Memoria Compartida y Semáforos System V

En el marco del problema del productor y del consumidor, proponemos la solución al problema de la sección crítica y al problema de la espera ocupada en ambos procesos utilizando un semáforo binario para la exclusión mutua de la sección crítica y dos semáforos contadores para eliminar la espera ocupada en el productor y en el consumidor.

Código del Productor

```
#define BUFFER_SIZE 5

int main(void)
{
   int in = 0 ;
   while(1)
    {
      while ( count == BUFFER_SIZE) ;   // Espera ocupada
      count++;   // Sección crítica
      buffer[in] = producirElemento();
      in = (in+1) % BUFFER_SIZE;
   }
   return 0 ;
}
```

Código del Consumidor

```
#define BUFFER_SIZE 5

int main(void)
{
   char elemento ;
   int out = 0 ;
   while(1)
    {
      while ( count == 0 ) ;   // Espera ocupada
      count--;   // Sección crítica
      elemento = buffer[out];
      out = (out+1) % BUFFER_SIZE;
   }
   return 0 ;
}
```

Creamos memoria compartida para buffer y count

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

int main(int argc, char *argv[])
{
  int shmid = shmget(0xa,5,IPC_CREAT|IPC_EXCL|0600);
  printf("shmid %d\n",shmid);
  shmid = shmget(0xb,sizeof(int),IPC_CREAT|IPC_EXCL|0600);
  printf("shmid %d\n",shmid);
  exit(0);
}
```

Inicializamos la memoria compartida para buffer y count

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

int main(int argc, char *argv[])
{
  int shmidbuf = shmget(0xa,0,0);
  printf("shmidbuf %d\n",shmidbuf);
  int shmidcou = shmget(0xb,0,0);
  printf("shmidcou %d\n",shmidcou);

  char * dirbuf = (char *) shmat(shmidbuf,0,0);
  int * dircou = (int *) shmat(shmidcou,0,0);

  printf("shmidbuf %p  shmidcou %p\n",shmidbuf , shmidcou);

  char * dirbufaux = dirbuf ;

  int i = 0 ;
  while(i < 5)
  {
    *dirbufaux = '.';
    dirbufaux++;
    i++;
  }

  *dircou = 0 ;
```

```
  shmdt(dirbuf);
  shmdt(dircou);

  exit(0);
}
```

Mostramos el contenido de la memoria compartida para buffer y count

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

int main(int argc, char *argv[])
{
  int shmidbuf = shmget(0xa,0,0);
  printf("shmidbuf %d\n",shmidbuf);
  int shmidcou = shmget(0xb,0,0);
  printf("shmidcou %d\n",shmidcou);

  char * dirbuf = (char *) shmat(shmidbuf,0,0);
  int * dircou = (int *) shmat(shmidcou,0,0);

  printf("shmidbuf %p  shmidcou %p\n",shmidbuf , shmidcou);

  char * dirbufaux = dirbuf ;

  int i = 0 ;
  while(i < 5)
  {
    printf("%c ",*dirbufaux );
    dirbufaux++;
    i++;
  }
  printf(" \n ");


  printf("%d \n",*dircou );

  shmdt(dirbuf);
  shmdt(dircou);

  exit(0);
}
```

Creamos los semáforos

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

int main(int argc, char *argv[])
{
  int semid = semget(0xa,3,IPC_CREAT|IPC_EXCL|0600);
  printf("semid %d\n",semid);
  exit(0);
}
```

Inicializamos los semáforos

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

int main(int argc, char *argv[])
{
  int semid = semget(0xa,0,0);
  printf("semid %d\n",semid);

  semctl(semid,0,SETVAL,1); // SEMAFORO BINARIO de MUTUA EXCLUSIÓN
  semctl(semid,1,SETVAL,5); // SEMAFORO CONTADOR PARA EL PRODUCTOR reemplazo de la espera ocupada
  semctl(semid,2,SETVAL,0); // SEMAFORO CONTADOR PARA EL CONSUMIDOR reemplazo de la espera ocupada
  exit(0);
}
```

Mostramos el valor de los semáforos

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

int main(int argc, char *argv[])
{
  int semid = semget(0xa,0,0);
  printf("semid %d\n",semid);

  printf("sem 0 = %d\n",semctl(semid,0,GETVAL));
```

```
  printf("sem 1 = %d\n",semctl(semid,1,GETVAL));
  printf("sem 2 = %d\n",semctl(semid,2,GETVAL));
  exit(0);
}
```

Links Videos

Ppt Sincronización de Procesos
https://youtu.be/rLDkdAc0NIc

C Linux Semáforos System V 01
https://www.youtube.com/watch?v=ub2YTpCq3Aw

C Linux Semáforos System V Ejemplo 02
https://www.youtube.com/watch?v=vvI2-CZ1Bik

C Linux Semáforos System V Otros Dos Ejemplos 03
https://www.youtube.com/watch?v=Gy4baWme_ek