

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

SISTEMAS OPERATIVOS I

PRACTICA 9: MEMORIA COMPARTIDA



Docente:

Prof. Marcos González Flores

Alumno:

Jesús Huerta Aguilar

Alex Abdiel Ruano Flores

Matricula:

202041509

202075025

NRC: 46152

Sección: 003

QUINTO SEMESTRE

Practica No.9

Nombre: Memoria compartida

Objetivo: Aprender a crear un segmento de memoria compartida, así como a utilizar y eliminar dicha memoria:

Desarrollo: Crear un segmento de memoria.

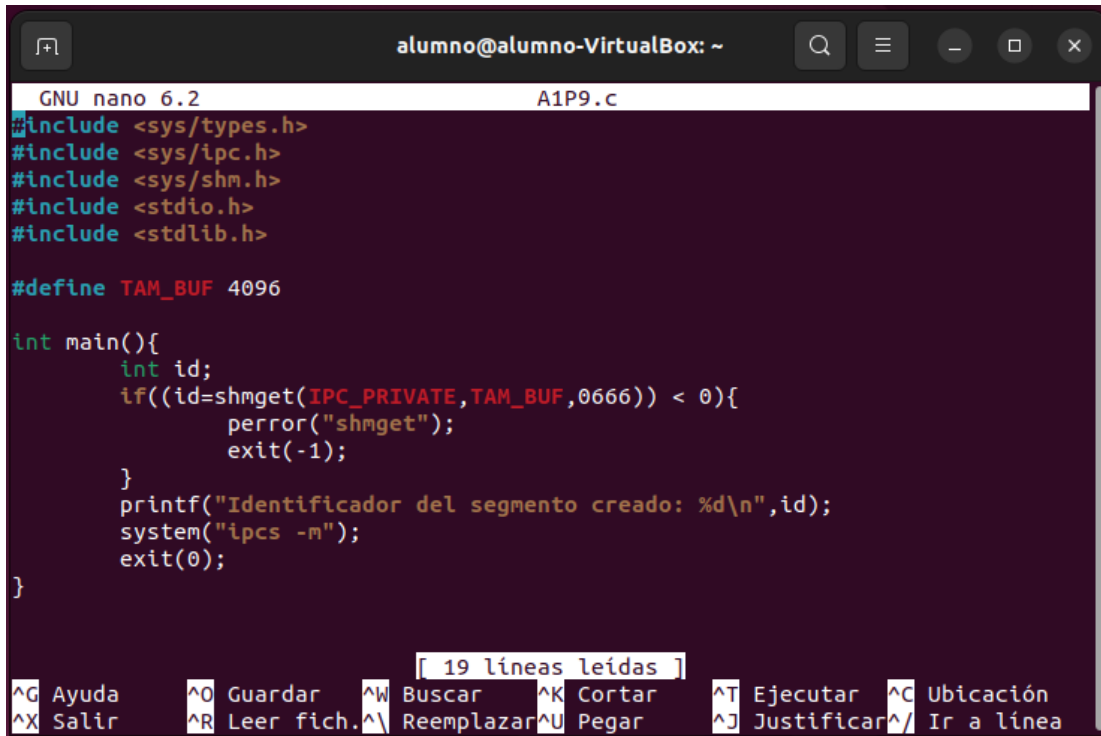
EJERCICIO 1

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

#define TAM_BUF 4096

int main(){
    int id;
    if((id=shmget(IPC_PRIVATE, TAM_BUF,0666)) < 0){
        perror("shmget");
        exit(-1);
    }
    printf("Identificador del segmento creado: %d\n",id);
    system("ipcs -m");
    exit(0);
}
```

