

Avisek Mandal
102203700

Question 1
Write a program to implement producer consumer scenario using POSIX shared memory.

Code:

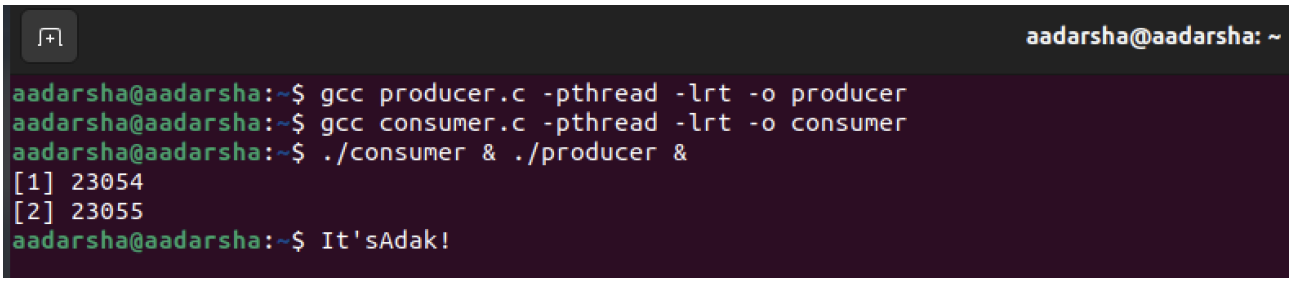
GNU nano 6.2 producer.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <sys/mman.h>
#include <unistd.h>
int main()
{

    const int SIZE = 4096;
    const char* name = "OS";
    const char* message_0 = "It's";
    const char* message_1 = "Adak!";
    int shm_fd;
    void* ptr;
    shm_fd = shm_open(name, O_CREAT | O_RDWR, 0666);
    ftruncate(shm_fd, SIZE);
    ptr = mmap(0, SIZE, PROT_WRITE, MAP_SHARED, shm_fd, 0);
    sprintf(ptr, "%s", message_0);
    ptr += strlen(message_0);
    sprintf(ptr, "%s", message_1);
    ptr += strlen(message_1);
    return 0;
}
```

GNU nano 6.2 consumer.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <sys/mman.h>
int main()
{

    const int SIZE = 4096;
    const char* name = "OS";
    int shm_fd;
    void* ptr;
    shm_fd = shm_open(name, O_RDONLY, 0666);
    ptr = mmap(0, SIZE, PROT_READ, MAP_SHARED, shm_fd, 0);
    printf("%s", (char*)ptr);
    shm_unlink(name);
    return 0;
}
```

OUTPUT:



QUESTION NO:2
Write a program to implement inter process communication between the parent process and the child process using ordinary Pipes.

Code:

```
#include<stdio.h>
#include<unistd.h>
int main() {
   int pipefds[2];
   int returnstatus;
   int pid;
   char writemessages[2][20]={"Hi", "Hello"};
   char readmessage[20];
   returnstatus = pipe(pipefds);
   if (returnstatus == -1) {
      printf("Unable to create pipe\n");
      return 1;
  }
   pid = fork();
   if (pid == 0)
{
      read(pipefds[0], readmessage, sizeof(readmessage));
      printf("Child Process - Reading from pipe \u2013 Message 1 is %s\n",
readmessa>
      read(pipefds[0], readmessage, sizeof(readmessage));
      printf("Child Process - Reading from pipe \u2013 Message 2 is %s\n",
readmessa>
   }
else
{
printf("Parent Process - Writing to pipe - Message 1 is %s\n", writemessa>
      write(pipefds[1], writemessages[0], sizeof(writemessages[0]));
      printf("Parent Process - Writing to pipe - Message 2 is %s\n", writemessa>
      write(pipefds[1], writemessages[1], sizeof(writemessages[1]));
```

```
  }
  return 0;
}
```

```
 GNU nano 6.2                                                    ass8q2.c
#include<stdio.h>
#include<unistd.h>
int main() {
   int pipefds[2];
   int returnstatus;
   int pid;
   char writemessages[2][20]={"Hi", "Hello"};
   char readmessage[20];
   returnstatus = pipe(pipefds);
   if (returnstatus == -1) {
      printf("Unable to create pipe\n");
      return 1;
  }
   pid = fork();
   if (pid == 0) {
      read(pipefds[0], readmessage, sizeof(readmessage));
      printf("Child Process - Reading from pipe - Message 1 is %s\n", readmessage);
      read(pipefds[0], readmessage, sizeof(readmessage));
      printf("Child Process - Reading from pipe - Message 2 is %s\n", readmessage);
   } else {
      printf("Parent Process - Writing to pipe - Message 1 is %s\n", writemessages[0]);
      write(pipefds[1], writemessages[0], sizeof(writemessages[0]));
      printf("Parent Process - Writing to pipe - Message 2 is %s\n", writemessages[1]);
      write(pipefds[1], writemessages[1], sizeof(writemessages[1]));
   }
   return 0;
}
```

