

Reporte de la práctica 6

Hernández Tapia Luis Enrique

Sistemas Operativos

Grupo 2CM7

Profesor: Montes Casiano Hermes Francisco

13 de abril de 2018

1. Introducción

En UNIX ®System V, se presenta un mecanismo que permite a dos o más procesos tomar un apuntador a una dirección de memoria de la cual se han reservado *n bytes* contiguos, acceden a la memoria para leer o escribir, lo cual permite la comunicación entre ellos, cabe resaltar que puede haber condiciones de competencia.

2. Desarrollo

Se nos pide escribir el siguiente programa 6-1 y guardarlo como programa6-1.c, después ejecutar en dos terminales para ver la interacción entre si.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/types.h>
4  #include <sys/ipc.h>
5  #include <sys/shm.h>
6
7  int leer_car(){
8      char letra;
9      char almacen[80];
10
11      scanf("%s",almacen);
12      sscanf(almacen, "%c", &letra);
13      return letra;
14 }
15
16 int main(int argc, char const *argv[])
17 {
18     int shmid, *variable;
19     key_t llave;
```

```

20     llave = ftok(argv[0], 'K');
21     if ((shmid = shmget(llave, sizeof(int), IPC_CREAT |
22         0600)) == -1)
23     {
24         perror("Error en shmget");
25         exit(-1);
26     }
27     /*Nos atamos a la memoria compartida*/
28     if ((variable = (int *) shmat(shmid, NULL, 0)) == (int
29         *) (-1))
30     {
31         perror("Fallo shmat");
32         exit(-1);
33     }
34     while(1)
35     {
36         printf("\nIntroduzca m para modificar el valor
37             de la variable, v para visualizarla y t para
38             terminar\n");
39         switch(leer_car())
40         {
41             case 't': /*Libera la memoria compartida
42                 */
43                 shmctl(shmid, IPC_RMID, 0);
44                 exit(0);
45             case 'v': /*Visualiza la variable*/
46                 printf("variable = %d\n", *
47                     variable);
48                 break;
49             case 'm': printf("Nuevo valor de la
50                 variable en memoria compartida\n");
51                 scanf("%d", variable);
52                 break;
53             default:
54                 printf("Se introdujo una letra
55                     incorrecta\n");
56                 break;
57         }
58     }
59     return 0;
60 }

```

3. Pruebas

Con el programa 6-1 se pide responder las siguientes preguntas.

Pregunta 6.1 - Escriba los valores de las llaves obtenidas en los casos siguientes:

Llave devuelta por `ftok(argv[0], 'K')`

Llave devuelta por `ftok(argv[0], ' ')`

Llave devuelta por `ftok(' ', ' ')`

Llave devuelta por `ftok("./chetito", 'K')`

R:

Llave devuelta por `ftok(argv[0], 'K')` : 1258381768

Llave devuelta por `ftok(argv[0], ' ')` : 536961480

Llave devuelta por `ftok(' ', ' ')` : -1

Llave devuelta por `ftok("./chetito", K)` : -1

Pregunta 6.2 - Modifique el programa para imprimir los valores mencionados tanto por el padre como el hijo, no olvide imprimir en forma hexadecimal (

R:

```
1 | printf("Zona de memoria compartida: %x\n", shmid); //  
2 | printf("Puntero de la memoria compartida %p\n", variable); //
```

Pregunta 6.3 - ¿Qué puede ocurrir si no se imprime la dirección de memoria en formato hexadecimal?

R: Imprime la variable en formato entero, es decir el tamaño.

Ejercicio 6.1 - Modifique el programa original, para simular una pila de 10 elementos de tipo carácter, un proceso debe funcionar como productor e insertar elementos en la pila, otro como consumidor, sacando elementos de la pila y un tercero para visualizar el contenido de la pila.

R:

```
1 | #include <stdio.h>  
2 | #include <stdlib.h>  
3 | #include <sys/types.h>  
4 | #include <sys/ipc.h>  
5 | #include <sys/shm.h>  
6 |  
7 | int main(int argc, char const *argv[])  
8 | {
```

```

9      int shmid, *variable;
10     key_t llave;
11     llave = ftok(argv[0], 'K');
12     if ((shmid = shmget(llave, sizeof(int), IPC_CREAT |
13         0600)) == -1)
14     {
15         perror("Error en shmget");
16         exit(-1);
17     }
18     /*Nos atamos a la memoria compartida*/
19     if ((variable = (int *) shmat(shmid, NULL, 0)) == (int
20         *) (-1))
21     {
22         perror("Fallo shmat");
23         exit(-1);
24     }
25     char pila[10] = {32,32,32,32,32,32,32,32,32,32};
26     *variable = 100;
27     char x;
28     int i = 9, j = 65;
29     printf("\nIntroduzca p para producir, c para consumir y
30         v para visualizar, t para terminar\n");
31     while(scanf("%c",&x) != EOF)
32     {
33         printf("\nIntroduzca p para producir, c para
34             consumir y v para visualizar, t para terminar
35             \n");
36         switch(x)
37         {
38             case 'p': /*Productor*/
39                 pila[i] = j; i--; j++;
40                 printf("Producido con exito!\n");
41                 ;
42                 break;
43             case 'c': /*Consumidor*/
44                 for (int k = 0; k < 10; ++k)
45                 {
46                     if(pila[k] != 32) {
47                         j = pila[k];
48                         pila[k] = 32;
49                         i = k;
50                         break;
51                     }
52                 }
53                 printf("Consumido con exito!\n");

```

```

49         ;
50         break;
51     case 'v': /*Visualizar*/
52         for (int i = 0; i < 10; i++){
53             printf("%c\n", pila[i]);}
54         printf("Visualizado con exito!\n");
55         break;
56     case 't': /*Terminar*/
57         shmctl(shmid, IPC_RMID, 0);
58         exit(0);
59         break;
60     default:
61         printf("Se introdujo una letra
62             incorrecta\n");
63         break;
64     }
65 }
66 return 0;
67 }

```

Pregunta 6.4 - Modifique el programa para tratar de utilizar sólo una línea *scanf* en la función *leer-car*, ¿qué sucede?

R: No detecta el carácter.

4. Conclusiones

La memoria compartida es muy útil para programas que subprocesos con *fork*, sin embargo no necesariamente es ese su límite, la memoria compartida igual puede ser usada por programas totalmente distintos como un medio para enviarse señales y compartir información en común.