

SISTEMAS CONCURRENTE Y DISTRIBUIDOS

Pablo Manresa Nebot.
2º A2

Índice:

1. SC FIFO Y LIFO MÚLTIPLE	-----	1
2. SC FIFO Y LIFO MÚLTIPLE	-----	2
3. PROBLEMA DE LOS FUMADORES	-----	3
4. PROBLEMA DEL BARBERO DURMIENTE	-----	4

1. SC LIFO Y FIFO MÚLTIPLE.

Con la semántica señalar y continuar, el proceso señalador continúa ejecutándose dentro del monitor, después de signal. El señalado deja la cola de condición y espera en la cola del monitor para volver a adquirir la exclusión mútua.

El proceso señalador y los demás, pueden hacer false la condición después de que el proceso señalado salga de la cola condición, dando errores en la condición de los lectores;

```
while ( primera_libre == 0 )  
    ocupadas.wait( guarda );
```

otro proceso puede ganar la exclusión mútua, y de este modo entrar al procedimiento leer(), y ahí ya no cumple la condición primera_libre == 0. Debido a que, se decrementa la variable primera_libre en 1, puede darse el caso de que el lector que anteriormente fue desbloqueado adquiera exclusión mútua y continúe el procesamiento del monitor por donde anteriormente iba, de este modo, no comprobaría la condición, y leería del vector la casilla -1.

De este modo, el proceso que ha sido señalado puede no garantizar que se cumpla la condición al terminar, siendo necesario volverla a comprobar.

Como conclusión, anteriormente funcionaba debido a que, sólo había una hebra productora y otra consumidora, siendo imposible que otras hebras modifiquen los valores de condición.

2. SU LIFO Y FIFO MÚLTIPLE.

Usando la semántica señalar y espera urgente ya no es necesario el bucle `while`, ya que, cuando la hebra señalada se desbloquee, continuará directamente su ejecución, sin que otra hebra pueda interferir, por lo que, no hará falta volver a comprobar la condición.

Las variables de condición usadas son, **libres** y **ocupadas** que sirven para controlar cuando se puede escribir y cuando leer, de modo que, si:

- No se puede leer porque `n` o primera libre son `== 0`, se produce `ocupadas.wait()` y tendrán que esperar.

- En cambio si primera_libre o `n > 0`, se leerá y a continuación se enviará un signal a libres(`libres.signal();`).

- Con la escritura ocurre lo mismo, si primera_libre o `n == num_celdas_total` entonces `libres.wait()`, se escribe en el buffer y a continuación se envía un signal a ocupadas(`ocupadas.signal();`).

3. PROBLEMA DE LOS FUMADORES.

Las variables permanentes son:

```
static const int num_fumadores = 3;  
Que indica el número total de fumadores.
```

```
int ingr_disp;  
Que indica el ingrediente disponible.
```

Variables adicionales:

```
mutex mtx;  
Para acceder en exclusión mutua a los cout para mostrar por pantalla.  
  
std::vector<std::string> ingredientes{"cerillas", "tabaco", "papel"};  
Para poder mostrar con facilidad el ingrediente producido/consumido.
```

Las colas son:

```
CondVar mostr_vacio,  
Para indicar que el mostrador está vacío y puede producirse un nuevo  
ingrediente.
```

```
fumador[num_fumadores];
```

Número de fumadores del problema, siendo 3. Cada uno está asociado a un ingrediente.

Pseudo-código de los tres procedimientos del monitor:

```
def Procedure ponerIngrediente(ingrediente)  
    ingrediente_del_monitor := ingrediente  
    fumador[ingrediente_del_monitor].signal()  
end procedure
```

```
def Procedure obtenerIngrediente(fumador)  
    Si fumador != ingrediente_del_monitor then  
        fumador[fumador].wait()  
  
    ingrediente_del_monitor := -1 //Consumido  
    mostrador_vacio.signal() //Puede producir nuevo ingrediente  
end procedure
```

```
def Procedure esperarRecogidaIngrediente  
    Si ingrediente_del_monitor != -1 then  
        mostrador_vacio.signal() //Hay un ingrediente en el mostrador no  
consumido  
end procedure
```

