**TASK 11**

Inter-Process Communication (IPC) Using Shared Memory

Implement an IPC mechanism using shared memory in C:

Create a shared memory segment and attach it to the process's memory space.

Write a string message to the shared memory in one process.

Read and display the message from the shared memory in another process.

**READER CODE**

#include <stdio.h>

#include <stdlib.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#define SHM\_SIZE 1024

int main() {

    key\_t key = ftok("shmfile", 65); // same key as writer

    int shmid = shmget(key, SHM\_SIZE, 0666); // get shared memory

    if (shmid == -1) {

        perror("shmget failed");

        return 1;

    }

    char \*str = (char \*) shmat(shmid, (void \*)0, 0); // attach

    if (str == (char \*)-1) {

        perror("shmat failed");

        return 1;

    }

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

    shmdt(str); // detach

    shmctl(shmid, IPC\_RMID, NULL); // remove shared memory

    return 0;

}

**FILE**

****

**WRITER CODE**

#include <stdio.h>

#include <stdlib.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <string.h>

#define SHM\_SIZE 1024  // size of shared memory

int main() {

    key\_t key = ftok("shmfile", 65); // generate unique key

    int shmid = shmget(key, SHM\_SIZE, 0666 | IPC\_CREAT); // create shared memory

    if (shmid == -1) {

        perror("shmget failed");

        return 1;

    }

    char \*str = (char \*) shmat(shmid, (void \*)0, 0); // attach to memory

    if (str == (char \*)-1) {

        perror("shmat failed");

        return 1;

    }

    printf("Writing to shared memory...\n");

    strcpy(str, "Hello from Writer Process using Shared Memory!");

    printf("Done. Detaching and exiting.\n");

    shmdt(str); // detach

    return 0;

