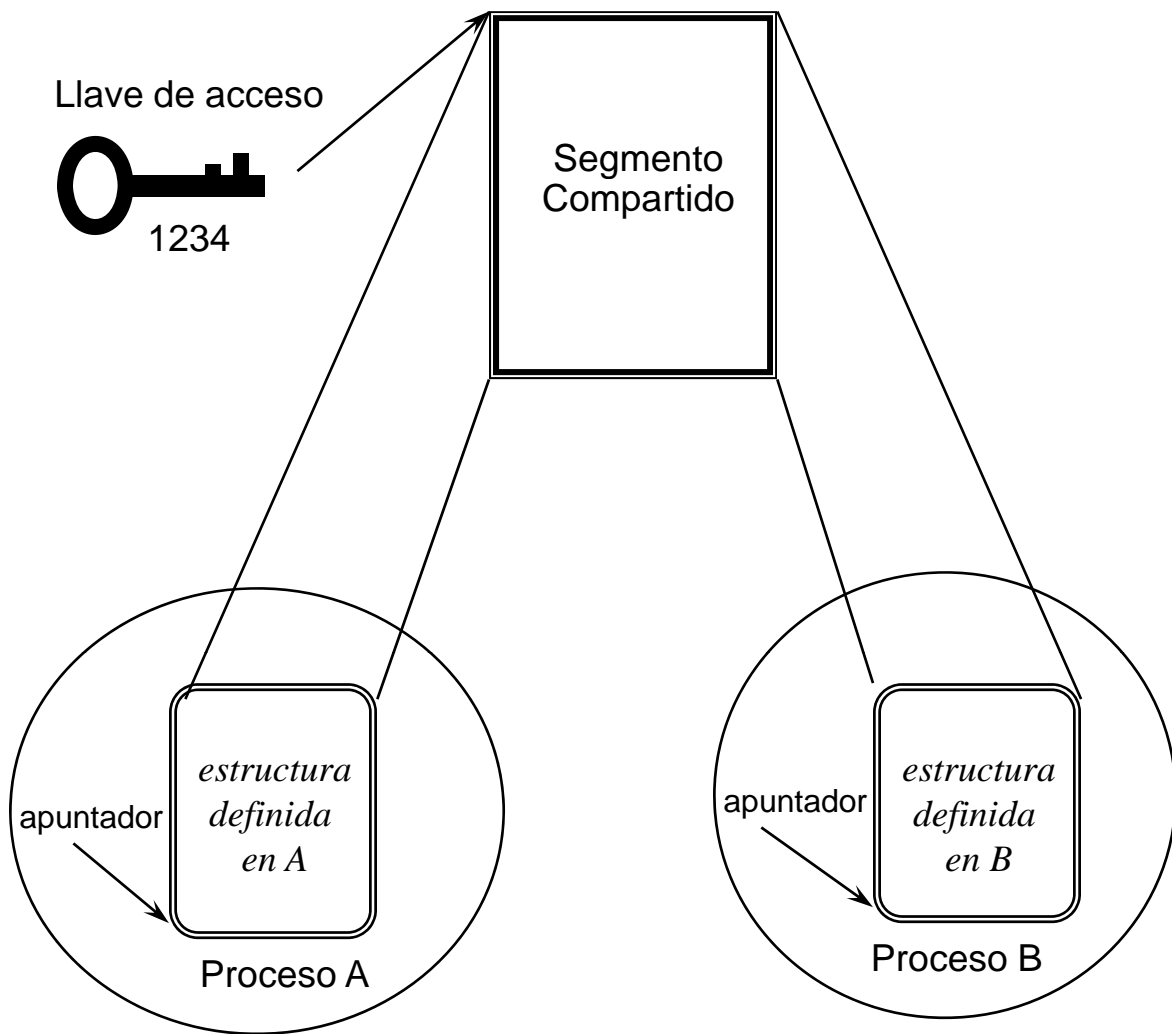


# Memoria Compartida



## Librerías y estructuras

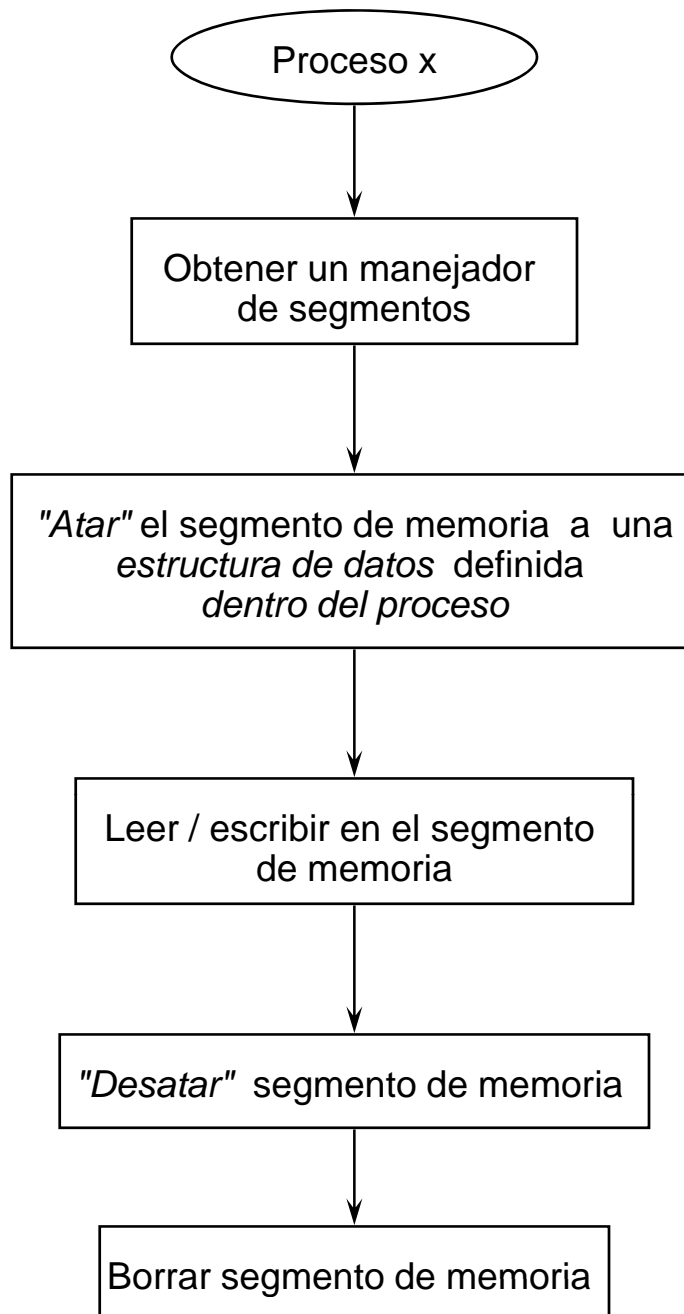
librerías:

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

Kernel mantiene la información siguiente:

```
struct shimd_ds {
    struct ipc_perm  shm_perm;    /* estructura operación permiso */
    int              shm_segz;    /* tamaño segmento */
    struct XXX      shm_YYY;     /* depende de la info */
    ushort          shm_lpid;    /* pid de la última operación */
    ushort          shm_cpid;    /* creador pid */
    ushort          shm_nattch;  /* # del atado actual */
    ushort          shm_cnattch; /* # del atado in core */
    time_t          shm_atime;   /* tiempo ultimo atado */
    time_t          shm_dtime;   /* tiempo ultimo desatado */
    time_t          shm_ctime;   /* tiempo ultimo cambio */
}
```

## Pasos para acceder una memoria compartida



## shmget(key, size, flag)

- Creación de un segmento.
- Obtención manejador (handler) de un segmento

id = shmget(key, size, flag)

### int key

llave numerica de identificación del segmento

### int size

tamaño del segmento en bytes

### int flag

bandera para los derechos del segmento, 0 si el segmento ya esta creado.

Numérico	Simbólico	Descripción
0400	SHM_R	Read by owner
0200	SHM_W	Write by owner
0040	SHM_R >> 3	Read by group
0020	SHM_W >> 3	Write by group
0004	SHM_W >> 6	Read by world
0002	SHM_W >> 6	Write by world
	IPC_CREAT	Special IPC flag
	IPC_EXCL	Special IPC flag

### int id

manejador del segmento

## shmat(id, addr, flag)

- Ata un segmento existente en el espacio de direcciones del proceso.

```
ptr = shmat(id, addr, flag)
```

### int id

manejador del proceso, (obtenido a partir de shmget())

### int addr

especificación de una dirección de mapeo.  
generalmente cero, (el sistema se ocupa)

### int flag

bandera para manejo de la dirección de mapeo.  
generalmente 0

### ptr\_type \*ptr

apuntador del tipo de información almacenada en el segmento de memoria.

