

Tên	Đinh Phương My
MSSV	52100703
Nhóm thực hành	N101

BÀI BÁO CÁO THỰC HÀNH LAB 5.3

Câu 1: Tiến trình cha chuyển đổi số đầu tiên (argv [1]) là một số nguyên lớn hơn 3 cho tiến trình con thông qua vùng nhớ chia sẻ. Tiến trình con nhận, tính giá trị $n! = 1 * 2 * \dots * n$ và ghi nó vào vùng nhớ chia sẻ. Tiến trình cha nhận và xuất dữ liệu ra màn hình..

```
>./baitap2A.out 4
4! = 24
```

Share Memory

Code:

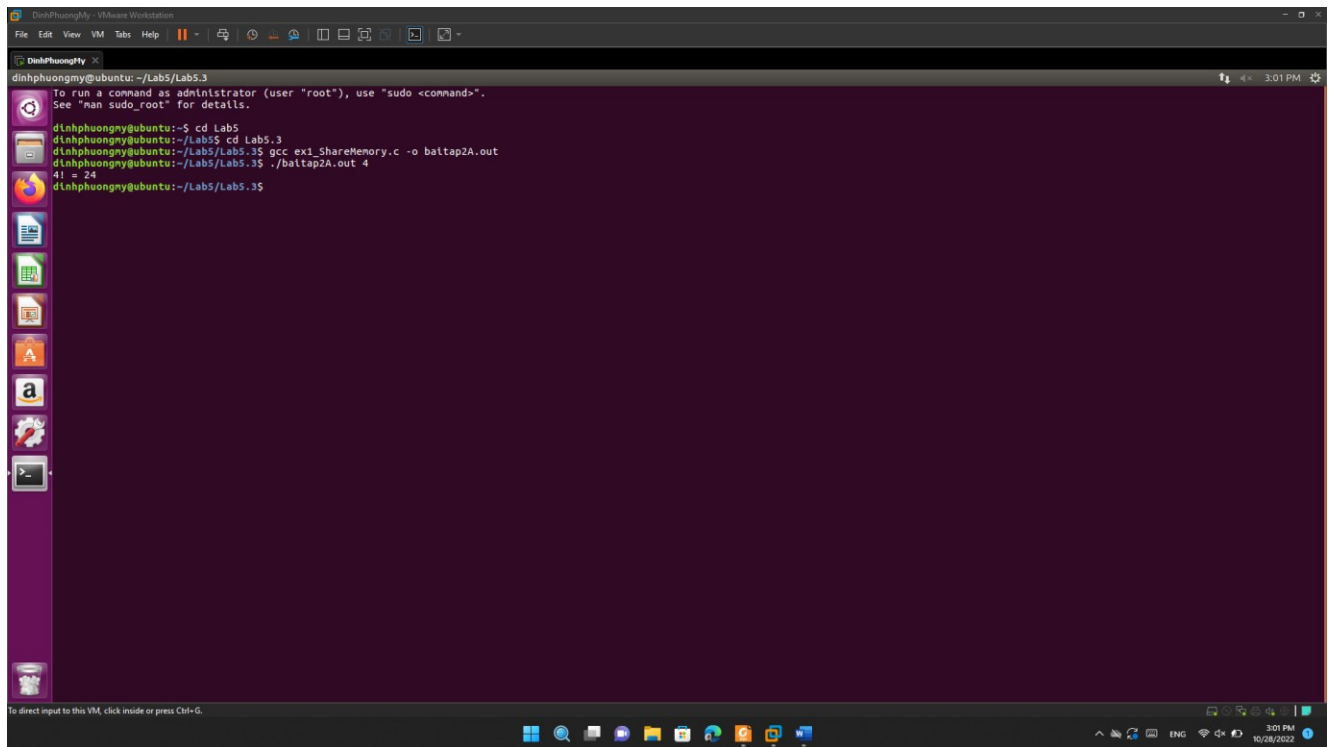
```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>

int main(int argc, char* argv[])
{
    int *shn, shmid, pid;
    key_t key;
    if((key = ftok(".", 65)) == -1)
    {
        printf("Key created.\n");
        return 1;
    }
    shmid = shmget(key, 256, IPC_CREAT | 0666);
    if (shmid == -1)
    {
        printf("Shared memory created.\n");
        return 2;
    }
    shn = (int*) shmop(shmid, 0, 0);
    pid = fork();
    if(pid == 0)
    {
        int s = 1, i;
        for(i = 1; i <= shn[0]; i++)
            s *= i;
        shn[1] = s;
        shmctl((void*) shn);
        return 0;
    }
    else if(pid > 0)
    {
        shn[0] = atoi(argv[1]);
        sleep(3);
        printf("xdi = %d\n", shn[0], shn[1]);
        shmctl((void*) shn);
        shmctl(shmid, IPC_RMID, (struct shmld_ds*)0);
        return 0;
    }
    else
    {
        printf("Fork failed.");
        return 4;
    }
    return 0;
}

```

Run:

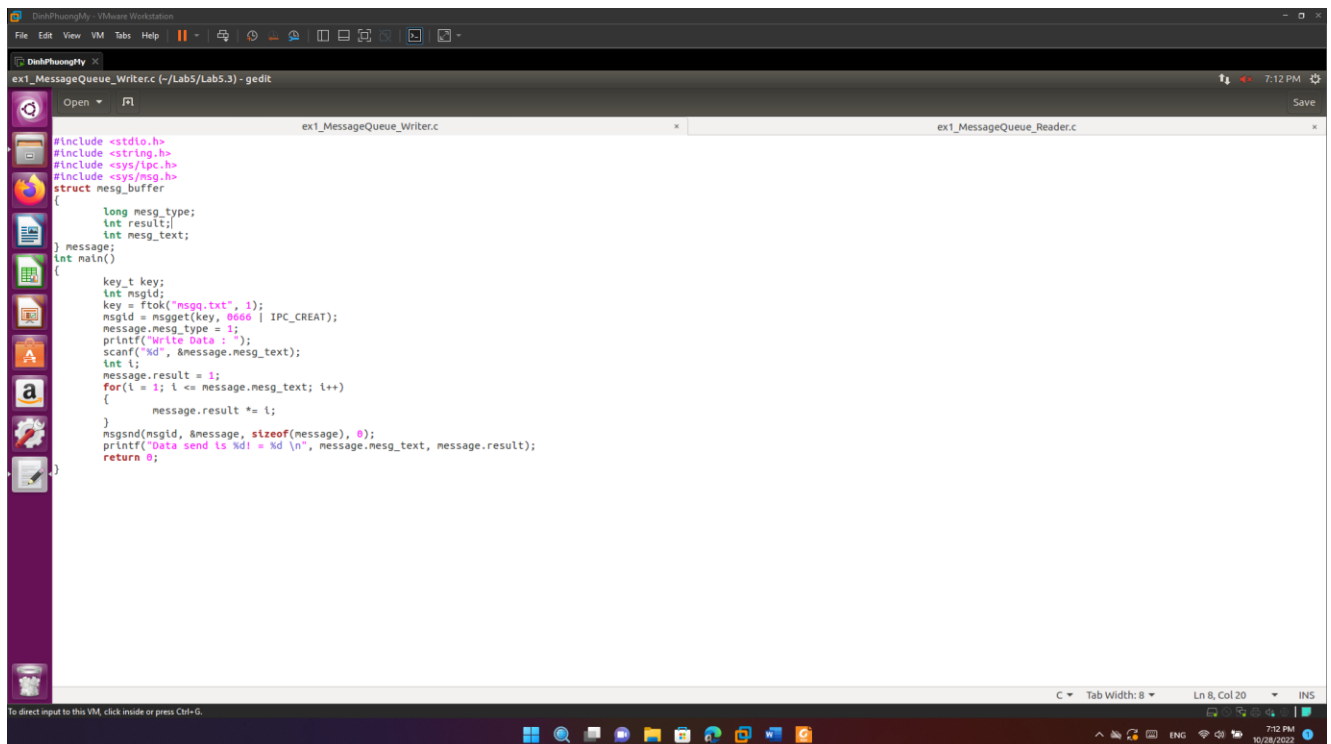


```
dinhphuongmy@ubuntu: ~/Lab5/Lab5.3
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dinhphuongmy@ubuntu:~/Lab5$ cd Lab5
dinhphuongmy@ubuntu:~/Lab5$ gcc ex1_ShareMemory.c -o battap2A.out
dinhphuongmy@ubuntu:~/Lab5$ ./battap2A.out 4
41 = 24
dinhphuongmy@ubuntu:~/Lab5$
```

