

Algoritmos y Programación Paralela

Dra. Ing. Ana Cori Morón

EL PROBLEMA DEL PRODUCTOR CONSUMIDOR

- Es un problema común en sistemas operativos, un proceso productor genera información que es utilizada por un proceso consumidor

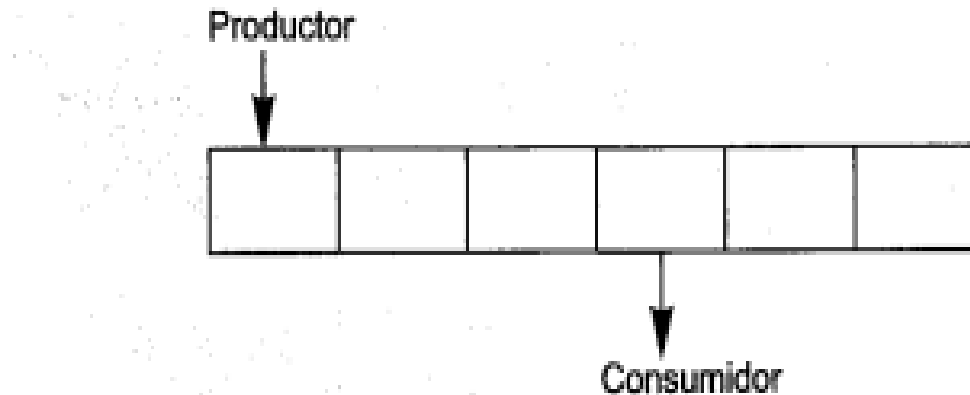


Figura 4.1. Esquema de trabajo del productor y el consumidor.

PROCESOS PRODUCTOR-CONSUMIDOR

process productor

begin

repeat

producir elemento;

protocolo de entrada;

Sección crítica → **insertar elemento en el buffer;**

protocolo de salida;

forever

end;

process consumidor

begin

repeat

protocolo de entrada;

extraer elemento en el buffer;

protocolo de salida;

consumir elemento;

forever

end;

Sección
crítica

Qué variables utilizaremos

- Para dar solución a este problema usaremos los siguientes semáforos:
 - **mutex:** permitirá el acceso en exclusión mutua al buffer.
 - **ocupados:** permitirá conocer la cantidad de posiciones ocupadas. Se inicializará a 0.
 - **libres:** permitirá conocer la cantidad de posiciones vacías en el buffer. Se inicializará a n .
 - **n :** tamaño del buffer.
- Consideraciones
 - Bloquear al proceso consumidor cuando no haya elementos para consumir (es decir cuando el buffer esté vacío, o cuando el semáforo ocupados = 0)
 - Bloquear al proceso productor cuando no haya espacio en el buffer para colocar los elementos producidos (es decir cuando el buffer esté lleno, o cuando el semáforo libres = 0)

process productor

begin

repeat

producir elemento;

protocolo de entrada;

insertar elemento en el buffer;

protocolo de salida;

forever

end;

Proceso Productor

Producir_elemento

Wait (libres)

Wait (mutex)

Insertar_elemento_en_el_buffer

Signal (mutex)

Signal (ocupados)

Proceso Consumidor

Wait (ocupados)

Wait (mutex)

Extraer_elemento_en_el_buffer

Signal (mutex)

Signal (libres)

Consumir_elemento

process consumidor

begin

repeat

protocolo de entrada;

extraer elemento en el buffer;

protocolo de salida;

consumir elemento;

forever

end;

Sección
crítica

Inicializar:
Initial(mutex,1)
Initial(ocupados,0)
Initial(libres,n)

Sección crítica

Para dar solución a este problema usaremos los siguientes semáforos:

mutex: permitirá el acceso en exclusión mutua al buffer.

ocupados: permitirá conocer la cantidad de posiciones ocupadas. Se inicializará a 0.

libres: permitirá conocer la cantidad de posiciones vacías en el buffer. Se inicializará a n .

n: tamaño del buffer.

```
Process productor
begin
  repeat
    producir elemento;
    wait(libres);
    wait(mutex);
    cola[final]=elemento;
    final=(final+1) mod n
    signal(mutex);
    signal(ocupados);
  forever
end;
```

```
process consumidor
begin
  repeat
    wait(ocupados);
    wait(mutex);
    elemento=cola[frente];
    frente=(frente+1) mod n
    signal(mutex);
    signal(libres);
    consumir elemento;
  forever
end;
```