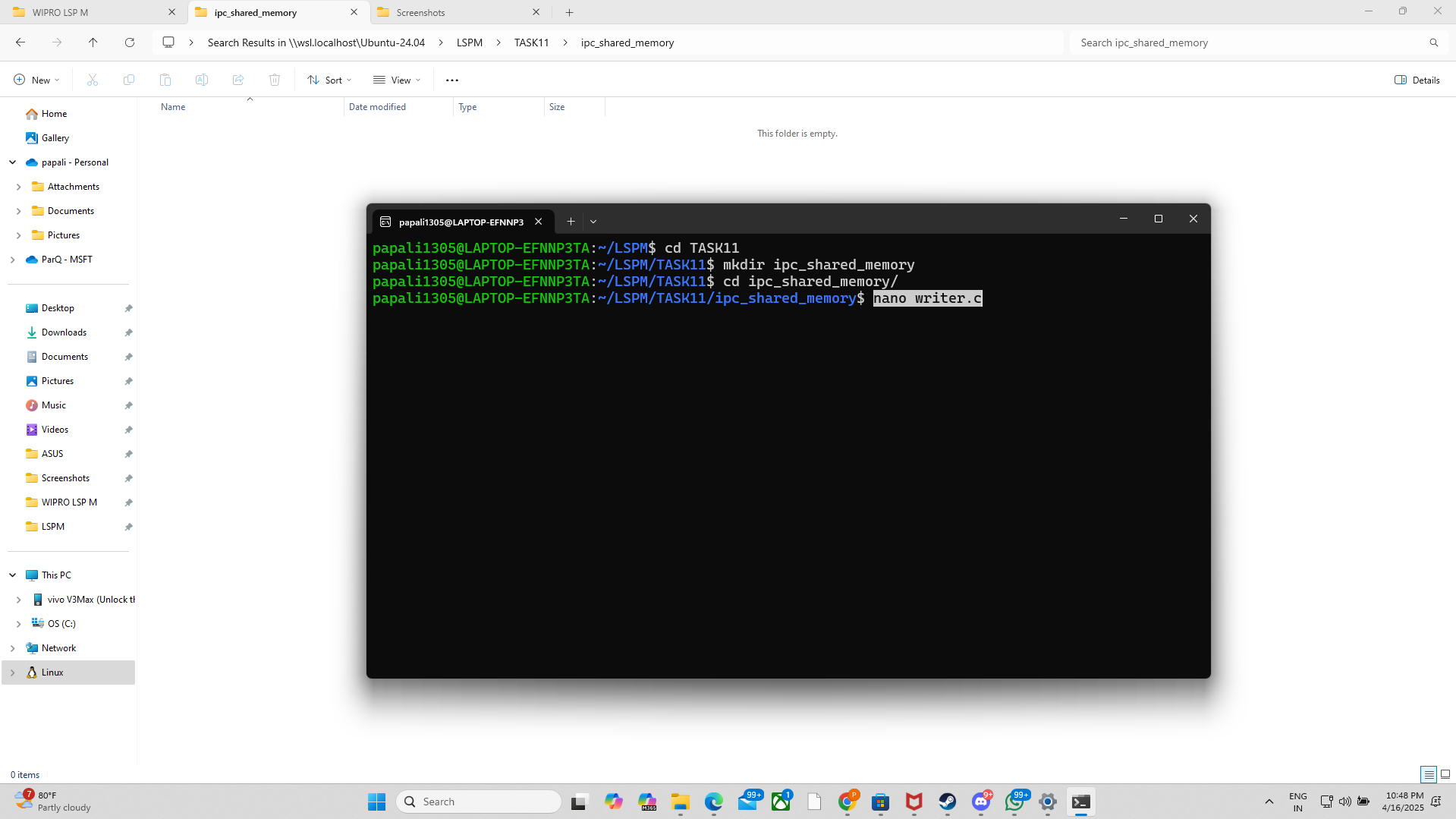
}

**FILE**