Message Queue

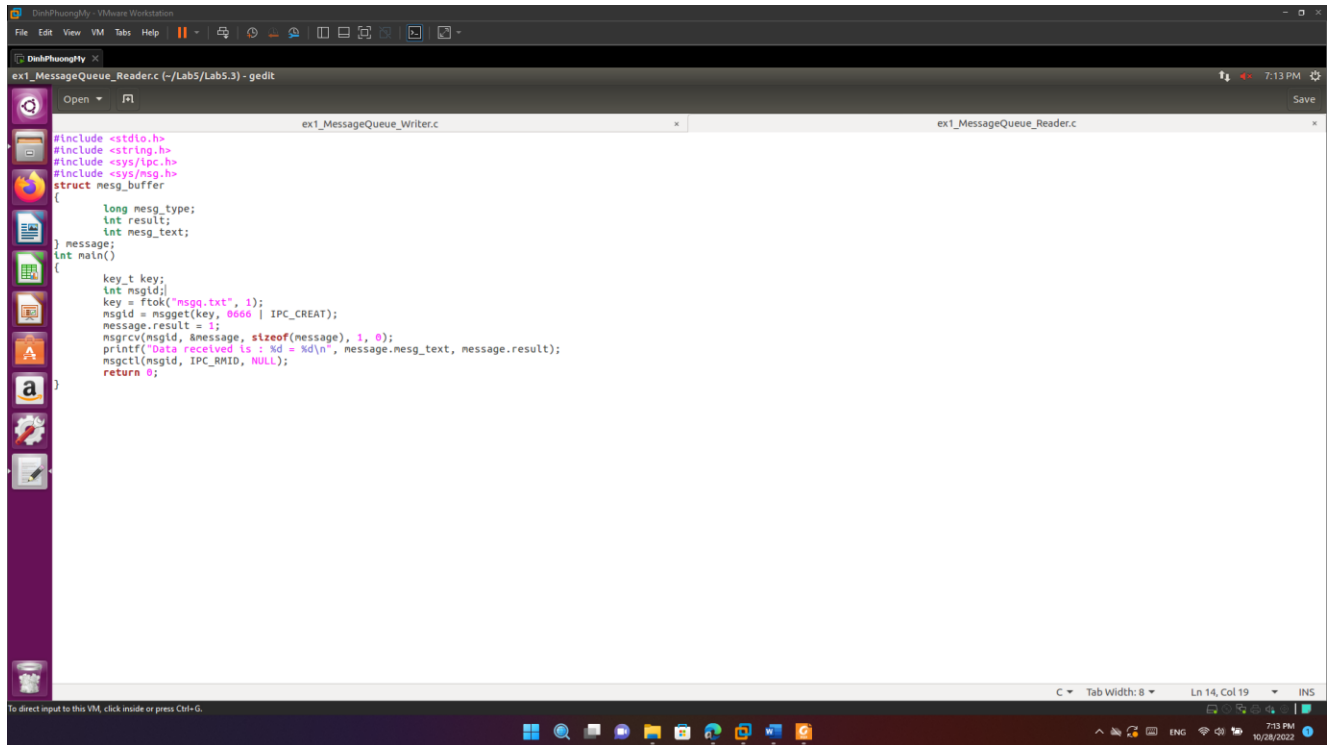
Writer



```
ex1_MessageQueue_Writer.c (-/Lab5/Lab5.3) - gedit
ex1_MessageQueue_Writer.c
ex1_MessageQueue_Reader.c

#include <stdio.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/msg.h>
struct msg_buffer
{
    long msg_type;
    int result;
    int msg_text;
} message;
int main()
{
    key_t key;
    int msgId;
    key = ftok("msgq.txt", 1);
    msgId = msgget(key, 0666 | IPC_CREAT);
    message.msg_type = 1;
    printf("Write Data : ");
    scanf("%d", &message.msg_text);
    int i;
    message.result = 1;
    for(i = 1; i <= message.msg_text; i++)
    {
        message.result *= i;
    }
    msgsnd(msgId, &message, sizeof(message), 0);
    printf("Data send is %d! = %d \n", message.msg_text, message.result);
    return 0;
}
```

Reader:

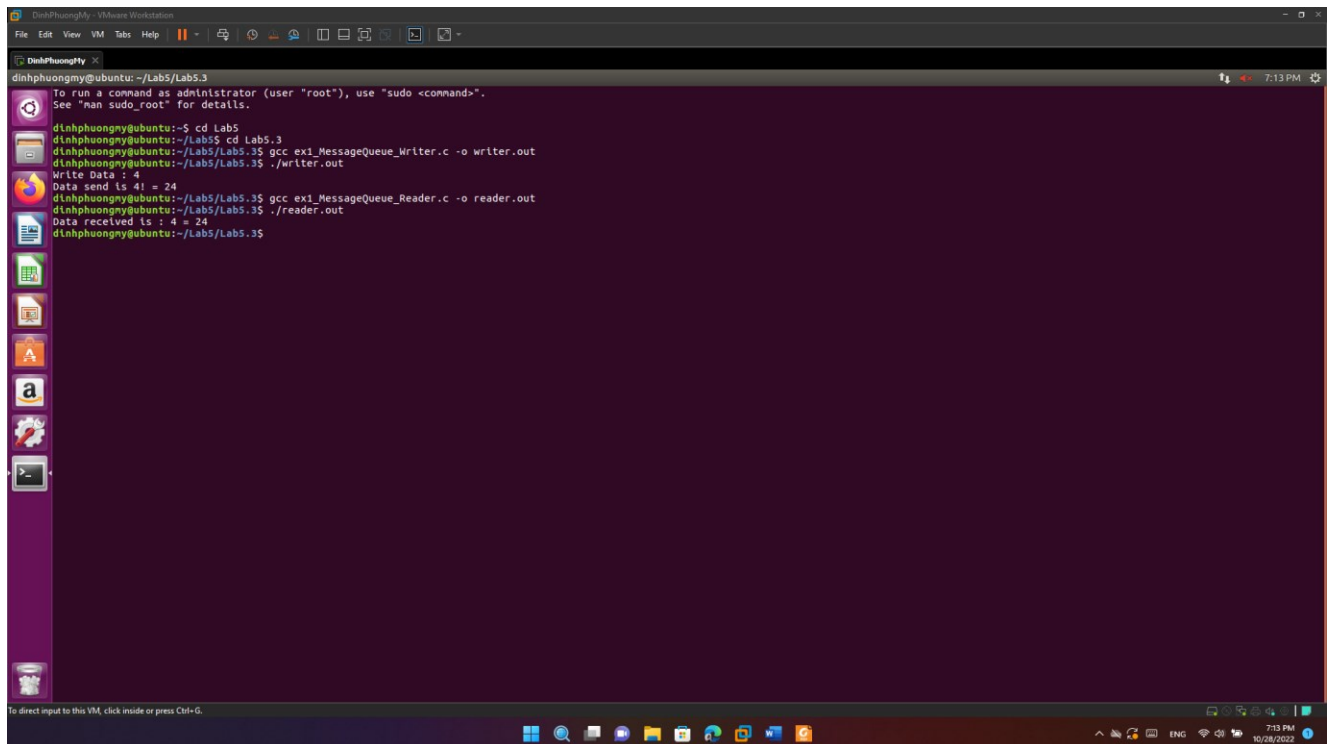


The screenshot shows a VMware Workstation window titled "DinhPhuongMy - VMware Workstation". Inside, a terminal window titled "DinhPhuongMy" is open, displaying the code for "ex1_MessageQueue_Reader.c". The code is as follows:

```
#include <stdio.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/msg.h>
struct msg_buffer
{
    long msg_type;
    int result;
    int msg_text;
} message;
int main()
{
    key_t key;
    int msgId;
    key = ftok("msgq.txt", 1);
    msgId = msgget(key, 0666 | IPC_CREAT);
    message.result = 1;
    msgrcv(msgId, &message, sizeof(message), 1, 0);
    printf("Data received is : %d = %d\n", message.msg_text, message.result);
    msgctl(msgId, IPC_RMID, NULL);
    return 0;
}
```

The terminal window also shows the file name "ex1_MessageQueue_Reader.c" and the path "~/Lab5/Lab5.3". The status bar at the bottom indicates "Ln 14, Col 19" and "INS".

Run:



The screenshot shows a VMware Workstation window titled "DinhPhuongMy - VMware Workstation". Inside, a terminal window titled "DinhPhuongMy" is open, displaying the execution of the Reader program. The output is as follows:

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dinhphuongmy@ubuntu:~/Lab5$ cd Lab5
dinhphuongmy@ubuntu:~/Lab5$ gcc ex1_MessageQueue_Writer.c -o writer.out
dinhphuongmy@ubuntu:~/Lab5$ ./writer.out
Write Data : 4
Data send is 41 = 24
dinhphuongmy@ubuntu:~/Lab5$ gcc ex1_MessageQueue_Reader.c -o reader.out
dinhphuongmy@ubuntu:~/Lab5$ ./reader.out
Data received is : 4 = 24
dinhphuongmy@ubuntu:~/Lab5$
```

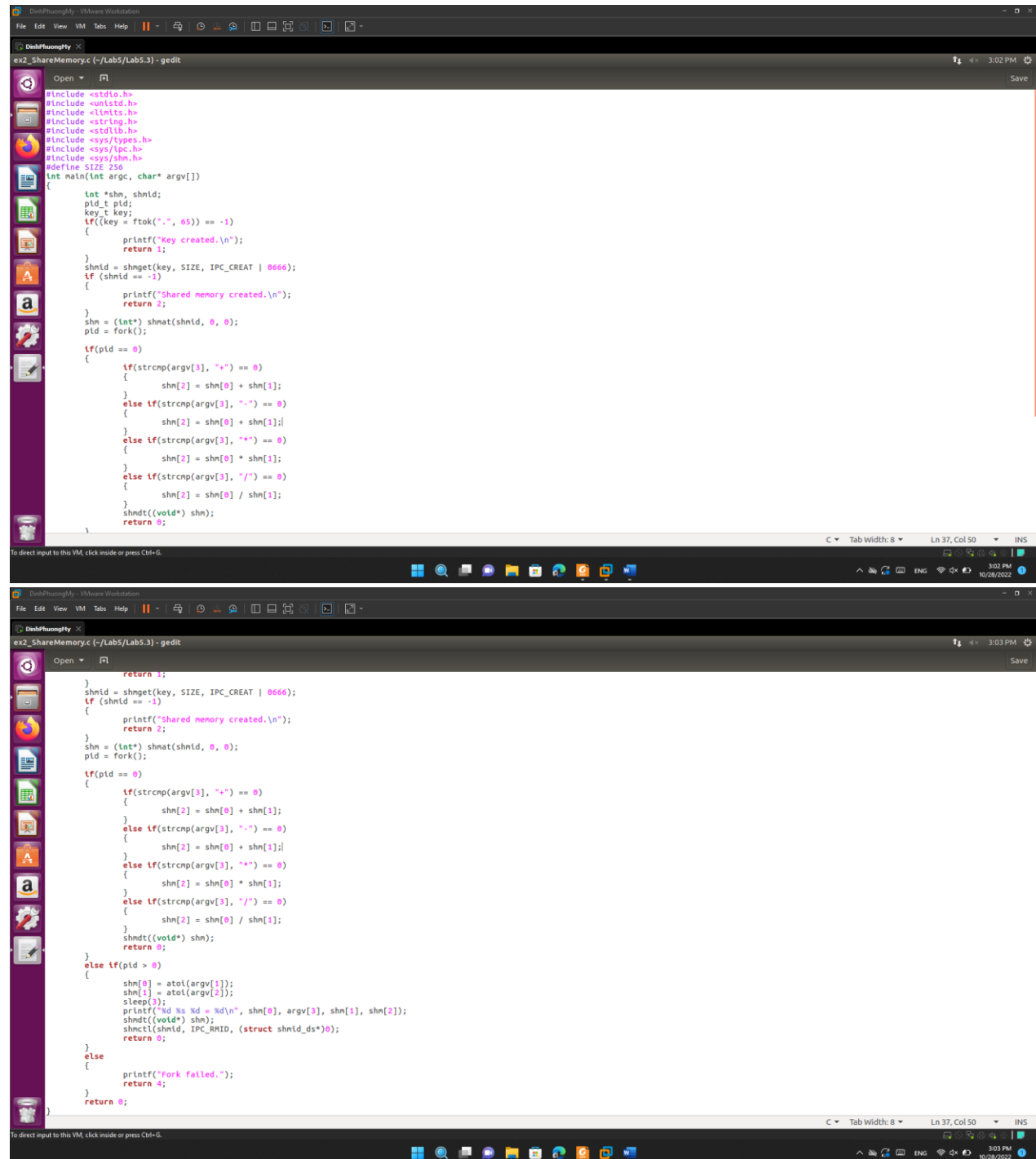
The terminal window also shows the file name "ex1_MessageQueue_Reader.c" and the path "~/Lab5/Lab5.3". The status bar at the bottom indicates "Ln 14, Col 19" and "INS".

Câu 2: Tiến trình cha đọc hai số nguyên và một thao tác +, -, *, / và chuyển tất cả cho tiến trình con. Tiến trình con tính toán kết quả và trả về cho tiến trình cha để in ra màn hình. Việc liên lạc sử dụng một vùng nhớ chia sẻ.