Inicializar:

Initial(mutex,1)

Initial(ocupados,0)

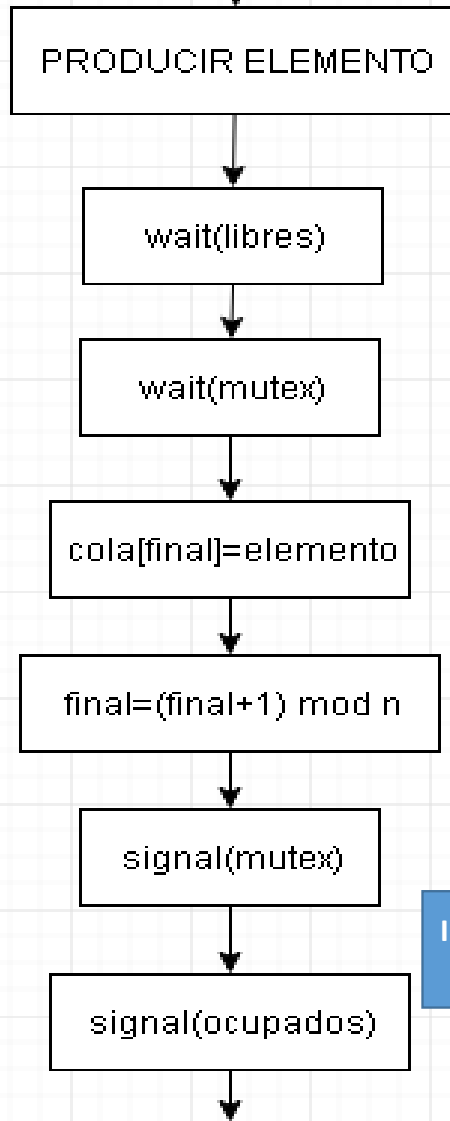
Initial(libres,n)

Diagrama de flujo

Generar un numero aleatorio
random (10)

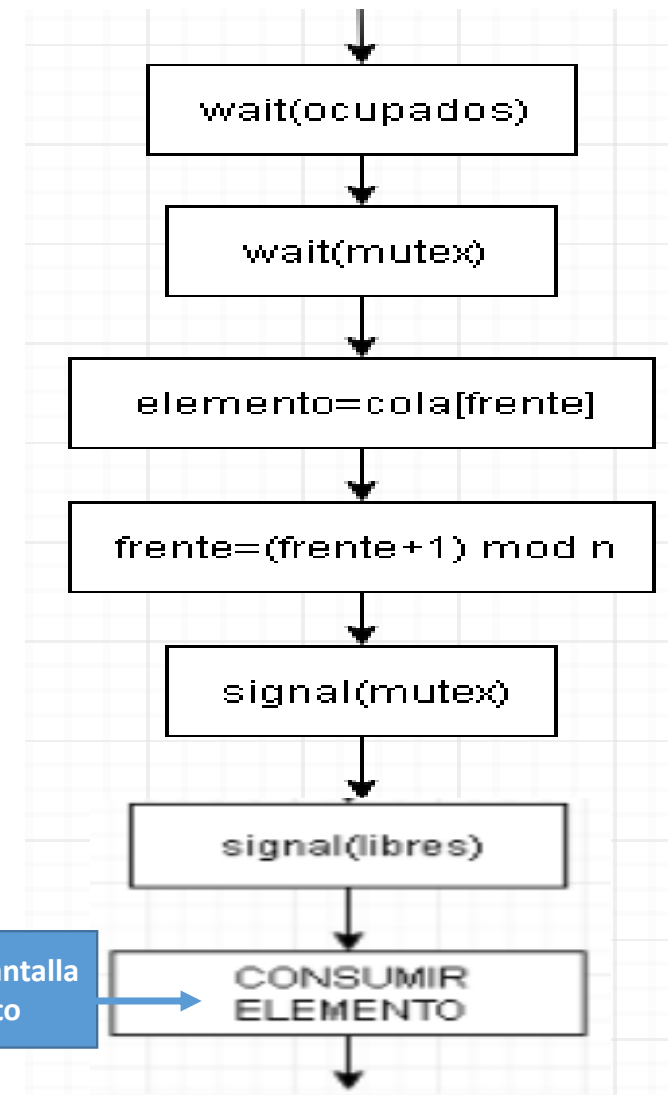
Inicializar:
Initial(mutex,1)
Initial(ocupados,0)
Initial(libres,n)

