



JUEGO DELIVERY

Trabajo Práctico 1

PROGRAMACIÓN CONCURRENTES

BOZZANO, Felipe - Legajo: 102050

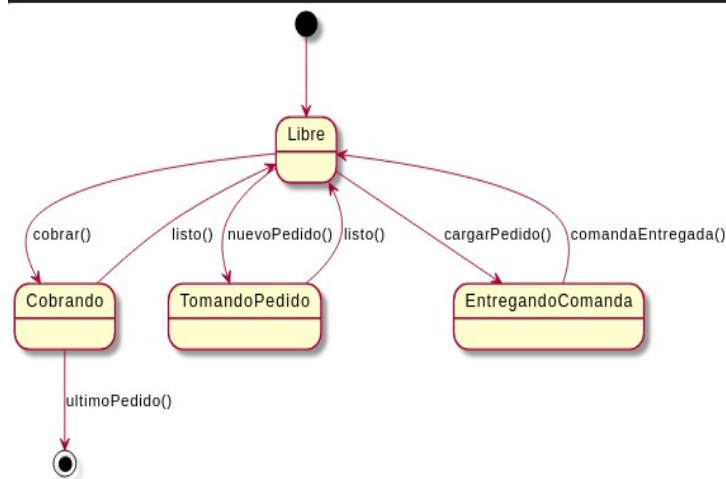
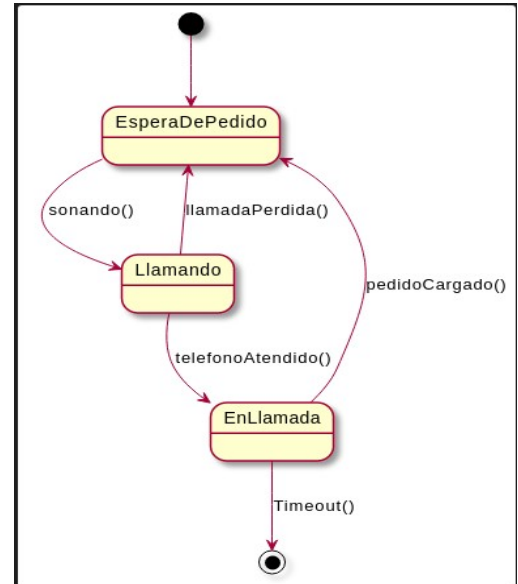
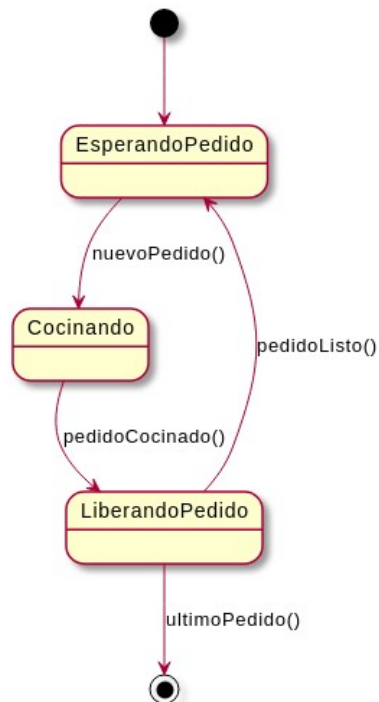
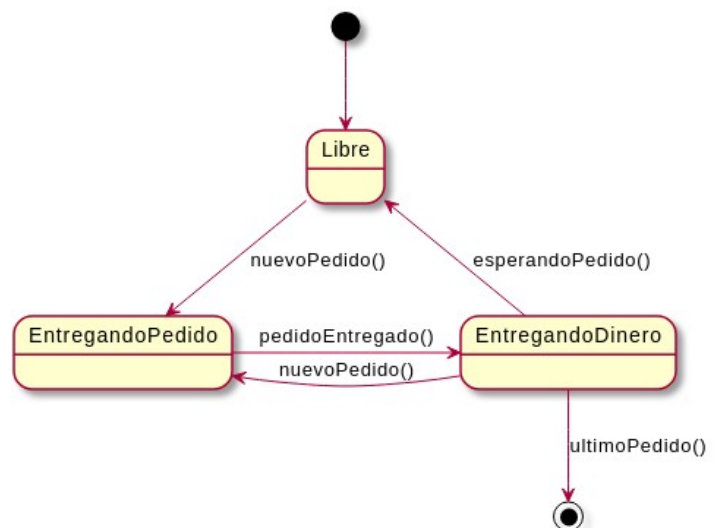
OLMOS, Francisco - Legajo: 102369

Profesor: Maximiliano Andrés Eschoyez

Problemática:

El objetivo de este proyecto es realizar un controlador de un juego de Delivery por medio de variables compartidas y sincronización entre procesos. El programa deberá manejar 3 procesos diferentes con sus correspondientes hilos. El proceso principal debe ser el que controla el sistema completo, y los otros dos son los productores y consumidores.

Debe haber un Teléfono que “suene” aleatoriamente por un tiempo determinado el cual debe ser atendido por el Encargado del local. En este caso el Encargado tomará el pedido que recibe del Teléfono y lo cargará junto al resto de pedidos por cocinar, sumando así un punto al score del jugador. Llegado el caso de que no pueda atender el Teléfono, entonces este pedido se perderá y no sumará puntos. Por otro lado los cocineros deberán revisar si hay pedidos por cocinar y cocinarlos. Una vez cocinados, los deberán colocar junto al resto de pedidos cocinados. Por último, los Delivery serán los encargados de repartir los pedidos listos y entregar el dinero al Encargado cuando éste esté disponible para cobrarles.