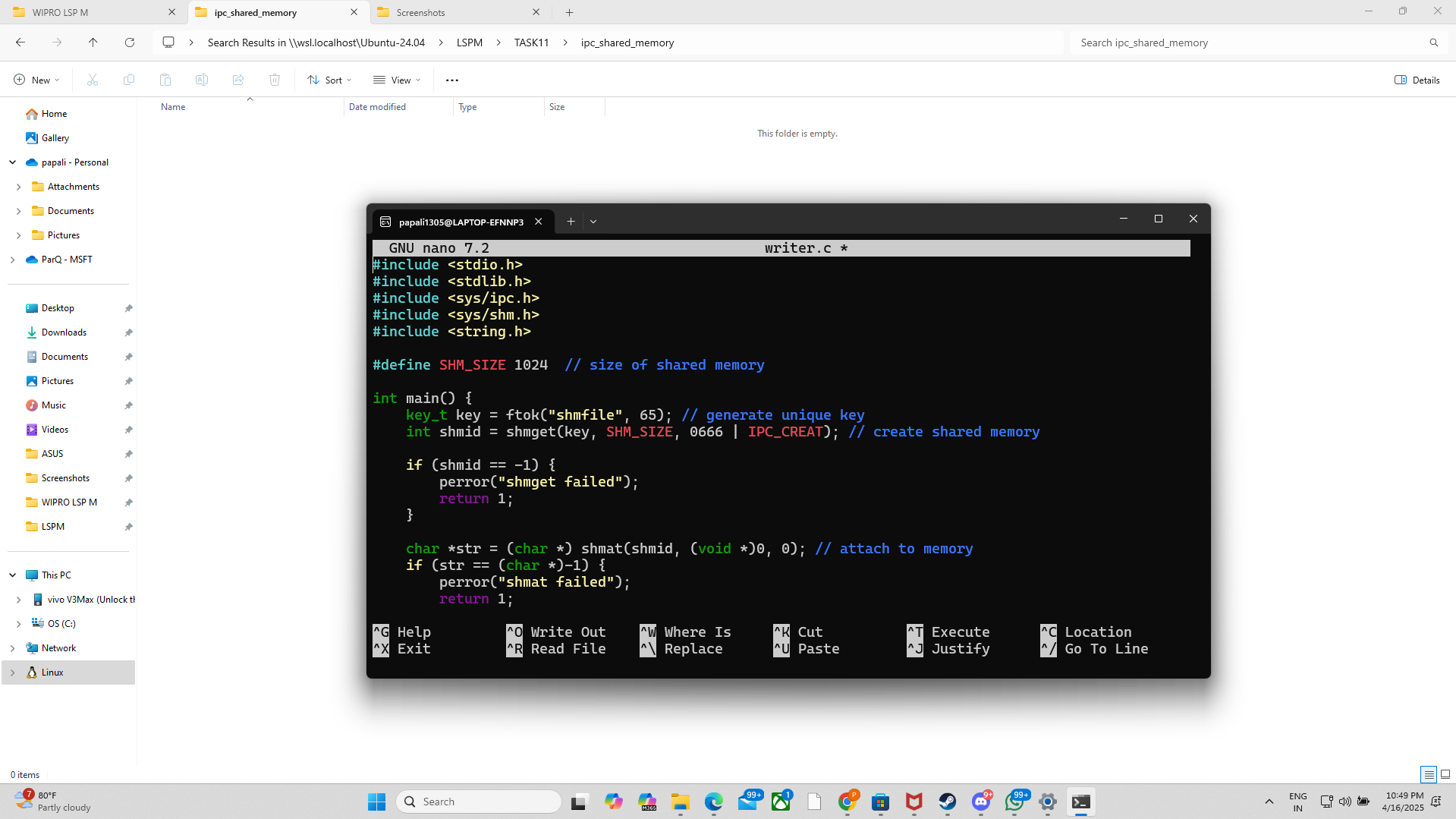
****

**OUTPUT AND SCREENSHOT**

