

Informe Trabajo Práctico 1 - Técnicas de Programación Concurrente (75.59)

Victoria Perelló (89608) Agustín Mezzina (89637)
Pablo Rodríguez (93970)

15 de mayo de 2014

Índice

1. Introducción	2
1.1. Descripción del problema	2
1.2. Modo de uso	2
2. Procesos	2
3. Comunicación entre procesos	3
4. Problemas típicos de concurrencia	3
5. Mecanismos de concurrencia a utilizar	3
5.1. Init - Jefe de estación	4
5.2. Jefe de estación - Empleado(s)	4
5.3. Empleado - Empleado	4
5.4. Empleado(s) - Consultar Caja	5
6. Otras consideraciones	5
6.1. Inicio y finalización	5
6.2. Log	5
7. Diagramas	5
7.1. Diagrama de clases	5
7.2. Diagrama de transición de estados: Jefe de Estación	8

1. Introducción

1.1. Descripción del problema

El presente trabajo práctico consiste en una simulación de la dinámica de operación de una estación de servicio. En ella, los operarios deben atender a los clientes entrantes utilizando los recursos de los cuales dispone la estación. El eje del problema es la comunicación y la sincronización de los operarios en el uso de dichos recursos para el correcto funcionamiento del negocio.

1.2. Modo de uso

Adjunto a este informe se encuentra el código fuente de la aplicación y un archivo Makefile para realizar la compilación. El sistema debe ser compilado utilizando la directiva “make all”, el cuál generará cinco ejecutables con los programas que van a interactuar en el sistema. Para comenzar la simulación, debe ejecutarse en una consola el archivo “ConcuStation”. El sistema pedirá al usuario que ingrese el número de empleados y de surtidores que posee la estación. Luego de ingresados, dará inicio a la simulación. Durante el transcurso de la misma, el usuario puede ingresar dos comandos por entrada estándar:

1. “c”: Consultar Caja. Imprime por salida estándar la recaudación actual de la Caja.
2. “q”: Finalizar simulación.

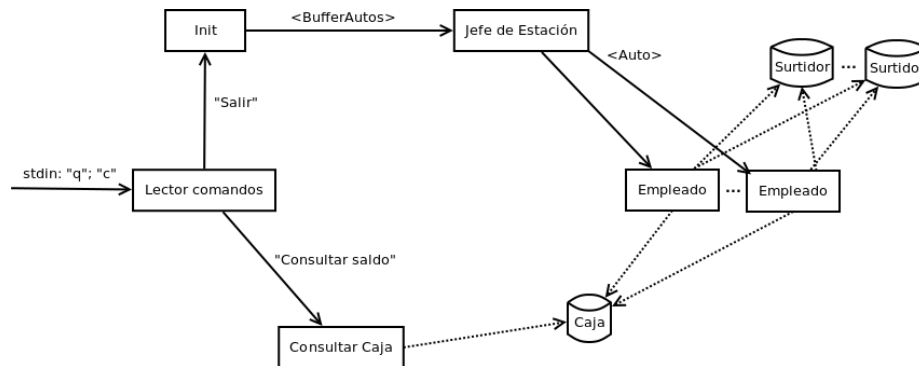
2. Procesos

La aplicación se divide en los siguientes procesos:

1. Init: genera un auto cada cierta cantidad aleatoria de tiempo transcurrido y los envía al jefe de estación para su procesamiento.
2. Jefe de estación: recibe los autos entrantes y los asigna a empleados libres. En caso de no haber empleados libres, los despacha.
3. Empleado: asigna autos a surtidores libres. Si no hay surtidores libres, espera a que se libere alguno. Realiza el servicio. Cobra y guarda la recaudación en la caja.
4. Consulta caja registradora: consulta asincrónica sobre el valor de la recaudación en caja.
5. Lector de comandos: escucha comandos entrantes por entrada estándar y toma las medidas correspondientes para responder a ellos.