### Ejemplo

```
struct info{  
    char *nombre;    int    edad;  
};  
struct info *ptr;  
ptr = (struct info *) shmat(id, 0, 0)
```

## La llamada `shmdt()`

### *`shmdt(addr)`*

- Desatar un segmento cuando el proceso terminó de utilizarlo.
- No borra el segmento de memoria.

**`int *addr`**

Dirección del segmento

### *Ejemplo*

```
#include algo.h
int id;
struct info *ctrl;

main
{
    :
    id = shmget(KEY, SEGSIZE, IPC_CREAT | 0666);
    ctrl = (struct info*) shmat(id, 0, 0);
        < código uso memoria compartida >
    shmdt(ctrl);
}
```

## La llamada `shmctl()`

Operaciones de control de la memoria compartida

Sintáxis:

```
int shmctl(shmid, cmd, buf)
```

**int shmid**

identificador, (manejador), del segmento de memoria compartida

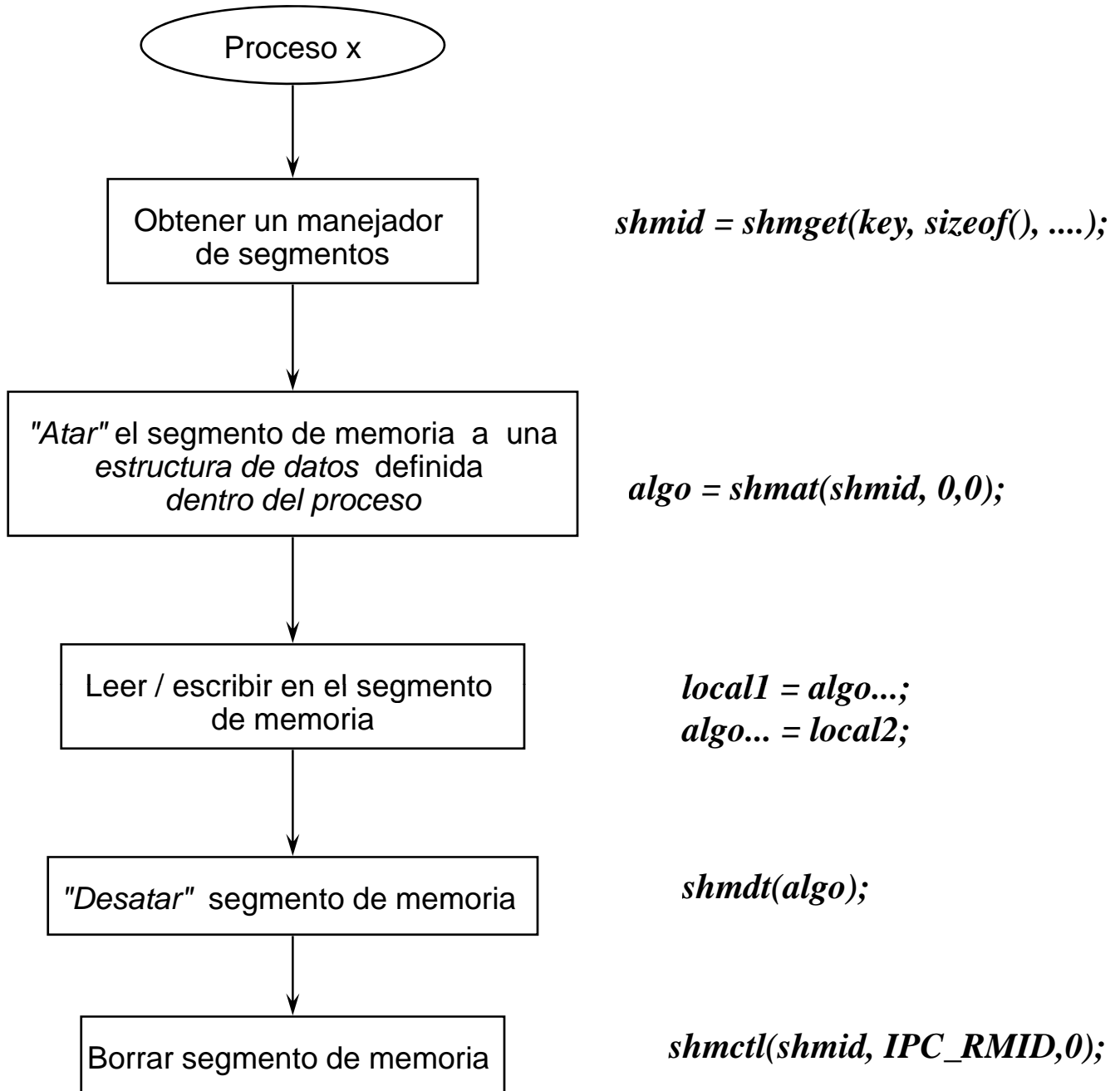
**int cmd**

<i>IPC_STAT</i>	asigna cada uno de los valores de los campos de la estructura de datos asociada con <i>shmid</i> , en la estructura apuntada por <i>buf</i>
<i>IPC_SET</i>	asigna el valor de los campos: <i>shm_perm.uid shm_perm.gid shm_perm.mode</i> en la estructura apuntada por <i>buf</i>
<i>IPC_RMID</i>	borra el identificador del segmento de memoria del sistema especificado por <i>shmid</i> , y destruye el segmento de memoria y la estructura de datos asociada a él
<i>SHM_LOCK</i>	bloquea el segmento de memoria especificado por el identificador <i>shmid</i>
<i>SHM_UNLOCK</i>	desbloquea el segmento de memoria especificado por el identificador <i>shmid</i>

**struct shmctl\_ds \*buf**

estructura en la que se almacena información del estatus del segmento de memoria, (no es usada por todas las opciones).

## Acceso memoria compartida





## Ejemplo memoria compartida

*salida en pantalla  
proceso pline*

```
$ pline  
aaaaaaaaaa  
aaaaaaaaaa  
bb  
bb
```

*salida en pantalla  
proceso cline*

```
$ cline b 2  
$
```

pline

cline

<i>char c</i> 'a'
<i>int n</i> 10

*segmento de memoria  
compartida*

## Contenido "*line.h*"

- Los dos programas deben conocer la llave del segmento y la estructura de datos.
- Declaración en el archivo *line.h* (header file)

```
/* estructura de los datos almacenados en el segmento */
```

```
struct info {  
    char c;  