Salida del programa tras la ejecución:

```
Fumadores con monitor
Fumador 0 necesita ingrediente cerillas
Fumador 1 necesita ingrediente tabaco
Fumador 2 necesita ingrediente papel
Se ha producido el ingrediente 1 (tabaco)
Se ha puesto ingrediente tabaco
Retirado ingrediente: tabaco por el fumador 1
Se ha producido el ingrediente 0 (cerillas)
Se ha puesto ingrediente cerillas
Fumador 1 : empieza a fumar (95 milisegundos)
Retirado ingrediente: cerillas por el fumador 0
Se ha producido el ingrediente 2 (papel)
Se ha puesto ingrediente papel
Fumador 0 : empieza a fumar (149 milisegundos)
Retirado ingrediente: papel por el fumador 2
Fumador 2 : empieza a fumar (158 milisegundos)
Se ha producido el ingrediente 1 (tabaco)
Se ha puesto ingrediente tabaco
Fumador 1 : termina de fumar, comienza espera de ingrediente.
Se ha producido el ingrediente 2 (papel)
Se ha puesto ingrediente papel
Retirado ingrediente: tabaco por el fumador 1
Fumador 1 : empieza a fumar (59 milisegundos)
Fumador 0 : termina de fumar, comienza espera de ingrediente.
Fumador 1 : termina de fumar, comienza espera de ingrediente.
Fumador 2 : termina de fumar, comienza espera de ingrediente.
Se ha producido el ingrediente 1 (tabaco)
Se ha puesto ingrediente tabaco
Retirado ingrediente: papel por el fumador 2
Fumador 2 : empieza a fumar (159 milisegundos)
Retirado ingrediente: tabaco por el fumador 1
Fumador 1 : empieza a fumar (30 milisegundos)
Se ha producido el ingrediente 0 (cerillas)
Se ha puesto ingrediente cerillas
Retirado ingrediente: cerillas por el fumador 0
Fumador 0 : empieza a fumar (63 milisegundos)
Se ha producido el ingrediente 2 (papel)
Se ha puesto ingrediente papel
Fumador 1 : termina de fumar, comienza espera de ingrediente.
Fumador 0 : termina de fumar, comienza espera de ingrediente.
Fumador 2 : termina de fumar, comienza espera de ingrediente.
Se ha producido el ingrediente 2 (papel)
Retirado ingrediente: papel por el fumador 2
Fumador 2 : empieza a fumar (98 milisegundos)
Se ha puesto ingrediente papel
Fumador 2 : termina de fumar, comienza espera de ingrediente.
Se ha producido el ingrediente 1 (tabaco)
```

Se ha puesto ingrediente tabaco
Retirado ingrediente: papel por el fumador 2
Fumador 2 : empieza a fumar (181 milisegundos)
Retirado ingrediente: tabaco por el fumador 1
Fumador 1 : empieza a fumar (61 milisegundos)
Se ha producido el ingrediente 0 (cerillas)
Se ha puesto ingrediente cerillas
Retirado ingrediente: cerillas por el fumador 0
Fumador 0 : empieza a fumar (177 milisegundos)
Se ha producido el ingrediente 2 (papel)
Se ha puesto ingrediente papel
Fumador 1 : termina de fumar, comienza espera de ingrediente.
Fumador 0 : termina de fumar, comienza espera de ingrediente.
Fumador 2 : termina de fumar, comienza espera de ingrediente.
Se ha producido el ingrediente 1 (tabaco)
Se ha puesto ingrediente tabaco
Retirado ingrediente: papel por el fumador 2
Fumador 2 : empieza a fumar (155 milisegundos)
Retirado ingrediente: tabaco por el fumador 1
Fumador 1 : empieza a fumar (116 milisegundos)
Se ha producido el ingrediente 2 (papel)
Se ha puesto ingrediente papel
Fumador 1 : termina de fumar, comienza espera de ingrediente.
Fumador 2 : termina de fumar, comienza espera de ingrediente.
Se ha producido el ingrediente 1 (tabaco)
Se ha puesto ingrediente tabaco
Retirado ingrediente: papel por el fumador 2
Fumador 2 : empieza a fumar (40 milisegundos)
Retirado ingrediente: tabaco por el fumador 1
Fumador 1 : empieza a fumar (94 milisegundos)
Se ha producido el ingrediente 0 (cerillas)
Se ha puesto ingrediente cerillas
Retirado ingrediente: cerillas por el fumador 0
Fumador 0 : empieza a fumar (97 milisegundos)

3. PROBLEMA DEL BARBERO DURMIENTE.

Las variables permanentes son:

int clientes

Que indica el número de clientes que han entrado en la barberia, siendo mayor o igual que 1 si han de esperar en la sala de espera y 0 si el barbero debe dormir.

Las variables de condición son:

barbero

Indica si el barbero está dormido o atendiendo a un cliente.

Se envía un wait a barbero desde el método siguienteCliente si los clientes son igual a 0, y un signal por parte de un cliente desde el método cortarPelo(cliente) si barbero está durmiendo, para que, de este modo, lo pueda atender.

silla

Indica si hay algún cliente que está siendo atendido por el barbero.