TRANSCRITO + EJECUCIÓN:



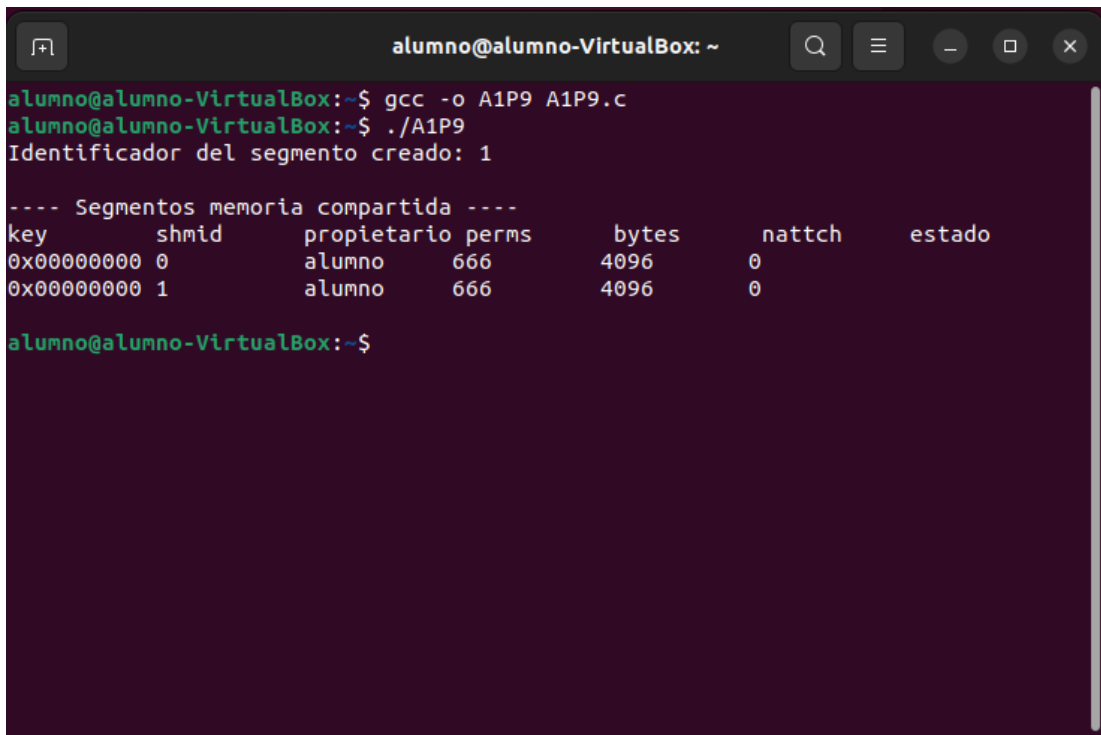
```
GNU nano 6.2 A1P9.c
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>

#define TAM_BUF 4096

int main(){
    int id;
    if((id=shmget(IPC_PRIVATE,TAM_BUF,0666)) < 0){
        perror("shmget");
        exit(-1);
    }
    printf("Identificador del segmento creado: %d\n",id);
    system("ipcs -m");
    exit(0);
}
```

[19 líneas leídas]

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^_ Ir a línea



```
alumno@alumno-VirtualBox: ~
alumno@alumno-VirtualBox:~$ gcc -o A1P9 A1P9.c
alumno@alumno-VirtualBox:~$ ./A1P9
Identificador del segmento creado: 1

---- Segmentos memoria compartida ----
key          shmid      propietario perms    bytes    nattch   estado
0x00000000  0          alumno      666      4096     0
0x00000000  1          alumno      666      4096     0