```
> ./baitap3A.out 4 6 +  
4 + 6 = 10
```

Share Memory

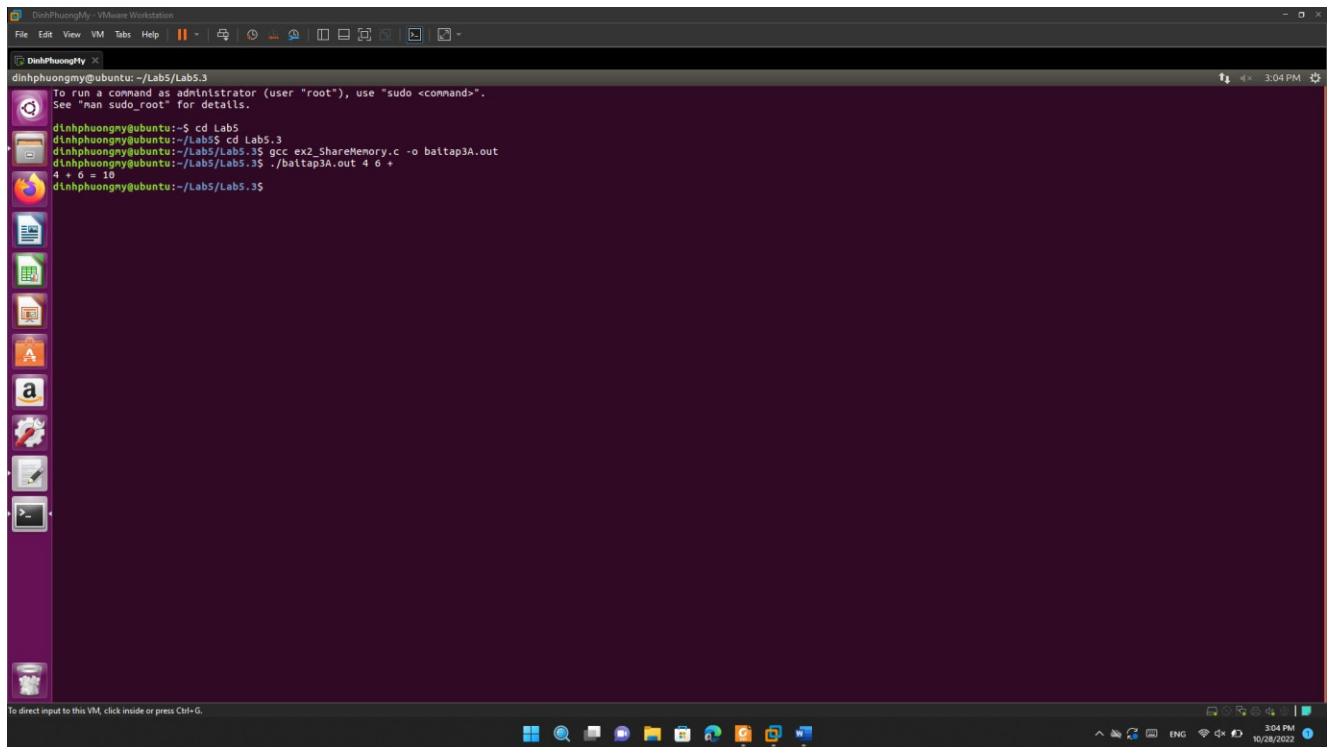
Code:



```
#include <stdio.h>
#include <unistd.h>
#include <limits.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#define SIZE 256

int main(int argc, char* argv[])
{
    int *shm, shmid;
    pid_t pid;
    key_t key;
    if((key = ftok(".", 65)) == -1)
    {
        printf("Key created.\n");
        return 1;
    }
    shmid = shmget(key, SIZE, IPC_CREAT | 0666);
    if (shmid == -1)
    {
        printf("Shared memory created.\n");
        return 2;
    }
    shm = (int*) shmat(shmid, 0, 0);
    pid = fork();
    if(pid == 0)
    {
        if(strcmp(argv[3], "+") == 0)
        {
            shm[2] = shm[0] + shm[1];
        }
        else if(strcmp(argv[3], "-") == 0)
        {
            shm[2] = shm[0] - shm[1];
        }
        else if(strcmp(argv[3], "*") == 0)
        {
            shm[2] = shm[0] * shm[1];
        }
        else if(strcmp(argv[3], "/") == 0)
        {
            shm[2] = shm[0] / shm[1];
        }
        shmdt((void*) shm);
        return 0;
    }
    else if(pid > 0)
    {
        shm[0] = atoi(argv[1]);
        shm[1] = atoi(argv[2]);
        sleep(3);
        printf("sd ss sd = %d\n", shm[0], argv[3], shm[1], shm[2]);
        shmdt((void*) shm);
        shmctl(shmid, IPC_RMID, (struct shm_id*)0);
        return 0;
    }
    else
    {
        printf("Fork failed.");
        return 4;
    }
    return 0;
}
```

Run:

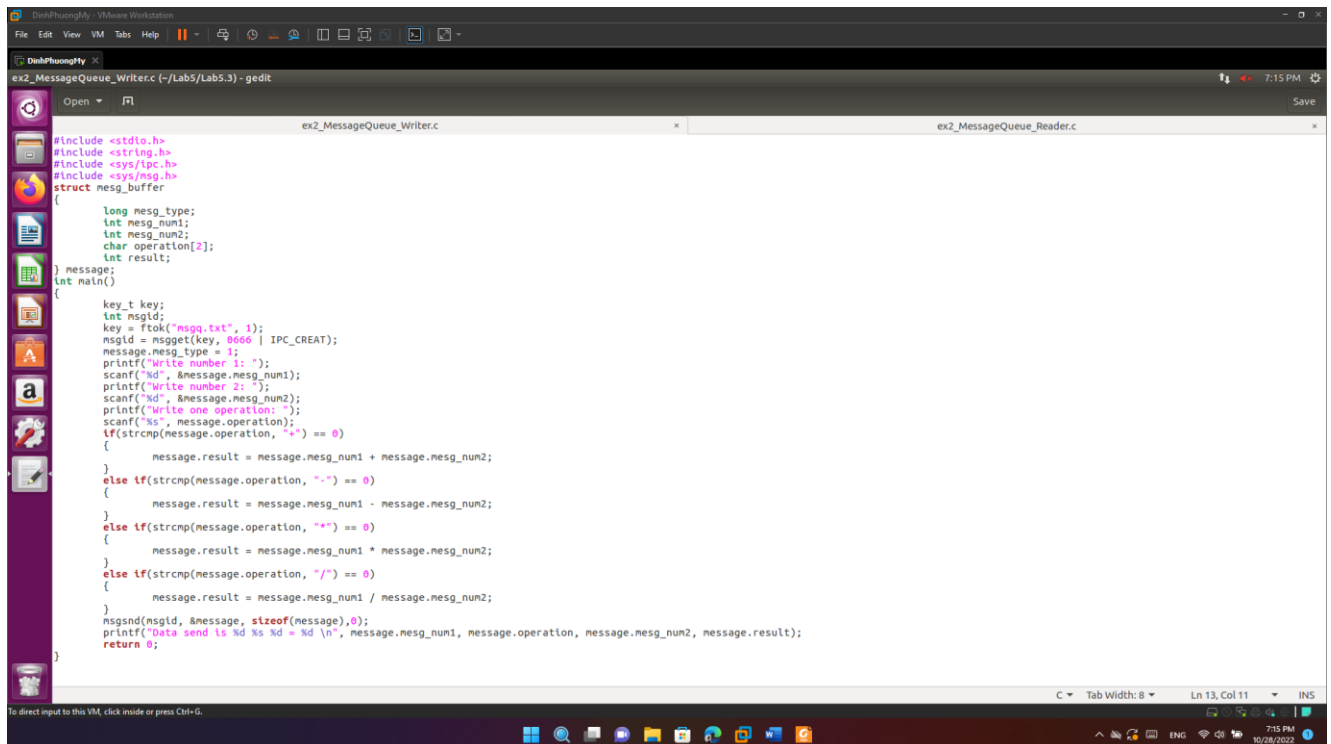