PROCESO PRODUCTOR



Imprimir en pantalla
el elemento

PROCESO CONSUMIDOR



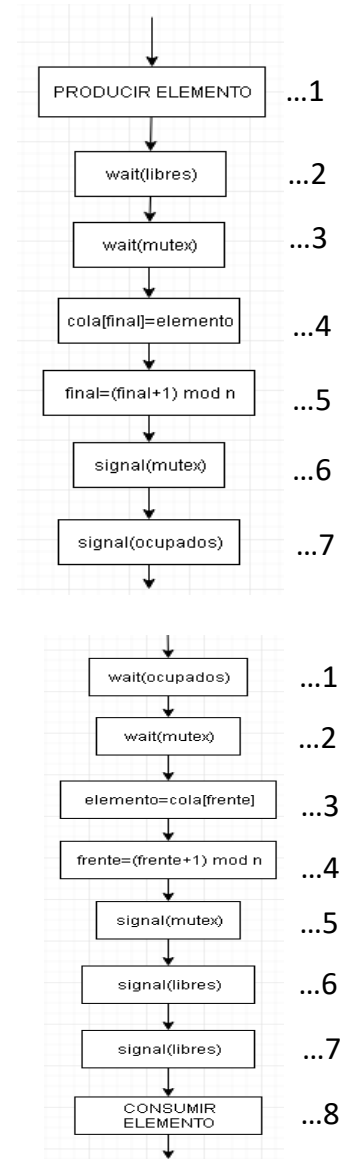
EJECUCIÓN

Suponiendo que inicia la ejecución el proceso consumidor

n=3

Producir elemento puede ser generar un numero aleatorio entre 0 y 10

Tiempo	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
libres	3	3	3	2	2	2	2	2	2						
mutex	1	1	1	1	0	0	0	1	1						
ocupados	0	Blq	Blq	Blq	Blq	Blq	Blq	Blq	Dblq						
cola[final]						7	7	7	7						
final	3	3	3	3	3	3	1	0	0						
frente	0	0	0	0	0	0	0	0	0						
elemento			7	7	7	7	7	7	7						
Pprod			1	2	3	4	5	6	7						
Pcons		1								2					



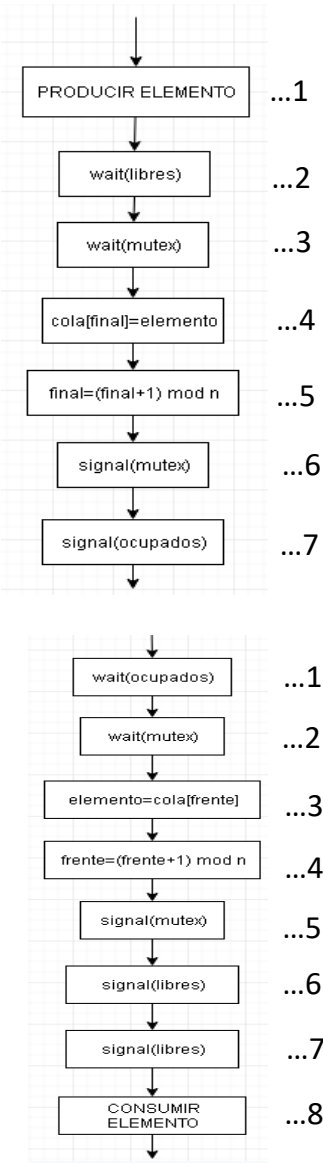
EJECUCIÓN

Suponiendo que inicia la ejecución el proceso productor

n=3

Producir elemento puede ser generar un numero aleatorio entre 0 y 10

Tiempo	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
libres	3	3	2	2	2	2	2	2	2	2	2	2	1	1	
mutex	1	1	1	0	0	0	1	1	0	0	0	1	1	1	
ocupados	0	Blq	Blq	Blq	Blq	Blq	Blq	Dbl	0	0	0	0	0	0	
cola[final]					7	7	7	7	7						
final	2	2	2	2	2	0	0	0	0						
frente	0	0	0	0	0	0	0	0	0						
elemento	7					7	7	7	7						
Pprod	1		2	3	4	5	6	7							
Pcons		1							2						



EJERCICIOS

- Implementar para varios procesos productores y varios procesos consumidores con variables tipo proceso
- <https://www.youtube.com/watch?v=pE8R5zypLLY>