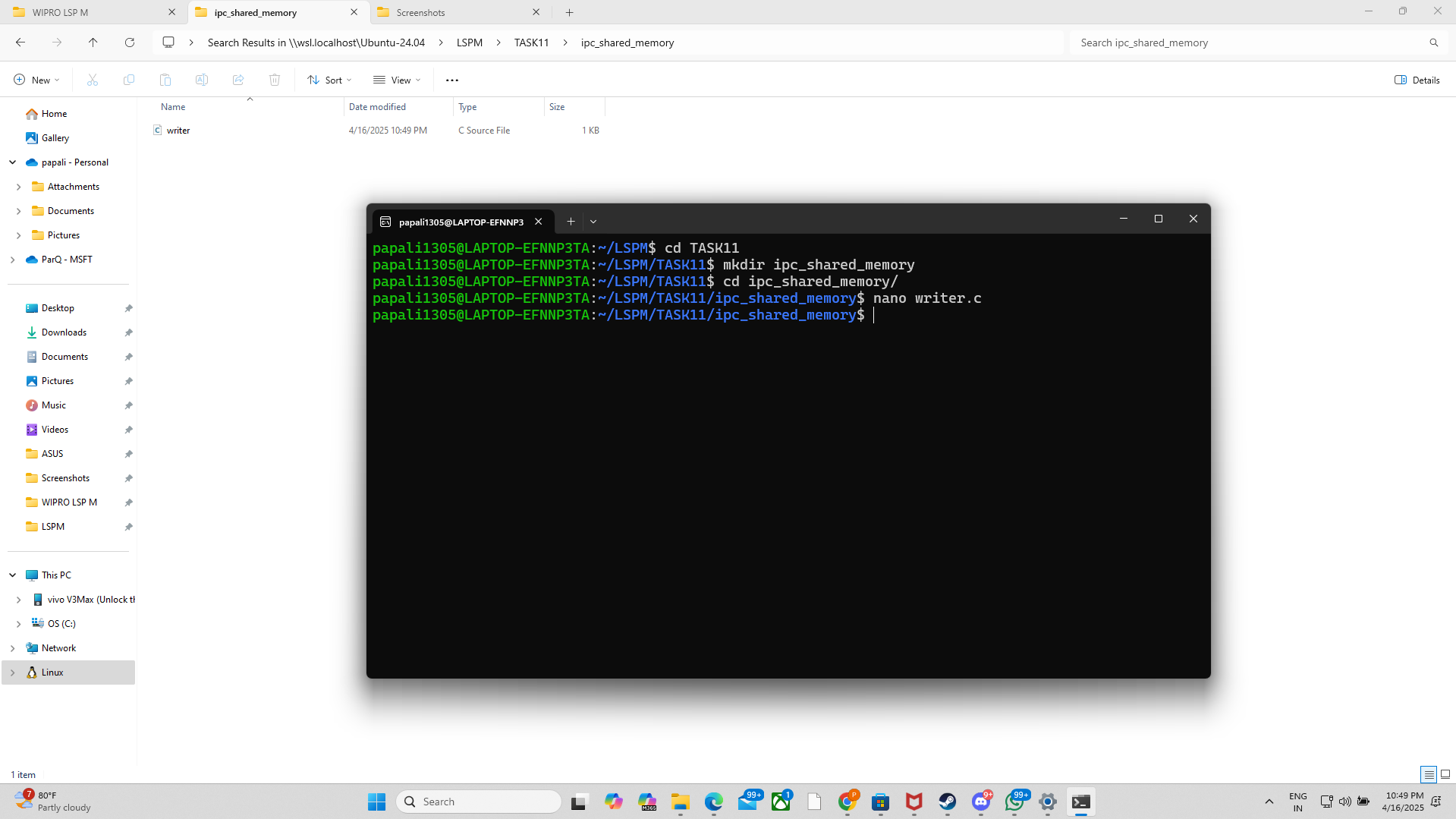
**01).**



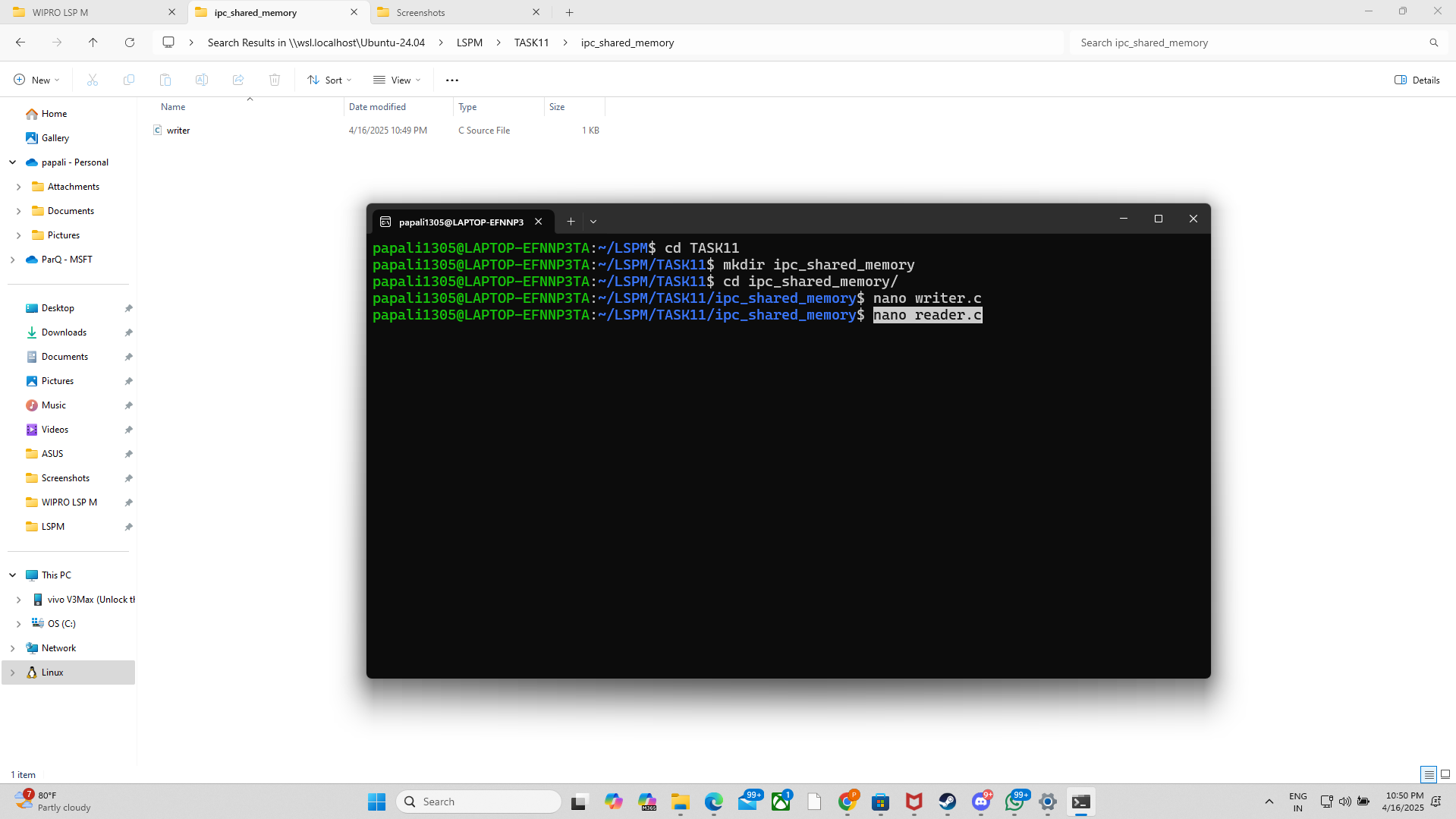