3. Comunicación entre procesos

En el siguiente diagrama, se muestra el esquema de comunicación entre los procesos del sistema. Incluye los datos que intercambian los procesos que se comunican entre sí y los recursos a los que deben acceder de manera concurrente.

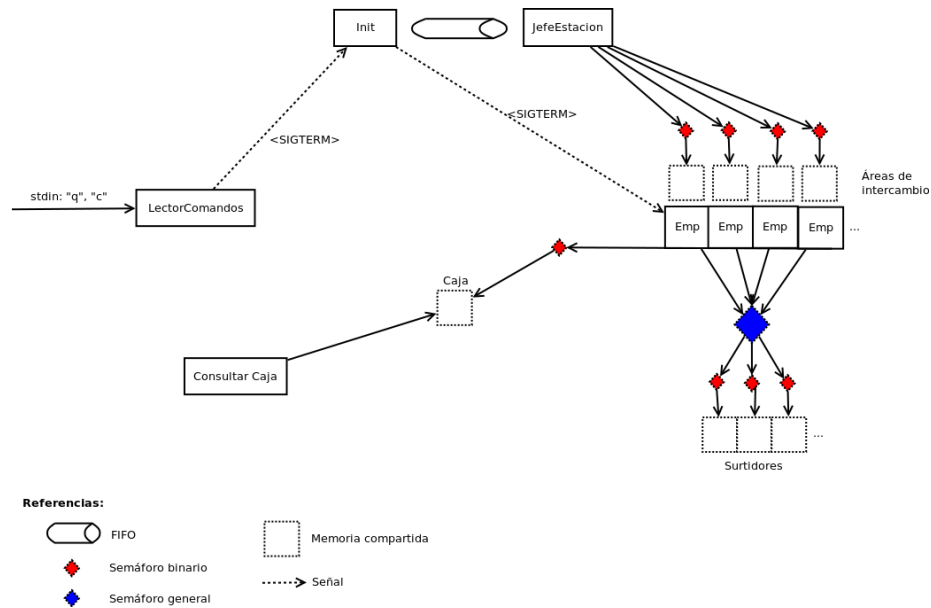


4. Problemas típicos de concurrencia

1. Init - Jefe de estación: Productor - Consumidor. El proceso base hace las veces de Productor, ingresando autos nuevos a un buffer compartido donde posteriormente el Jefe de Estación, en este caso Consumidor, los retira según su orden de llegada y los procesa.
2. Jefe de estación - Empleado(s): Productor - Consumidor. Se presenta un caso similar al anterior, con la variante de que el Productor (Jefe de estación) debe determinar a cuál de los Consumidores (Empleados) enviar el recurso generado.
3. Empleado - Empleado: Sección Crítica. Los procesos deben excluirse mutuamente al ejecutar código que acceda a los recursos Surtidores, ya que el mismo Surtidor no puede ser utilizado por dos Empleados al mismo tiempo.
4. Empleado(s) - Consultar Caja: Sección Crítica. Los procesos deben excluirse mutuamente al acceder a la Caja, ya sea para leer como para escribir.

5. Mecanismos de concurrencia a utilizar

En el siguiente diagrama se muestran los mecanismos de concurrencia utilizados para lograr la comunicación y sincronización de procesos:



5.1. Init - Jefe de estación

El buffer de autos se implementó utilizando un named pipe (o FIFO), de manera que el proceso inicial ingresa los autos al canal a medida que los va produciendo y el jefe de estación los toma en orden para procesarlos. Este mecanismo de concurrencia provee no sólo comunicación entre ambos procesos, sino que también sincronismo.

5.2. Jefe de estación - Empleado(s)

Cada Empleado mantiene con el Jefe de Estación un espacio de intercambio en memoria compartida, donde se guarda información sobre el estado del Empleado (libre u ocupado) y el auto que debe ser atendido. Cada uno de estos espacios de intercambio está protegido con un semáforo binario para lograr exclusión mutua entre el Jefe y el Empleado correspondiente en el acceso al mismo, ya sea para lectura o para escritura.