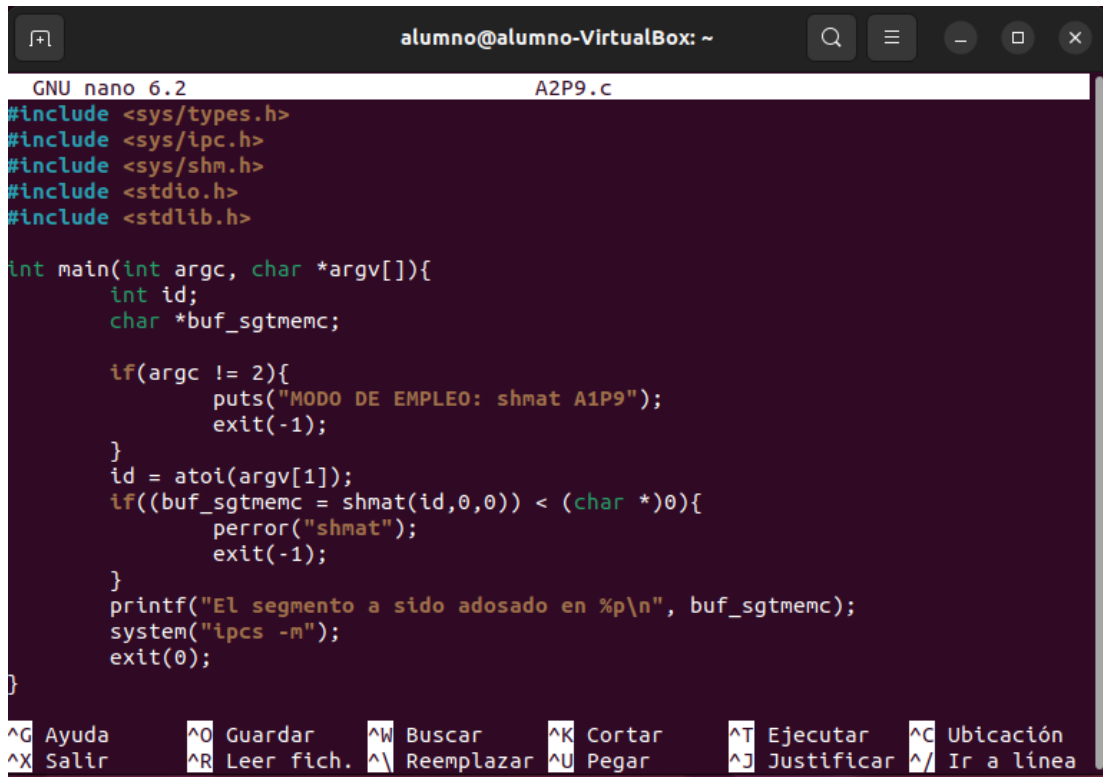
alumno@alumno-VirtualBox:~$
```

EJERCICIO 2

```
#include <sys/types.h>
#include <sys/tpc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]){
    int id;
    char *buf_sgtmemc;
    if(argc != 2){
        puts("MODO DE EMPLEO: shmat A1P9");
        exit(-1);
    }
    id = atoi (argv[1]);
    if((buf_sgtmemc = shmat(id,0,0)) < (char *)0){
        perror("shmat");
        exit(-1);
    }
    printf("El segmento a sido adosado en %p\n", buf_sgtmemc);
    system("ipcs -m");
    exit(0);
}
```

TRANSCRITO + EJECUCIÓN:

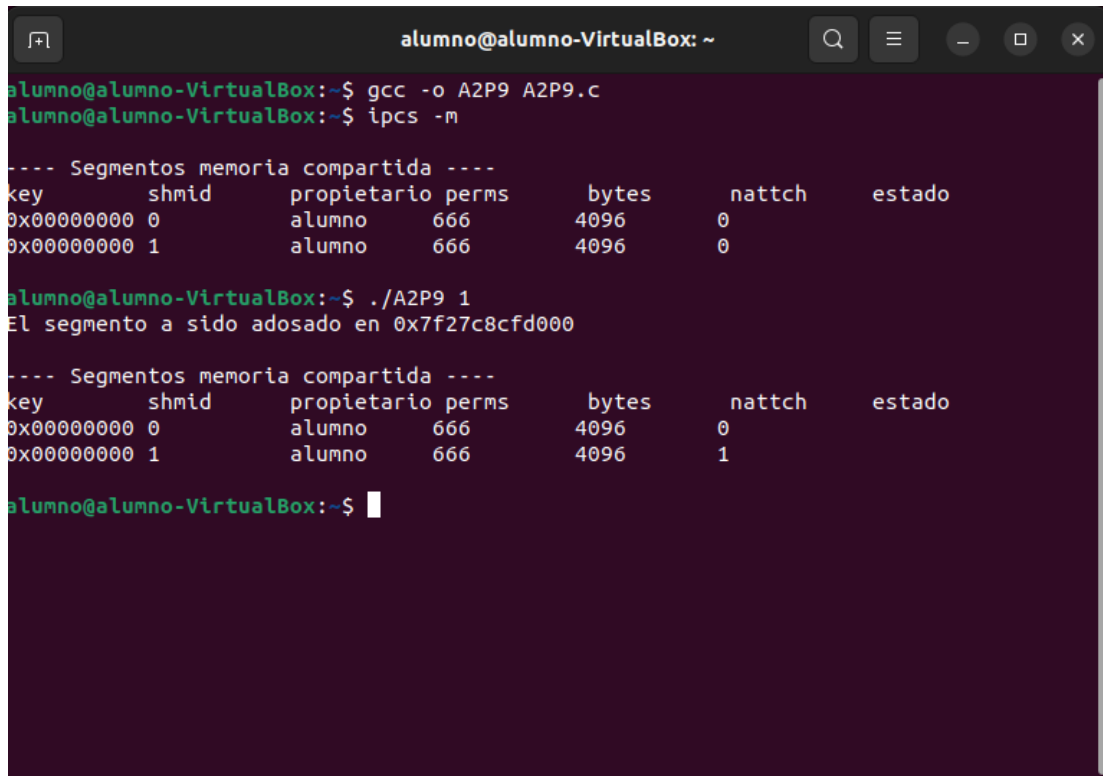