```
dinhphuongmy@ubuntu:~/Lab5/Lab5.3
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dinhphuongmy@ubuntu:~/Lab5/Lab5.3$ cd Lab5
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$ gcc ex2_ShareMemory.c -o battap3A.out
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$ ./battap3A.out 4 6 +
4 + 6 = 10
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$
```

Message Queue

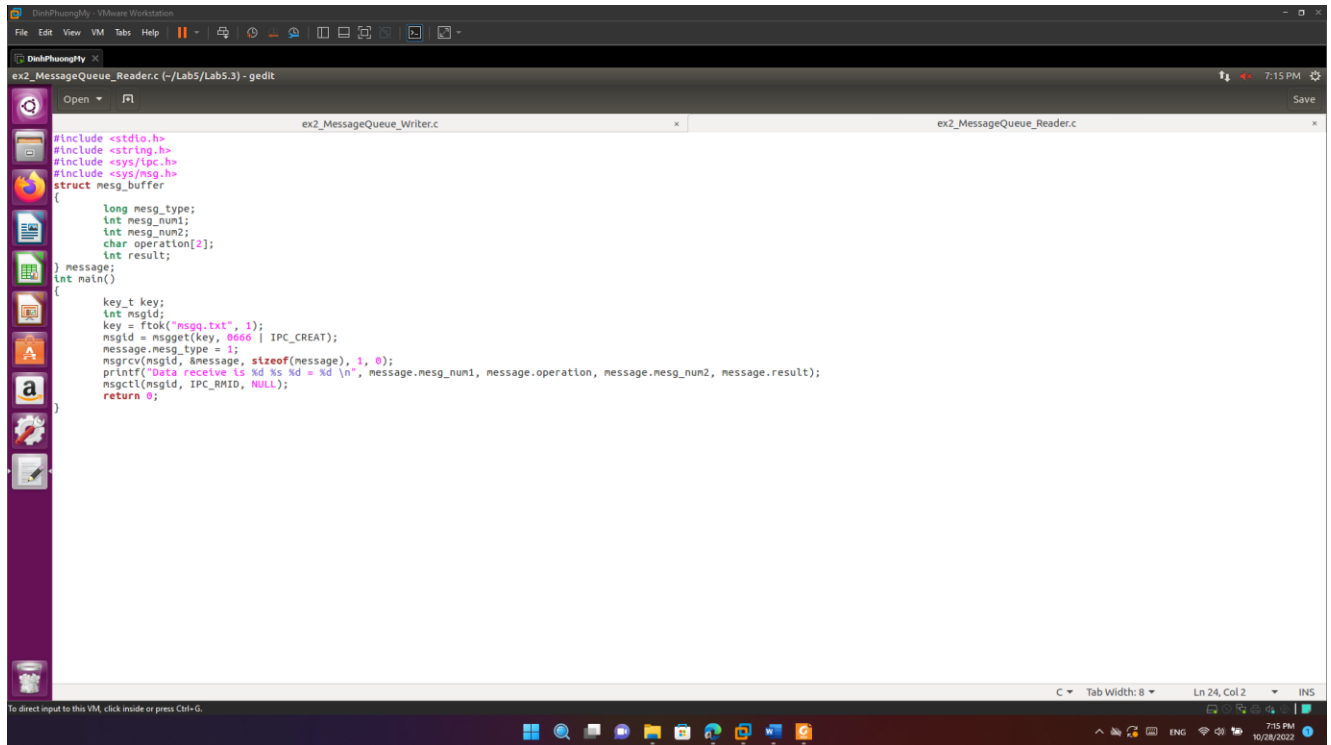
Writer



```
ex2_MessageQueue_Writer.c (-/Lab5/Lab5.3) - gedit
ex2_MessageQueue_Writer.c
ex2_MessageQueue_Reader.c

