

```

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

#include "memoria.h"

int main(int argc, char *argv[])
{
    key_t      Clave;
    int        ID;
    struct Memoria *ptr;

    if (argc != 5) {
        printf("Proceso1: Faltan argumentos, por favor escriba los datos a compartir\n");
        exit(1);
    }

    printf("Proceso1: Id es %d\n", (int)getpid());
    Clave = ftok(".", 'x');
    ID = shmget(Clave, sizeof(struct Memoria), IPC_CREAT | 0666);
    if (ID < 0) {
        printf("*** Error en shmget (Proceso1) ***\n");
        exit(1);
    }
    printf("Proceso1: Se utilizara memoria compartida para almacenar 4 valores enteros\n");

    ptr = (struct Memoria *) shmat(ID, NULL, 0);
    if ((int) ptr == -1) {
        printf("*** Proceso1: error en shmat ***\n");
        exit(1);
    }
    printf("Proceso1:Usando la memoria compartida...\n");

    ptr->estado = NOLISTO;
    ptr->datos[0] = atoi(argv[1]);
    ptr->datos[1] = atoi(argv[2]);
    ptr->datos[2] = atoi(argv[3]);
    ptr->datos[3] = atoi(argv[4]);
    printf("Proceso1: Los datos colocados en la memoria compartida son %d %d %d %d \n",
        ptr->datos[0], ptr->datos[1],
        ptr->datos[2], ptr->datos[3]);
    ptr->estado = LLENO;

    printf("Proceso1:Por favor ejecute al cliente en otra terminal\n");

    while (ptr->estado != RECIBIDO)
        sleep(1);

    printf("Proceso1: se a entregado la información...\n");
    shmdt((void *) ptr);
    shmctl(ID, IPC_RMID, NULL);
    printf("Proceso1: liberando memoria compartida.\n");
    printf("Proceso1: terminado\n");
    return 0;
}

```