```
GNU nano 6.2 A2P9.c
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]){
    int id;
    char *buf_sgtmemc;

    if(argc != 2){
        puts("MODULO DE EMPLEO: shmat A1P9");
        exit(-1);
    }
    id = atoi(argv[1]);
    if((buf_sgtmemc = shmat(id,0,0)) < (char *)0){
        perror("shmat");
        exit(-1);
    }
    printf("El segmento a sido adosado en %p\n", buf_sgtmemc);
    system("ipcs -m");
    exit(0);
}
```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^_ Ir a línea



```
alumno@alumno-VirtualBox:~$ gcc -o A2P9 A2P9.c
alumno@alumno-VirtualBox:~$ ipcs -m

---- Segmentos memoria compartida ----
key      shmid      propietario perms      bytes      nattch      estado
0x00000000 0      alumno      666      4096      0
0x00000000 1      alumno      666      4096      0

alumno@alumno-VirtualBox:~$ ./A2P9 1
El segmento a sido adosado en 0x7f27c8cfd000

---- Segmentos memoria compartida ----
key      shmid      propietario perms      bytes      nattch      estado
0x00000000 0      alumno      666      4096      0
0x00000000 1      alumno      666      4096      1

alumno@alumno-VirtualBox:~$
```

EJERCICIO 3

```
#include <sys/types.h>
#include <sys/tpc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

#define TAM_BUF 4096

int main(int argc, char *argv[]){
    int id;
    char *buf_sgtmemc;
    int salida, i;

    if(argc != 2){
        puts("MODULO DE EMPLEO: abrir_sgtmemc <identificador>");
        exit(-1);
    }
    id = atoi (argv[1]);
    if((buf_sgtmemc = shmat(id,0,0)) < (char *)0){
        perror("shmat");
        exit(-1);
    }
    if((buf_sgtmemc = malloc(sizeof(char)*TAM_BUF)) == NULL) {
        perror("malloc");
        exit(-1);
    }
    for(i=0; i < TAM_BUF; i++)
        buf_sgtmemc[i] = {x127;
        salida = open("text.c",O_CREAT|O_WRONLY,0600);
        write(salida, buf_sgtmemc, TAM_BUF);
        close (salida);
        printf("Eliminando memoria compartida");
        shmctl(id, IPC_RMID, 0);
        exit(0);
    }
```

TRANSCRITO + EJECUCIÓN:

```
alumno@alumno-VirtualBox: ~
GNU nano 6.2 A3P9.c
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

#define TAM_BUF 4096

int main(int argc, char *argv[]){
    int id;
    char *buf_sgtmemc;
    int salida,i;

    if(argc != 2){
        puts("MODULO DE EMPLEO: abrir_sgtmemc <identificador>");
        exit(-1);
    }
    id = atoi(argv[1]);
    if((buf_sgtmemc = shmat(id,0,0)) < (char *)0){
        perror("shmat");
        exit(-1);
    }
    if((buf_sgtmemc = malloc(sizeof(char)*TAM_BUF)) == NULL){
        perror("malloc");
        exit(-1);
    }
}
```