#include <stdio.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/msg.h>
struct msg_buffer
{
    long msg_type;
    int msg_num1;
    int msg_num2;
    char operation[2];
    int result;
} message;
int main()
{
    key_t key;
    int msgid;
    key = ftok("msgq.txt", 1);
    msgid = msgget(key, 0666 | IPC_CREAT);
    message.msg_type = 1;
    printf("Write number 1: ");
    scanf("%d", &message.msg_num1);
    printf("Write number 2: ");
    scanf("%d", &message.msg_num2);
    printf("Write one operation: ");
    scanf("%s", message.operation);
    if(strncmp(message.operation, "+") == 0)
    {
        message.result = message.msg_num1 + message.msg_num2;
    }
    else if(strncmp(message.operation, "-") == 0)
    {
        message.result = message.msg_num1 - message.msg_num2;
    }
    else if(strncmp(message.operation, "*") == 0)
    {
        message.result = message.msg_num1 * message.msg_num2;
    }
    else if(strncmp(message.operation, "/") == 0)
    {
        message.result = message.msg_num1 / message.msg_num2;
    }
    msgsnd(msgid, &message, sizeof(message), 0);
    printf("Data send is %d %s %d = %d\n", message.msg_num1, message.operation, message.msg_num2, message.result);
    return 0;
}
```

Reader:



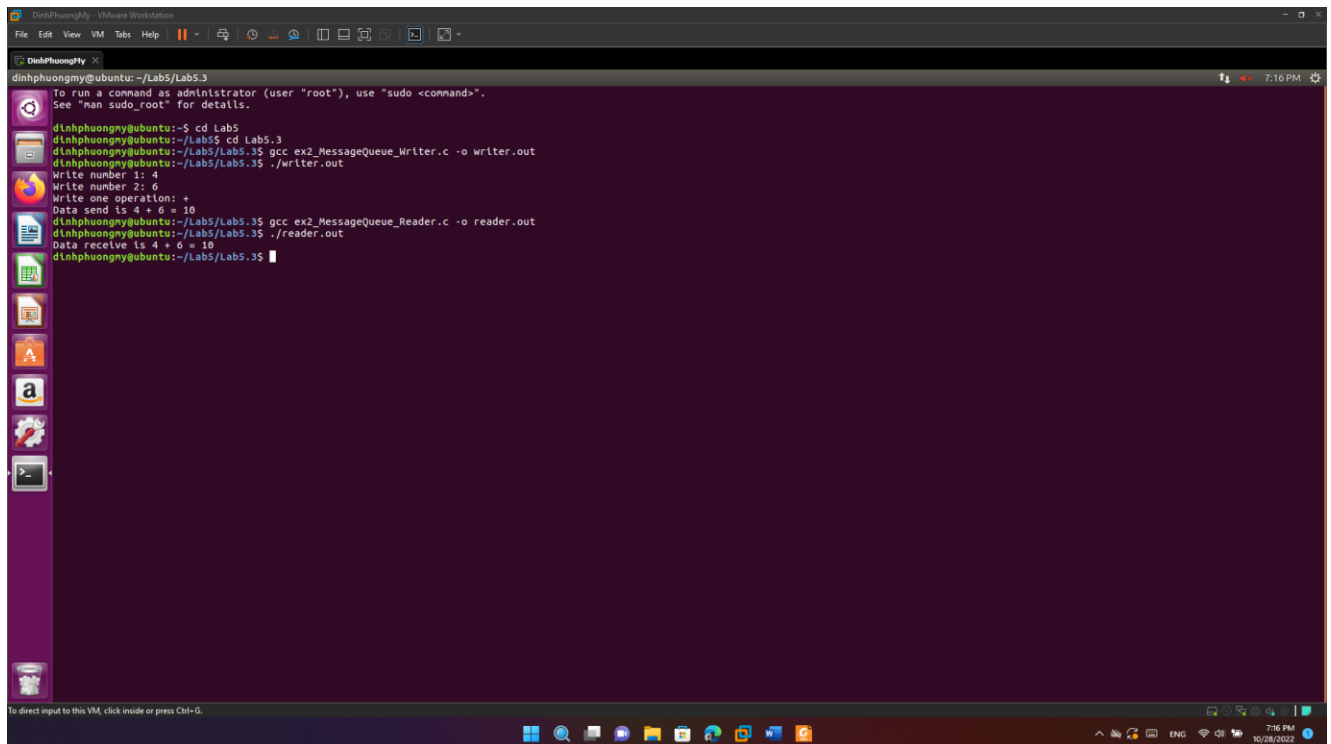
The screenshot shows a Virtual Machine window titled "DinhPhuongMy - VMware Workstation". Inside, a gedit text editor is open with the file "ex2_MessageQueue_Reader.c" located at "~/.Lab5/Lab5.3". The code is as follows:

```
#include <stdio.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/msg.h>
struct msg_buffer
{
    long msg_type;
    int msg_num1;
    int msg_num2;
    char operation[2];
    int result;
} message;

int main()
{
    key_t key;
    int msgid;
    key = ftok("msgq.txt", 1);
    msgid = msgget(key, 0660 | IPC_CREAT);
    message.msg_type = 1;
    msgrcv(msgid, &message, sizeof(message), 1, 0);
    printf("Data receive is %d %s %d = %d\n", message.msg_num1, message.operation, message.msg_num2, message.result);
    msgctl(msgid, IPC_RMID, NULL);
    return 0;
}
```

The status bar at the bottom of the editor indicates "Ln 24, Col 2". The VMware Workstation interface includes a menu bar (File, Edit, View, VM, Tabs, Help), a toolbar, and a sidebar with various icons. The bottom of the window shows the Windows taskbar with the time 7:15 PM and date 10/28/2022.

Run:



The screenshot shows the same Virtual Machine window, but now displaying a terminal window. The terminal output is as follows:

```
dinhphuongmy@ubuntu: ~/Lab5/Lab5.3
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dinhphuongmy@ubuntu:~/Lab5$ cd Lab5
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$ gcc ex2_MessageQueue_Writer.c -o writer.out
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$ ./writer.out
Write number 1: 4
Write number 2: 6
Write one operation: +
Data send is 4 + 6 = 10
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$ gcc ex2_MessageQueue_Reader.c -o reader.out
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$ ./reader.out
Data receive is 4 + 6 = 10
dinhphuongmy@ubuntu:~/Lab5/Lab5.3$
```

The terminal window is titled "DinhPhuongMy" and shows the user "dinhphuongmy" at the "ubuntu" host. The status bar at the bottom of the terminal indicates "Ln 24, Col 2". The VMware Workstation interface is the same as in the previous screenshot, with the time now 7:16 PM and date 10/28/2022.