    int  length;  
};
```

```
/* definicion de la llave y del tamaño de la estructura */
```

```
#define KEY          ((key_t)(1243))  
#define SEGSIZE     sizeof(struct info)
```

## Código pline.c

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include "line.h"
main()
{
    int i, id;
    struct info *ctrl;
    struct shmid_ds shmbuf;
    id = shmget( KEY, SEGSIZE, IPC_CREAT | 0666);
    if (id < 0) {
        printf(" Error en el shmget \n ");
        exit(0);
    }
    ctrl = (struct info *) shmat(id, 0, 0);
    if (ctrl < (struct info *) (0)) {
        printf(" Error pline: error en el shmat \n ");
        exit(2);
    }
}
```

## Código pline.c

```
/* escritura de los valores iniciales */

ctrl->c = 'a';
ctrl->length = 10;
while (ctrl->length > 0) {
    for (i=0; i< ctrl->length; i++)
        putchar(ctrl->c);
    putchar("\n");
    sleep(4);
}
exit(0);
}
```

## Código cline.c

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

main( argc, argv)
    int argc; char *argv[];
{
    int i, id;
    struct info *ctrl;
    struct shmid_ds shmbuf;
    if (argc != 3) {
        printf(" Error de uso faltan parámetros \n ");
        exit(3);
    }
    id = shmget(KEY, SEGSIZE, 0);
    if (id < 0) {
        printf(" Error en el cline: falló el shmget");
        exit(1);
    }
}
```

## Código cline.c

```
ctrl = (struct info *) shmat(id,0,0);
if ( ctrl <= (struct info *) (0) ) {
    printf(" Error en cline: falló shmat \n ");
    exit(2);
}

/* copiado de la línea de comando a la memoria
común */

ctrl->c = argv[1][0];
ctrl->length = atoi(argv[2]);
exit(0);
}
```

## Detalles sobre shmdt() y shmctl()

```
rogomez@armagnac:32> pline
```

```
aaaaaaaaaa
```

```
aaaaaaaaaa
```

```
aaaaaaaaaa
```

```
XX
```

```
XX
```

```
XX
```

```
--- Se desato el segmento de memoria, (shmdt()) ---
```

```
Segmentation fault (core dumped)
```

```
rogomez@armagnac:33> \rm core
```

```
rogomez@armagnac:34> pline
```

```
aaaaaaaaaa
```

```
aaaaaaaaaa
```

```
aaaaaaaaaa
```

```
aaaaaaaaaa
```

```
ZZZ
```

```
ZZZ
```

```
ZZZ
```

```
--- Se borro el segmento de memoria (shmctl()) ---
```

```
ZZZ
```

```
ZZZ
```

```
ZZZ
```

```
^C
```

```
rogomez@armagnac:35>
```