**02).**



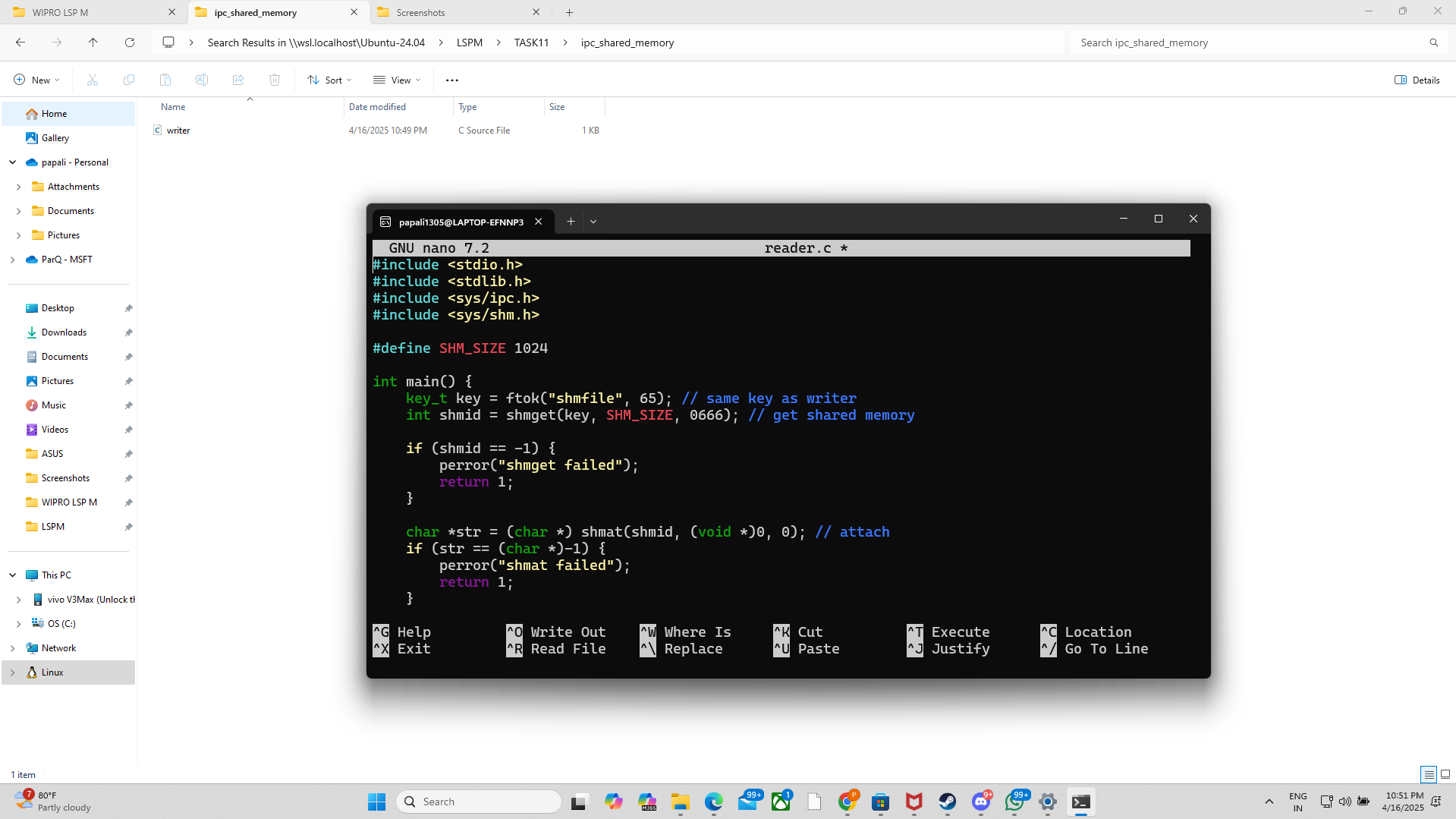