UML:EncargadoTeléfonoCocineroDelivery

Solución:

Para la implementación del programa contamos con un hilo Teléfono, un hilo Encargado, tres hilos Cocineros y dos hilos Delivery.

El hilo del Encargado es el responsable de atender el Teléfono, pasar los pedidos a los Cocineros y cobrar a los Delivery. Es el actor principal del juego ya que se encarga de la coordinación entre los demás actores y de avisarles cuando tienen que dejar de hacer sus tareas, y es controlado por el jugador.

El Teléfono implementa una alarma mediante señales IPC que le permite saber cuando tiene que mandar el último pedido que es falso para avisarle al Encargado que debe terminar el juego.

Para la comunicación entre el Encargado y los tres Cocineros implementamos un monitor con buffer circular, en donde dejamos los pedidos que el Teléfono le manda al Encargado (productor) y desde donde los Cocineros (consumidores) leen para poder cocinar. Una vez estén cocinados, se los guarda en otro monitor con buffer circular de donde los dos Delivery (consumidores) leen para entregarlos.

Los Delivery una vez que entregan los pedidos, vuelven a la pizzería y esperan a que el Encargado reciba el dinero del pedido y les cobre. Para ello implementamos una memoria compartida que cuenta con semáforos para la sincronización entre los hilos y el pedido a cobrar.

INSTRUCCIONES DE JUEGO:

El programa cuenta con un Menú de opciones que le permiten al usuario comenzar el juego apretando la tecla "1", ver su puntuación con "2" o salir apretando "3".

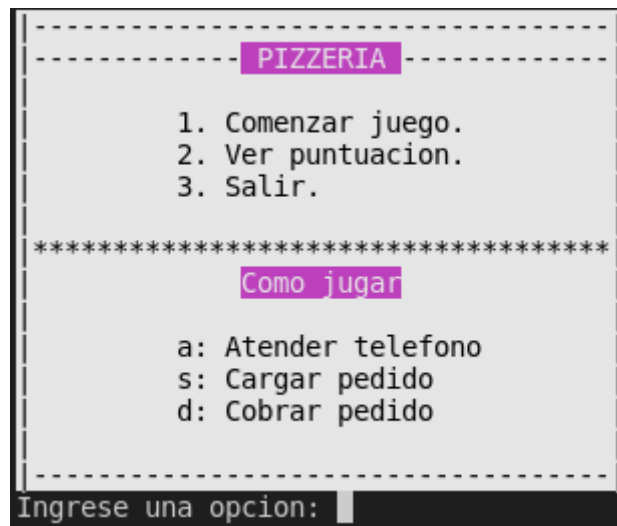
Para poder sumar puntos solo se debe atender el teléfono cuando este suene, lo cual se indicará con un mensaje en rojo diciendo "telefono sonando", apretando la tecla "a" y luego "ENTER". En caso de no atenderlo en el lapso de 1 segundo, se perderá el llamado y la posibilidad de incrementar la puntuación del jugador.

El siguiente paso es cargar los pedidos acumulados a los cocineros. Para ello se debe apretar la tecla “s” y luego ENTER cuando se le indique por pantalla un mensaje en color magenta indicando “comanda de pedido tanto lista para cargar”.

Es importante también cobrarle a los Delivery. Para poder hacer esto se debe apretar la tecla “d” y luego “ENTER” cuando se le indique por pantalla un mensaje en color amarillo diciendo “dejando dinero en caja de pedido tanto”.

Una vez se hayan cobrado todos los pedidos atendidos cocinados y entregados se deberá finalizar al juego apretando la letra “d” y luego “ENTER”.

El menú de opciones se verá de la siguiente forma:



Una ejecución del programa se verá de la siguiente forma:

```

a      telefono sonando
s      telefono atendido
a      comanda de pedido 2 lista para cargar
s      telefono sonando
a      telefono atendido
s      comanda de pedido 3 lista para cargar
d      Pedido 2 listo para cobrar
a      $300 guardados de pedido 2
s      telefono atendido
a      comanda de pedido 2 lista para cargar
s      telefono sonando
a      Pedido 3 listo para cobrar

a      Pedido 0 listo para cobrar
d      telefono atendido
s      comanda de pedido -1 lista para cargar
d      $100 guardados de pedido 0
s      Pedido 4 listo para cobrar
d      $500 guardados de pedido 4
a      Presione d para cerrar el local
s      Cerrando local
a      Escriba su nombre: 
  
```

Al apretar la letra “d” al final del juego cuando se le indica, deberá escribir su nombre con el que quiere que se le registre la puntuación obtenida. El programa genera un archivo de texto “puntuacion.txt” con las puntuaciones de los jugadores.

```

Francisco: 3
Matias: 4
Matias: 4
  
```

Como compilar el programa:

```
gcc -c -pthread -Wall -pedantic ./TP-1/MonitoresBuffer.c -o
MonitoresBuffer.o
```

```
gcc -pthread -Wall -pedantic ./TP-1/delivery_TP1.c MonitoresBuffer.o -
o prueba.c -lrt
```

Conclusión:

El correcto manejo de hilos garantiza la sincronización entre los actores de un programa. Ya sea utilizando semáforos, mutex, monitores, etc. Si los hilos tienen una implementación correcta, el programa general va a estar sincronizado.

Al comienzo el programa no permitía al usuario interactuar con el mismo, sino que corría automáticamente. Ingresar al usuario al sistema fue algo simple, ya que no variaba en gran medida la problemática principal del mismo, la sincronización.

La implementación de los monitores como una biblioteca es la mejor manera de ejecutar una solución para estos programas, ya que segrega la sincronización del funcionamiento del juego, y tener todo junto dificulta debuggear cuando existe un problema.