5.3. Empleado - Empleado

Los Surtidores están guardados en bloques de memoria compartida accesibles a todos los Empleados. Dicho bloque está protegido por un semáforo general que permite el acceso al área de memoria cuando haya algún surtidor libre. Cada espacio de memoria donde se encuentra un surtidor está, a su vez, protegido por un semáforo binario.

5.4. Empleado(s) - Consultar Caja

Estos procesos acceden a un espacio de memoria compartida donde está la Caja. Se excluyen mutuamente mediante un semáforo binario para coordinar el acceso a la misma.

6. Otras consideraciones

6.1. Inicio y finalización

Al iniciar la simulación, el proceso Init crea todas las memorias compartidas y semáforos (estos últimos también los inicializa) necesarios para el funcionamiento del sistema, así como también el FIFO que sirve como canal de comunicación con el Jefe de empleados. Luego lanza los procesos Jefe de Empleados, Lector de Comandos y todos los Empleados. El proceso Consultar Caja es lanzado por el Lector de Comandos cada vez que el usuario ingresa el comando “c”. Este proceso finaliza luego de imprimir el valor de la Caja en pantalla. Cuando el Lector de Comandos recibe el comando “q”, el sistema debe finalizar. Para esto, se envía la señal SIGTERM al proceso Init y este, a su vez, envía una señal similar a todos los Empleados y un mensaje especial al Jefe de Estación a través del FIFO. Estos procesos manejarán los mensajes eliminando todos sus recursos y finalizando correctamente. El proceso Init espera la finalización de los demás procesos y luego procede a eliminar todos los canales de comunicación creados inicialmente, para luego finalizar.

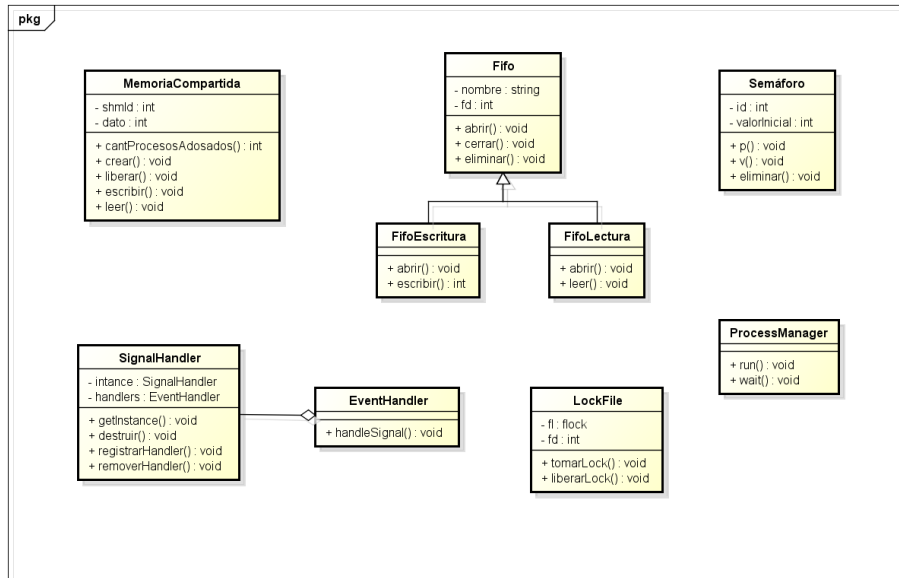
6.2. Log

Todos los procesos de la simulación informan las acciones que realizan mediante la escritura a un único archivo denominado “ConcuStation.log”. El mismo puede utilizarse para verificar el funcionamiento del sistema y el orden en el que se realizan las actividades. Para coordinar el acceso al archivo, se utilizan locks de escritura.