**03).**



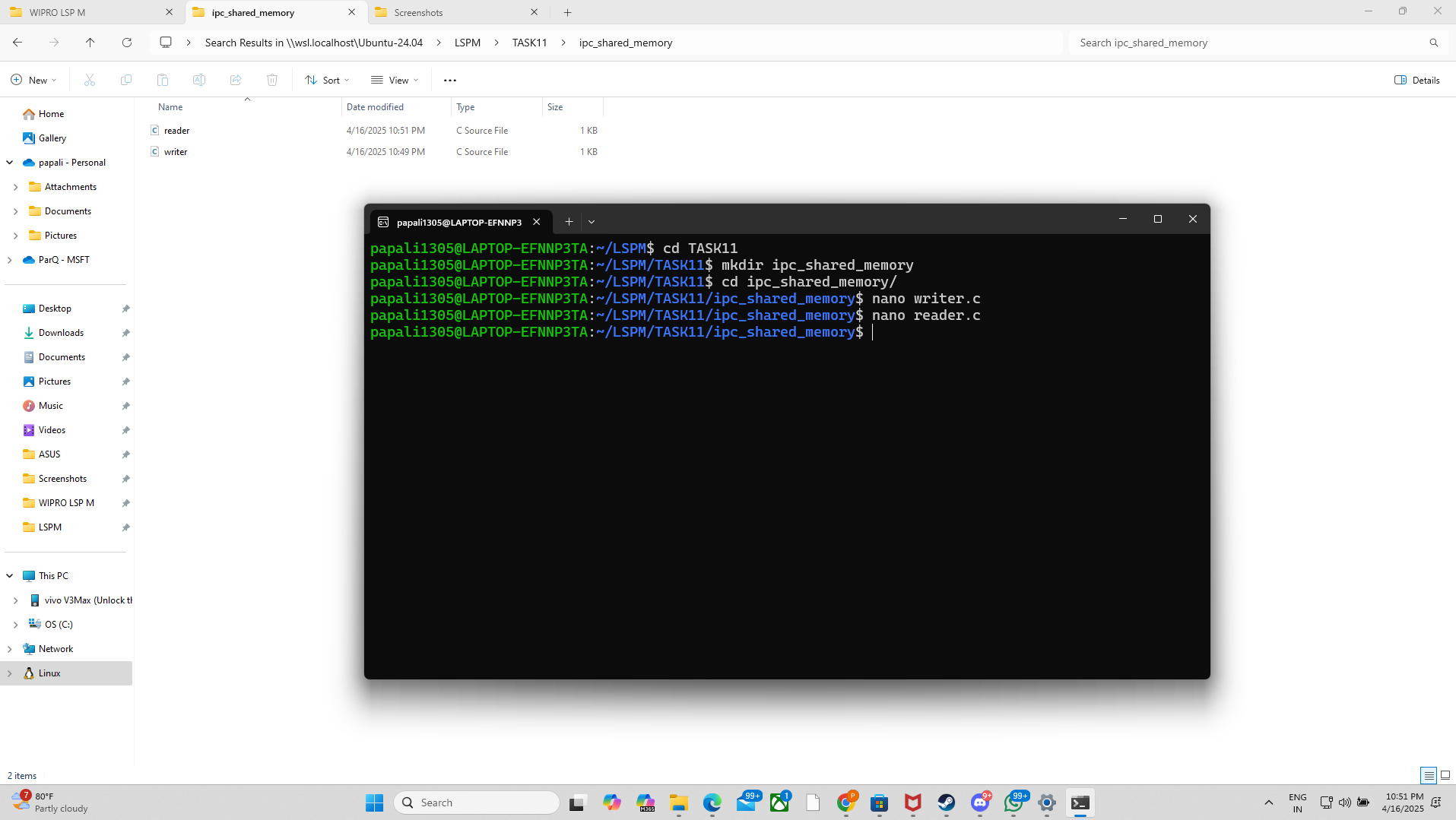
**04).**



