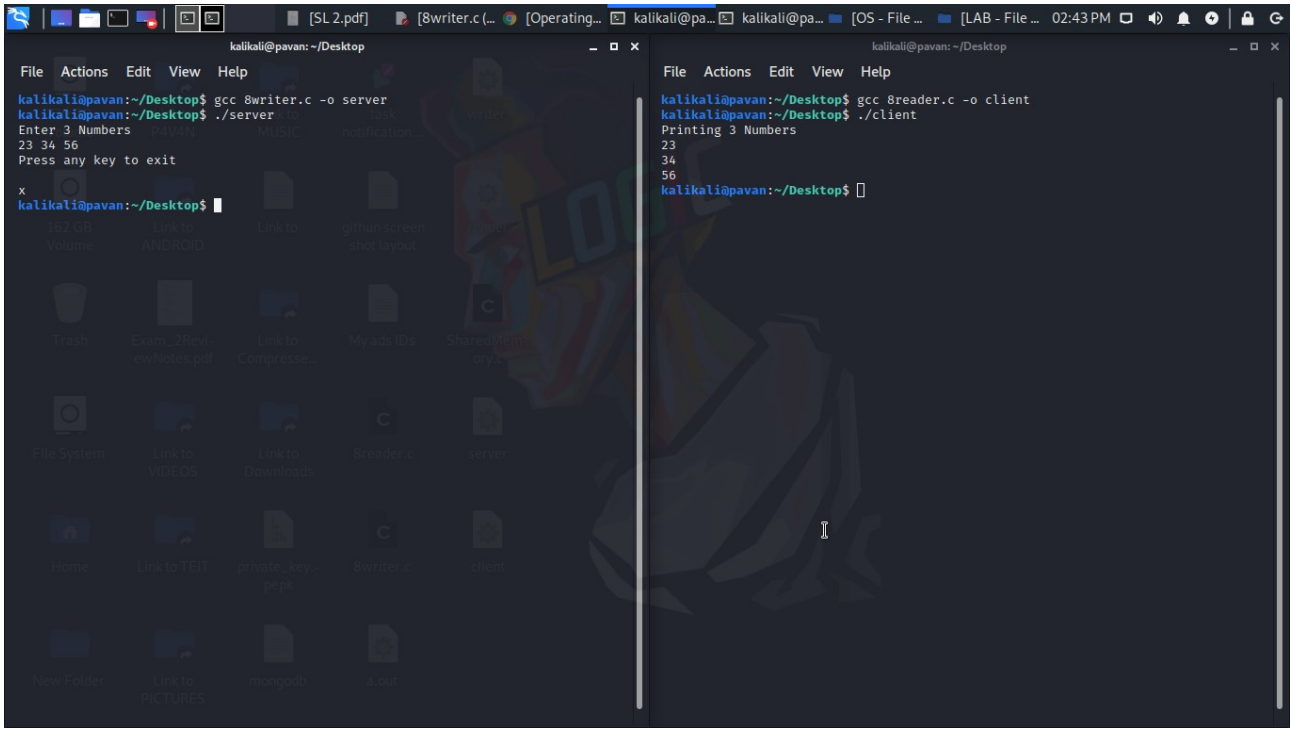
```
//         Reader
#include <stdio.h>
#include <stdlib.h>
#include "SharedMemory.c"

int main() {
   int shm_id, i;
   if ((shm_id = shm_init()) == -1) {
      perror("Error occured while initialising Shared Memory\n");
      exit(-1);
   }

   SharedMemory *mSharedMemory = attach(shm_id);

   if (mSharedMemory->status == READ_BY_CLIENT) {
      printf("Server hasn't written value yet\n");
      exit(-1);
   }

   printf("Printing %d Numbers\n", ARRAY_LENGTH);
   for (i = 0;i < ARRAY_LENGTH;i++) {
      printf("%d\n", mSharedMemory->array[i]);
   }

   mSharedMemory->status = READ_BY_CLIENT;

   if (detach(mSharedMemory) == -1) {
      perror("Error occured while detaching Shared memory\n");
      exit(-1);
   }

}

// Writer

#include <stdio.h>
#include <stdlib.h>
#include "SharedMemory.c"

int main() {
   int shm_id, i;
   if ((shm_id = shm_init()) == -1) {
      perror("Error occured while initialising Shared Memory\n");
      exit(-1);
   }

   SharedMemory *mSharedMemory = attach(shm_id);

   if (mSharedMemory->status == WRITTEN_BY_SERVER) {
      printf("Client hasn't read value yet\n");
      exit(-1);
   }
```

```
    printf("Enter %d Numbers\n", ARRAY_LENGTH);
    for (i = 0;i < ARRAY_LENGTH;i++) {
        scanf("%d", &mSharedMemory->array[i]);
    }

    mSharedMemory->status = WRITTEN_BY_SERVER;

    if (detach(mSharedMemory) == -1) {
        perror("Error occured while detaching Shared memory\n");
        exit(-1);
    }

    char c;
    printf("Press any key to exit\n");
    scanf(" %c", &c);

}

//          Shared Memory

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

#define PROJECT_ID 209

#define READ_BY_CLIENT 0
#define WRITTEN_BY_SERVER 1

#define ARRAY_LENGTH 3

// Status holds either value READ_BY_CLIENT or WRITTEN_BY_SERVER
// Server writes into an array of ARRAY_LENGTH and the client reads this
typedef struct SharedMemory {
    int status;
    int array[ARRAY_LENGTH];
}SharedMemory;

key_t getKey() {
    return ftok(".", PROJECT_ID);
}

int shm_init() {
    return shmget(getKey(), sizeof(SharedMemory), IPC_CREAT | 0666);
}

SharedMemory *attach(int shm_id) {
    return (SharedMemory *) shmat(shm_id, NULL, 0);
}

int detach(SharedMemory *shm) {
    return shmdt((void *) shm);
```

}


[SL 2.pdf]
[8writer.c (...
[Operating...
kalikali@pa...
kalikali@pa...
[OS - File ...
[LAB - File ...
02:43 PM
kalikali@pavan: ~/Desktop
File Actions Edit View Help
kalikali@pavan:~/Desktop$ gcc 8writer.c -o server
kalikali@pavan:~/Desktop$ ./server
Enter 3 Numbers
23 34 56
Press any key to exit
x
kalikali@pavan:~/Desktop$
kalikali@pavan: ~/Desktop
File Actions Edit View Help
kalikali@pavan:~/Desktop$ gcc 8reader.c -o client
kalikali@pavan:~/Desktop$ ./client
Printing 3 Numbers
23
34
56
kalikali@pavan:~/Desktop$