Se envía un wait desde cortarPelo(cliente), ya que dicho cliente va a ser atendido, y un signal desde finCliente() para indicar que la silla vuelve a estar vacía debido a que, el barbero ha finalizado con dicho cliente.

sala_espera

Indica a los clientes que están en cola, esperando a ser pelados.

Se envía un wait desde el método cortarPelo(cliente) si clientes es mayor que 1, indicando que el barbero está ocupado. Desde el método siguienteCliente se envía un signal a la sala de espera si la silla está vacía, indicando que no hay ningún cliente siendo atendido por el barbero, por lo tanto puede pasar el siguiente.

Pseudo-código:

```
def Procedure cortarPelo(cliente)
    cliente++ //entra_otro_cliente

    Si barbero.dormido then
        barbero.signal() //despierta a barbero
    Sino
        si clientes > 1 then
            sala_espera.wait() //No puede pasar aún el siguiente
            cliente porque ya hay uno que está siendo atendido

            silla.wait() // Pasa el siguiente cliente a ser atendido

end Procedure
```

```

def Procedure siguienteCliente()
    Si clientes == 0 then
        barbero.wait() // El barbero puede dormir

    Si silla.vacia then
        sala_espera.signal() //Pasa siguiente cliente

end Procedure

```

```

def Procedure finCliente()
    silla.signal() //Silla disponible para el próximo cliente
end Procedure

```

Salida del programa tras la ejecución:

En este caso, nuestro número de clientes(hebras cliente) son 10:

```
constexpr auto num_clientes = 10;
```

```

-----Barberia-----
El cliente 1 entra a la barberia
El cliente 1 se está pelando
El cliente 4 entra a la barberia
El cliente 4 va a la sala de espera
El cliente 2 entra a la barberia
El cliente 2 va a la sala de espera
El cliente 0 entra a la barberia
El cliente 0 va a la sala de espera
El cliente 3 entra a la barberia
El cliente 3 va a la sala de espera
El cliente 5 entra a la barberia
El cliente 5 va a la sala de espera
Se está cortando el pelo al cliente
El cliente 6 entra a la barberia
El cliente 6 va a la sala de espera
El cliente 8 entra a la barberia
El cliente 8 va a la sala de espera
El cliente 9 entra a la barberia
El cliente 9 va a la sala de espera
El cliente 7 entra a la barberia
El cliente 7 va a la sala de espera
FIN cliente
El cliente 1 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 4 se está pelando
Se está cortando el pelo al cliente

```


El cliente 1 entra a la barberia
El cliente 1 va a la sala de espera
FIN cliente
El cliente 4 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 2 se está pelando
Se está cortando el pelo al cliente
El cliente 4 entra a la barberia
El cliente 4 va a la sala de espera
FIN cliente
El cliente 2 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 0 se está pelando
Se está cortando el pelo al cliente
FIN cliente
El cliente 0 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 3 se está pelando
Se está cortando el pelo al cliente
El cliente 2 entra a la barberia
El cliente 2 va a la sala de espera
El cliente 0 entra a la barberia
El cliente 0 va a la sala de espera
FIN cliente
El cliente 3 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 5 se está pelando
Se está cortando el pelo al cliente
El cliente 3 entra a la barberia
El cliente 3 va a la sala de espera
FIN cliente
El cliente 5 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 6 se está pelando
Se está cortando el pelo al cliente
FIN cliente
El cliente 6 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 8 se está pelando
Se está cortando el pelo al cliente
FIN cliente
El cliente 8 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 9 se está pelando
Se está cortando el pelo al cliente
El cliente 6 entra a la barberia
El cliente 6 va a la sala de espera
El cliente 8 entra a la barberia
El cliente 8 va a la sala de espera
El cliente 5 entra a la barberia
El cliente 5 va a la sala de espera
FIN cliente
El cliente 9 espera fuera de la barberia
Se llama al siguiente cliente

El cliente 7 se está pelando
Se está cortando el pelo al cliente
El cliente 9 entra a la barberia
El cliente 9 va a la sala de espera
FIN cliente
El cliente 7 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 1 se está pelando
Se está cortando el pelo al cliente
FIN cliente
El cliente 1 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 4 se está pelando
Se está cortando el pelo al cliente
El cliente 7 entra a la barberia
El cliente 7 va a la sala de espera
FIN cliente
El cliente 4 espera fuera de la barberia
Se llama al siguiente cliente
El cliente 2 se está pelando
Se está cortando el pelo al cliente
El cliente 1 entra a la barberia
El cliente 1 va a la sala de espera
FIN cliente