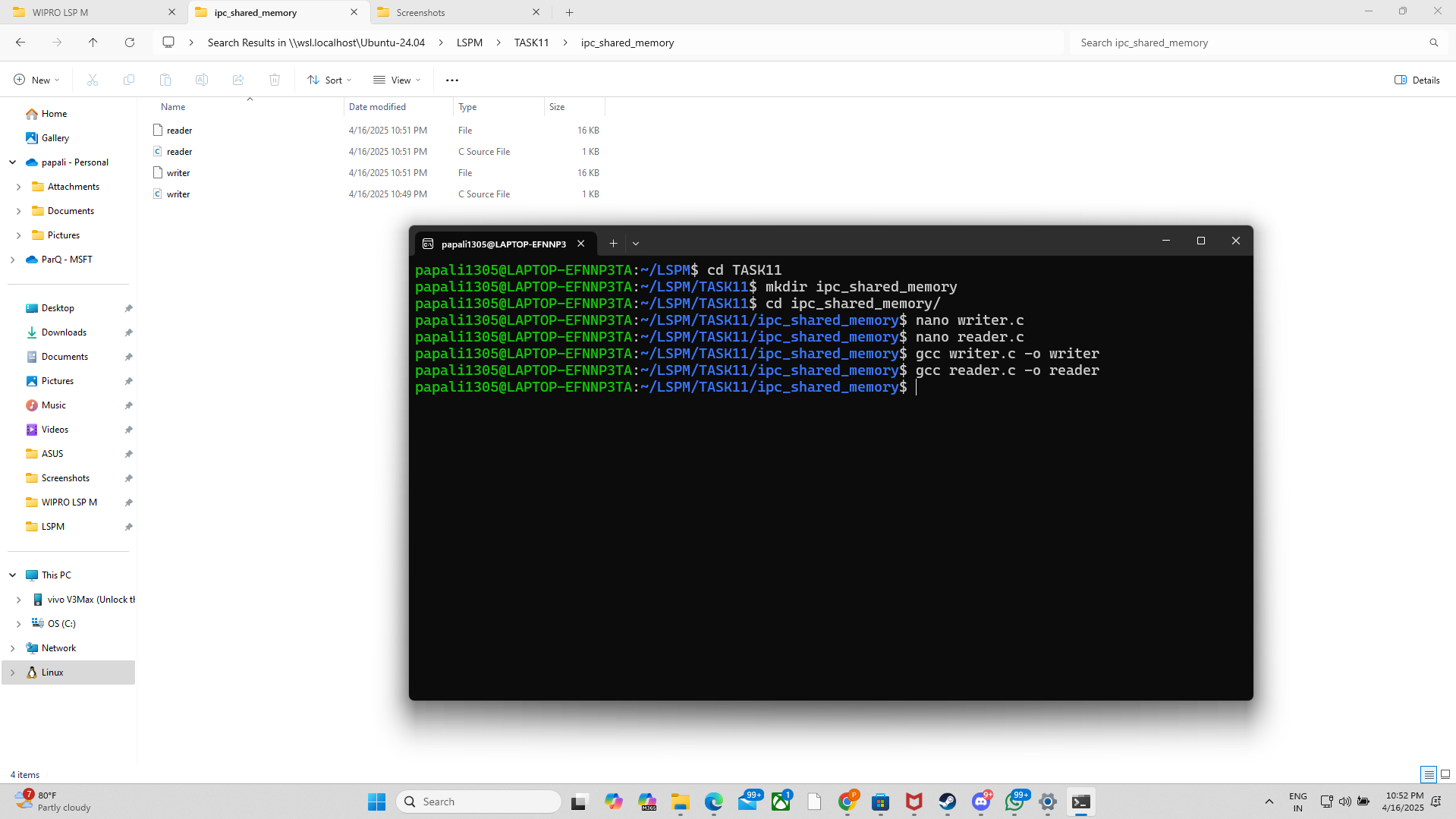
**05).**



**06).**



**07).**



**08).**

