



PRACTICA 4

LÍNEA PRODUCCIÓN



JORGE LUCAS FERRER
839316

Índice

1. Planteamiento del problema	3
2. Mecanismos de resolución y resolución del problema	4

Planteamiento del problema

El enunciado de esta práctica nos propone simular el funcionamiento de una fábrica de teléfonos móviles con tres líneas de producción diferentes. El funcionamiento se basa en el empacamiento de dichos móviles en cajas, las cuales, aguantan un peso máximo.

Como era de libre elección yo he planteado el ejercicio con :

- Un número máximo de 2 móviles en el mostrador
- El peso de los móviles tipo A es de 120 gramos, los de tipo B de 210 y los de tipo C de 170g
- El peso máximo que aguantan las cajas es de 1000 gramos/1 kilogramo

Y yo le he añadido unos tiempos de delay (A = 1 segundo, B= 1.1 segundo, C=0.9 segundos) para que tengan una pequeña variación entre unos y otros.

Al plantear el problema he tenido dos grandes ideas, la primera consistía en hacer una tarea para cada tipo de móvil y que su función se basase en generar móviles de forma infinita y por otro lado, otra tarea para la recogida de las cajas y que su función fuese esperar hasta que le llamasen para recoger una caja y poner otra. Esta primera idea la descarté por que el funcionamiento de la última tarea se asemejaba mucho al funcionamiento del médico o del enfermero de la anterior práctica y, quería probar cosas diferentes.

La segunda idea y la que lleve a cabo se basa en tres tareas, una para cada tipo de móvil, las cuales se ocuparían de generar móviles y de hacer el recorrido de la fábrica. Durante este recorrido, si una de las tres tareas detectase que es necesario cambiar la caja por que no es posible meter más móviles por el peso, avisaría al programa (robot empacador) para cambiar dicha caja por otra vacía.

(Me gustaría ponerte los dibujos que he hecho para organizarme pero, los he hecho a mano)

Mecanismos de resolución y resolución del problema

Como ya he dicho en el anterior apartado, he hecho uso de tres tareas (**task**) para la resolución del problema. A parte de las tres tareas, también me he ayudado de los objetos protegidos para simular un monitor. Dentro de dicho monitor he declarado los métodos y “**entries**” necesarios para poder ejecutar el programa de la forma que yo quería (todos los procedimientos se hacen en exclusión mutua por ser parte de un objeto protegido).

Para entender mejor el monitor lo voy a desglosar en dos : la declaración(.ads) y la implementación(.adb). En la declaración se encuentran las variables que vamos a usar de forma protegida y la declaración de métodos que vamos a implementar. En la implementación, están desglosados los procedimientos y función, que son : **movilMostrador**, **sumaPeso** y **empaquetarMovil**.

- **movilMostrador** : lo primero es comentar que este procedimiento es un **entry** porque, de esta forma, podemos asegurarnos que en el mostrador no hay más de los móviles de los que caben y, además, conservamos así un orden **FIFO** (first in, first out). A parte de esto, mantiene un conteo de cuantos y de qué tipo de móviles van entrado.
- **sumaPeso** : esta función es simple, comprueba si el móvil que está por entrar a la caja puede entrar o se pasa por peso. En caso de pasarse devuelve **verdadero** y en el caso de igualar o quedarse por debajo del peso, devuelve **falso**
- **empaquetarMovil** : este último procedimiento se ocupa la parte final de la fábrica, es decir, del empaquetamiento. Para el correcto funcionamiento de este procedimiento hacemos uso de “**sumaPeso**”. Si **sumaPeso** es **falso**, el móvil es removido del mostrador y metido en la caja pero, si es **verdadero**, el móvil se quedará esperando (gracias a que el procedimiento es de un objeto protegido y se hace en exclusión mutua* se conserva el orden) y cambiará la caja por una vacía.

*en caso de que no se hiciese en exclusión mutua, habría convertido a **empaquetarMovil** en un **entry** y habría creado una variable booleana, inicializada en verdadero, que midiese la disponibilidad de las cajas (true si hay caja, false si no la hay) y, esta variable, sería la barrera de entrada del propio entry.