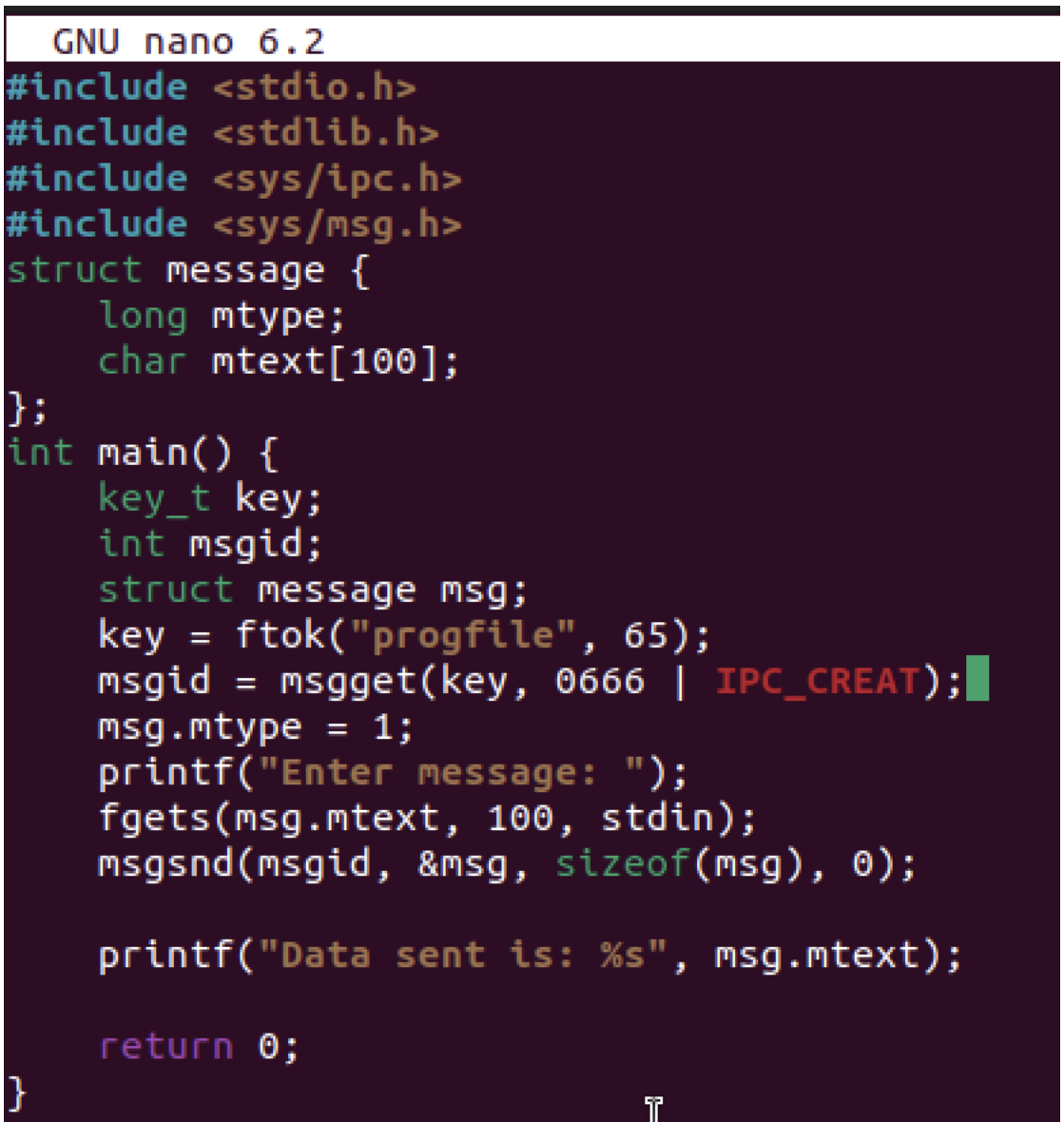
OUTPUT:

```
aadarsha@aadarsha:~$ nano ass8q2.c
aadarsha@aadarsha:~$ gcc ass8q2.c o- ass8q2
gcc: error: -E or -x required when input is from standard input
aadarsha@aadarsha:~$ chmod 777 ass8q2.c
aadarsha@aadarsha:~$ gcc ass8q2.c o- ass8q2
gcc: error: -E or -x required when input is from standard input
aadarsha@aadarsha:~$ ./a.out
Error opening source file: No such file or directory
aadarsha@aadarsha:~$ gcc ass8q2.c
aadarsha@aadarsha:~$ ./a.out
Parent Process - Writing to pipe - Message 1 is Hi
Parent Process - Writing to pipe - Message 2 is Hello
aadarsha@aadarsha:~$ Child Process - Reading from pipe - Message 1 is Hi
Child Process - Reading from pipe - Message 2 is Hello
```

QUESTION No:3
Write program to implement IPC through message queues.

