Primer Proyecto
75.59 Concurrencia 1

Departamento de Computación
Facultad de Ingeniería
Universidad de Buenos Aires

24 de septiembre de 2012

Índice

1. Análisis del problema

1.1. División en procesos

En esta sección delinearemos el análisis del problema realizado por los integrantes del equipo. Se presenta en el orden en que se llevó a cabo la discusión de posibles soluciones, y es por eso que las ideas iniciales pueden presentar inconsistencias o soluciones incompletas.

Idea preliminar De los requerimientos del proyecto resulta evidente que es necesario modelar las *entradas*, las *salidas* y el *administrador* como procesos. Son los primeros quienes actualizarán el estado del estacionamiento: tanto la cantidad de vehículos que aumenta al ingresar un auto por una entrada y disminuye al salir un auto por una salida, como el monto facturado que aumenta cuando un auto sale en función del tiempo de permanencia; estos datos serán consultados por las entradas, para determinar si se admiten más autos, y por el administrador. Estas consultas y actualizaciones deben sincronizarse, y pueden considerarse operaciones críticas: no sería aceptable que una entrada dejara ingresar un auto en exceso de la capacidad del estacionamiento, o que una salida sume un nuevo importe facturado a un total desactualizado; ambas situaciones podrían suceder si se permite la existencia de *race conditions*.

Modelado de los autos Una vez analizado el problema de los procesos, es preciso plantear cómo serán representados y cómo se *moverán* los autos por el programa. Una alternativa es generar los autos directamente en las entradas, simulando un proceso estocastico utilizando números pseudoaleatorios alimentados con semillas dependientes de algún parámetro de la ejecución, como la hora. Así cada entrada ejecutaría esa simulación y procesaría solamente los autos generados por ella. Al momento de la generación sería posible predeterminar el tiempo de permanencia y la salida elegida. Esto presenta al menos dos problemas. Por un lado las salidas, una vez enteradas del momento de salida de cada vehículo, los pondrán en espera para ejecutar la salida de los que egresan en un mismo instante de manera secuencial. Para hacerlo utilizarían pausas programadas y ejecución *instantánea*, es decir que el paso del tiempo simulado se detiene para ejecutar las salidas. Esto contradice el requerimiento de que la simulación no se interrumpa para simular entradas ni salidas.

Por este motivo los resulta necesario modelar también a los *autos* como procesos. Esta solución admite la simulación de acciones simultáneas de distintos *autos*, y además permite simplificar los procesos de *entradas* y *salidas*.

Ellos serán los encargados de determinar cuándo quieren salir del estacionamiento, y también comunicarán a los procesos de *entrada* su intención de entrar al estacionamiento, quedando a la espera de una respuesta ya que podría no ser posible.

Generación de autos Dada la solución planteada, resulta necesario proponer un modelo de comunicación entre los *autos* y las *entradas* y salidas. La solución

más simple resulta que se generen en un proceso *principal* también encargado de generar, previamente, las *entradas*, las *salidas* y el *administrador*. Luego de su generación, los *autos* deberán ser encolados, a la espera de que alguna de las entradas les envíe un mensaje aceptando o rechazando su entrada. Algo similar sucederá a la salida: una vez que los *autos* determinen que es hora de salir, se encolarán esperando a que una *salida* los atienda.

Si bien esta alternativa parece viable, el que los *autos* esperen en colas generales provoca que las *entradas* y *salidas* deban sincronizar su acceso a las mismas, evitando nuevamente que los movimientos de los *autos* se ejecuten simultáneamente. La solución que se propone entonces es la existencia de una cola en cada *entrada* y en cada *salida*. De esta manera cada *auto* elije la puerta que utilizará, y se coloca en una cola independiente.

De esta manera, la única sincronización que se realiza es la inevitable, que implica ordenar la actualización y consulta del estado del estacionamiento.

1.2. Esquemas de comunicación

1.3. Mecanismos de concurrencia

2. Implementación