```
alumno@alumno-VirtualBox: ~
GNU nano 6.2 A3P9.c
char *buf_sgtmemc;
int salida,i;

if(argc != 2){
    puts("MODULO DE EMPLEO: abrir_sgtmemc <identificador>");
    exit(-1);
}
id = atoi(argv[1]);
if((buf_sgtmemc = shmat(id,0,0)) < (char *)0){
    perror("shmat");
    exit(-1);
}
if((buf_sgtmemc = malloc(sizeof(char)*TAM_BUF)) == NULL){
    perror("malloc");
    exit(-1);
}

for(i=0;i < TAM_BUF;i++){
    buf_sgtmemc[i] = i%127;
    salida = open("text.c",O_CREAT|O_WRONLY,0600);
    write(salida,buf_sgtmemc,TAM_BUF);
    close(salida);
    printf("Eliminando memoria compartida");
    shmctl(id,IPC_RMID,0);
    exit(0);
}
```

```
alumno@alumno-VirtualBox: ~  
alumno@alumno-VirtualBox:~$ ./A1P9  
Identificador del segmento creado: 3  
  
---- Segmentos memoria compartida ----  
key      shmid      propietario perms      bytes      nattch      estado  
0x00000000 3          alumno      666        4096        0  
  
alumno@alumno-VirtualBox:~$ ./A2P9 3  
El segmento a sido adosado en 0x7f7222e6d000  
  
---- Segmentos memoria compartida ----  
key      shmid      propietario perms      bytes      nattch      estado  
0x00000000 3          alumno      666        4096        1  
  
alumno@alumno-VirtualBox:~$ ./A3P9 3  
Eliminando memoria compartidaalumno@alumno-VirtualBox:~$ ipcs -m  
  
---- Segmentos memoria compartida ----  
key      shmid      propietario perms      bytes      nattch      estado  
  
alumno@alumno-VirtualBox:~$
```


EJERCICIO 4

```
#include <sys/types.h>
#include <sys/tpc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>
#define LLAVE 234
#define MAX 10
int main() {
    int shmid, i;
    key_t key=LLAVE;
    char *vector;
    /*Creacion de un segmento de memoria compartida*/
    shmid = shmget(key, (MAX*sizeof(char)), IPC_CREAT | 0600);
    if(shmid == -1){
        printf("No se ha podido crear el segmento.");
        exit(-1);
    }
    else{
        /*Conexion del segmento al espacio de direcciones virtuales*/
        vector = shmat (shmid,0,0);
        if(vector == (char*)0){
            printf("Error en la concexion.");
            exit(-1);
        }
        /*Manipulacion del segmento de memoria compartida*/
        printf("Segmento mapeado (asociado)\n");
        for(i=0; i < MAX; i++)
            vector[i] = 'a';
        for(i=0; i < MAX; i++)
            printf("Valor almacenado %c\n", vector[i]);
        /*Desconexion del segmento al espacio de direcciones virtuales*/
        if(shmdt (vector) == -1){
            printf("Error en la desconexion.");
            exit(-1);
        }
        else
            printf("Exito en la desconexion\n");
        /*Borrado del segmento de memoria compartida*/
        if(shmctl(shmid,IPC_RMID,0) == -1){
            printf("Error en el borrado del segmento.");
            exit(-1);
        }
        else
            printf("Borrada memoria compartida\n");
    }
}
```



```
GNU nano 6.2 A4P9.c
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>

#define LLAVE 234
#define MAX 10

int main(){
    int shmid, i;
    key_t key=LLAVE;
    char *vector;

    /*Creacion de un segmento de memoria compartida*/
    shmid = shmget(key,(MAX*sizeof(char)), IPC_CREAT|0600);
    if(shmid == -1){
        printf("No se ha podido crear el segmento.");
        exit(-1);
    }
    else{
        /*Conexion del segmento al espacio de direcciones virtuales*/
        vector = shmat(shmid,0,0);
        if(vector == (char*)0){
            printf("Error en la concexion.");
            exit(-1);
        }

        printf("Error en la concexion.");
        exit(-1);
    }

    /*Manipulaxion del segmento de memoria compartida*/
    printf("Segmento mapeado (asociado)\n");
    for(i=0;i < MAX;i++)
        vector[i] = 'a';
    for(i=0;i < MAX;i++)
        printf("Valor almacenado %c\n", vector[i]);

    /*Desconexion del segmento al espacio de direcciones virtuales*/
    if(shmdt(vector) == -1){
        printf("Error en la desconexion.");
        exit(-1);
    }
    else
        printf("Exito en la desconexion\n");