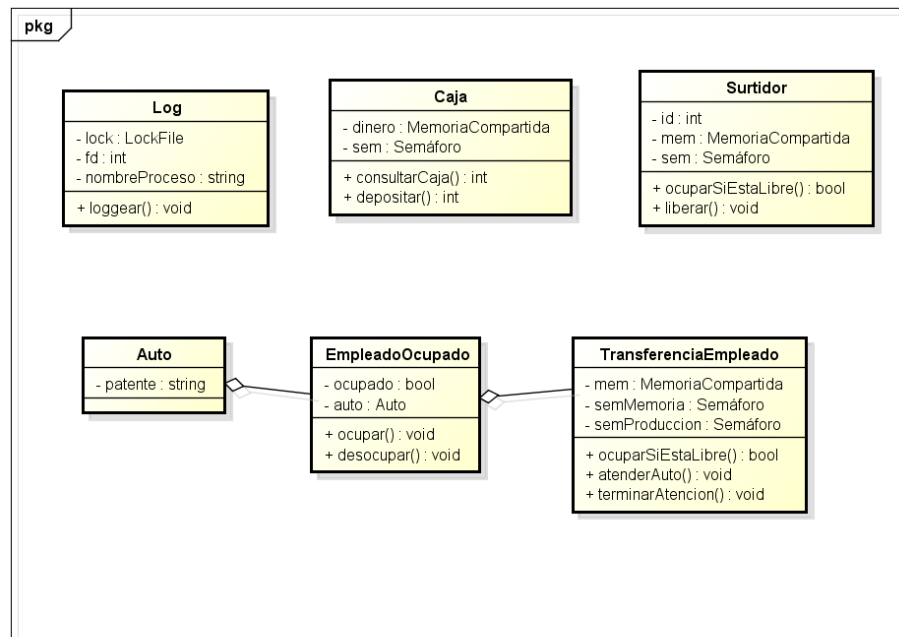
7. Diagramas

7.1. Diagrama de clases

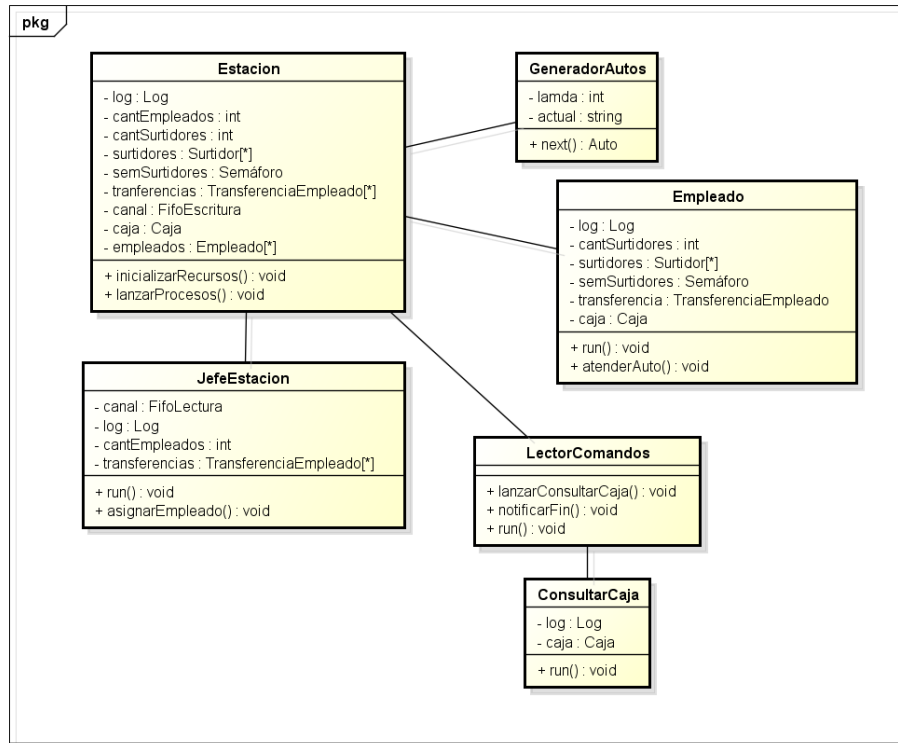
Mecanismos de concurrencia:



Recursos:



Procesos:



powered by Astah

7.2. Diagrama de transición de estados: Jefe de Estación