CODE:
SENDER

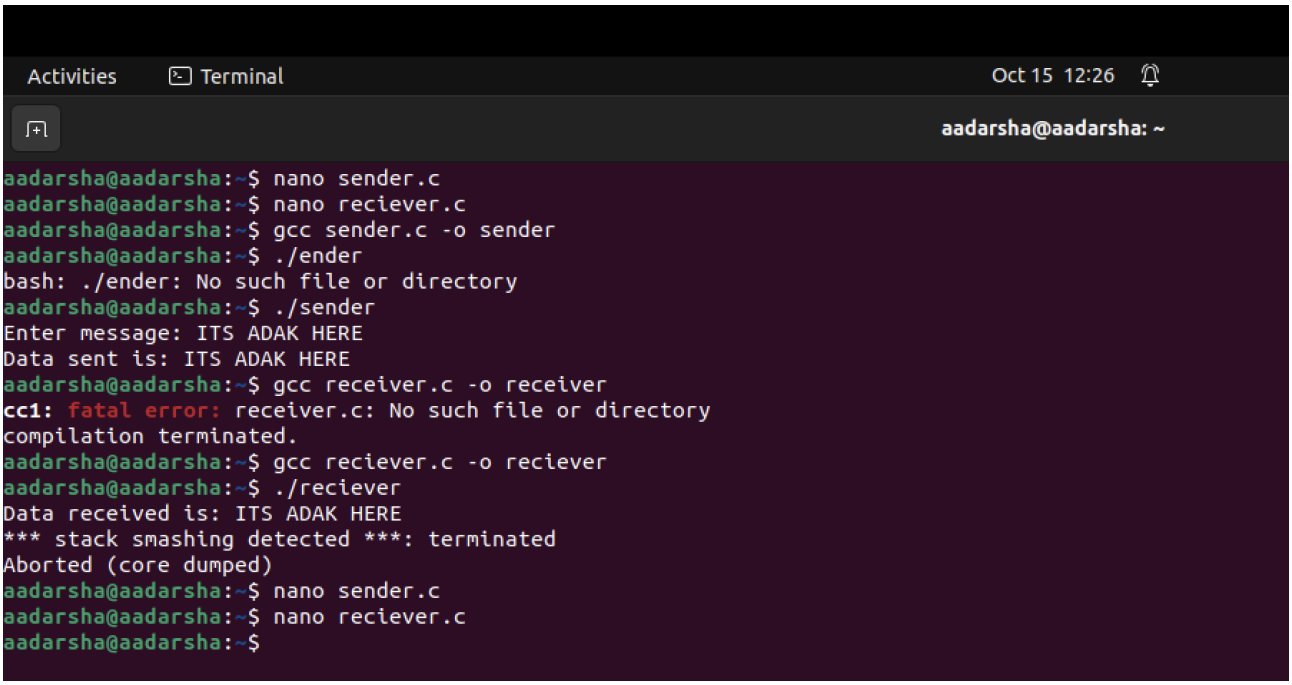
```
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/msg.h>
struct message {
    long mtype;
    char mtext[100];
};
int main() {
    key_t key;
    int msgid;
    struct message msg;
    key = ftok("progfile", 65);
    msgid = msgget(key, 0666 | IPC_CREAT);
    msg.mtype = 1;
    printf("Enter message: ");
    fgets(msg.mtext, 100, stdin);
    msgsnd(msgid, &msg, sizeof(msg), 0);
    printf("Data sent is: %s", msg.mtext);
    return 0;
}
```



RECEIVER

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/msg.h>
struct message {
    long mtype;
    char mtext[100];
};
int main() {
    key_t key;
    int msgid;
    struct message msg;
    key = ftok("progfile", 65);
    msgid = msgget(key, 0666 | IPC_CREAT);
    msgrcv(msgid, &msg, sizeof(msg), 1, 0);
    printf("Data received is: %s", msg.mtext);
    msgctl(msgid, IPC_RMID, NULL);
    return 0;
}
```

OUTPUT:


Activities
Terminal
Oct 15 12:26
aadarsha@aadarsha: ~
aadarsha@aadarsha:~$ nano sender.c
aadarsha@aadarsha:~$ nano reciever.c
aadarsha@aadarsha:~$ gcc sender.c -o sender
aadarsha@aadarsha:~$ ./ender
bash: ./ender: No such file or directory
aadarsha@aadarsha:~$ ./sender
Enter message: ITS ADAK HERE
Data sent is: ITS ADAK HERE
aadarsha@aadarsha:~$ gcc receiver.c -o receiver
cc1: fatal error: receiver.c: No such file or directory
compilation terminated.
aadarsha@aadarsha:~$ gcc reciever.c -o reciever
aadarsha@aadarsha:~$ ./reciever
Data received is: ITS ADAK HERE
*** stack smashing detected ***: terminated
Aborted (core dumped)
aadarsha@aadarsha:~$ nano sender.c
aadarsha@aadarsha:~$ nano reciever.c
aadarsha@aadarsha:~$