    /*Borrado del segmento de memoria compartida*/
    if(shmctl(shmid,IPC_RMID,0) == -1){
        printf("Error en el borrado del segmento.");
        exit(-1);
    }
    else
        printf("Borrada memoria compartida\n");
}

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar ^/_ Ir a línea
```

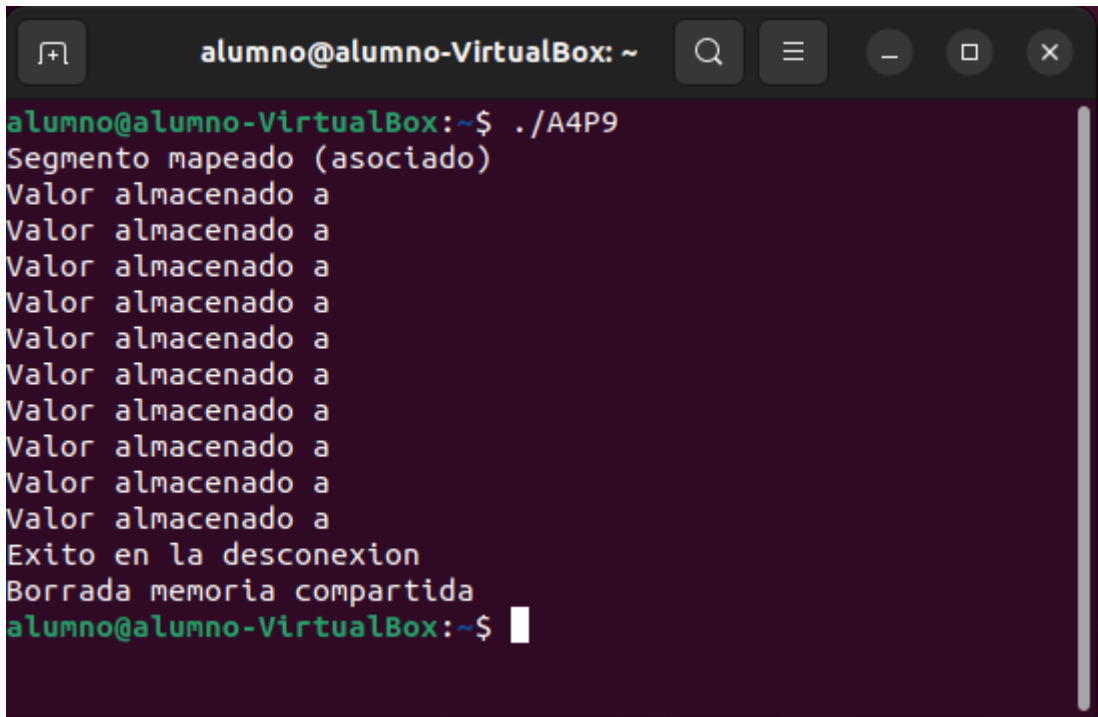
```
GNU nano 6.2 A4P9.c
        printf("Error en la concexion.");
        exit(-1);
    }

    /*Manipulaxion del segmento de memoria compartida*/
    printf("Segmento mapeado (asociado)\n");
    for(i=0;i < MAX;i++)
        vector[i] = 'a';
    for(i=0;i < MAX;i++)
        printf("Valor almacenado %c\n", vector[i]);

    /*Desconexion del segmento al espacio de direcciones virtuales*/
    if(shmdt(vector) == -1){
        printf("Error en la desconexion.");
        exit(-1);
    }
    else
        printf("Exito en la desconexion\n");

    /*Borrado del segmento de memoria compartida*/
    if(shmctl(shmid,IPC_RMID,0) == -1){
        printf("Error en el borrado del segmento.");
        exit(-1);
    }
    else
        printf("Borrada memoria compartida\n");
}

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar ^/_ Ir a línea
```



```
alumno@alumno-VirtualBox: ~  
alumno@alumno-VirtualBox:~$ ./A4P9  
Segmento mapeado (asociado)  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Valor almacenado a  
Exitos en la desconexión  
Borrada memoria compartida  
alumno@alumno-VirtualBox:~$
```