

Bloque 1 – Práctica 2

En este trabajo vamos a poner en práctica conocimientos relacionados con procesos, memoria compartida y semáforos.

A partir de aquí, vamos a ver una idea general de lo que hay que hacer y lo vamos a ir concretando poco a poco y punto por punto.

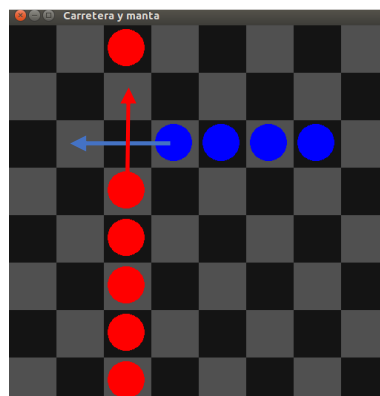
Ítem 1: Idea general de lo que hay que hacer

Vamos a simular el trayecto de dos trenes por un paso a nivel.

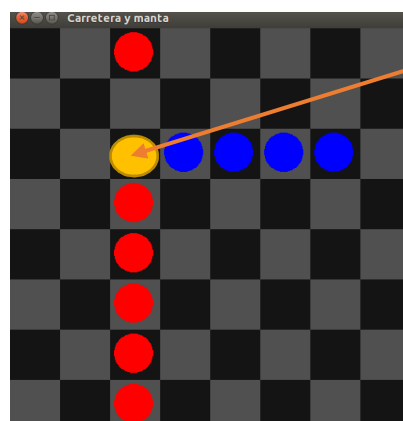
El paso a nivel en la casilla (2, 2) del mapa.

Tenemos dos trenes:

- El tren azul:
 - Se mueve en horizontal de derecha a izquierda sobre $y=2$.
 - Avanza una casilla a cada segundo.
 - Tiene 4 vagones.
- El tren rojo:
 - Se mueve en vertical de abajo a arriba sobre la $x=2$.
 - Avanza una casilla a cada dos segundos.
 - Tiene 6 vagones.



La casilla donde pueden colisionar estos dos trenes es la (2,2).

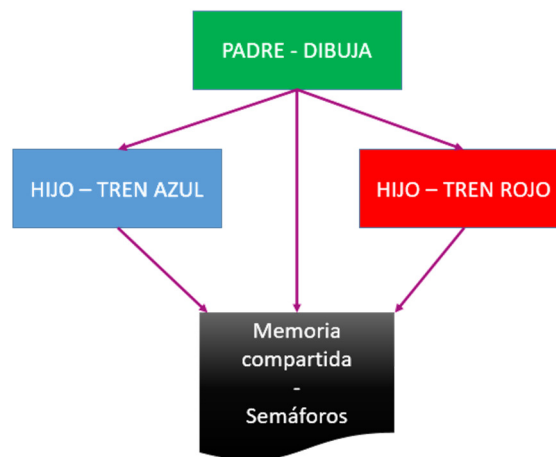


Cruce en la casilla (2,2)

Ítem 2: Organización en procesos

Tendremos tres procesos:

- Proceso padre
 - Inicializa la memoria compartida
 - Inicializa los semáforos
 - Pinta el mapa y la posición de los trenes
 - Recoge el estado de finalización de los hijos
 - Libera memoria compartida y semáforos
- Proceso tren (tanto el rojo como el azul)
 - Inicializa la primera posición.
 - Avanza teniendo en cuenta la restricción del cruce
 - Va indicando su posición en la zona de memoria compartida (que es la que está utilizando el padre para pintar)



Vamos a ver lo que se espera de cada uno procesos.

Ítem 3: Proceso padre

Se espera lo siguiente:

1. Debe reservar la memoria compartida
 - a. Para hacerlo, primero debes pensar qué necesitas que haya en esa zona de memoria compartida.
2. Debe inicializar el o los semáforos.
 - a. En primer lugar, debes pensar cuántos semáforos necesitas y qué zonas debes proteger de acceso compartido.
3. Crea a los hijos/trenes y les asigna el rol a cada uno (rojo o azul)
4. A partir de aquí, entra en el while principal, que consiste en pintar por pantalla las posiciones que van actualizando los hijos.
5. Al acabar, se debe recoger el estado de los hijos y liberar memoria compartida y semáforos.

Ítem 4: Proceso hijo/tren

Se espera lo siguiente:

1. Debe inicializar la posición del tren.

2. Debe mover el tren durante 20 iteraciones en las que se moverá una casilla en la dirección y sentido que le corresponda y con la temporización que toque (cada 1 segundo el tren azul y cada 2 el rojo).
3. Al acabar, desvincula la memoria compartida y finaliza normalmente.

Ítem 5: Consejos

1. No empieces a programar sin tener claro lo que tienes que hacer. Hazte pequeños ejemplos de código donde vayas probando lo que quieres hacer poco a poco. Así te aseguras de que lo que vas incorporando funciona. Por ejemplo, asegúrate de que los hijos inicializan correctamente las posiciones de los vagones y el padre lo ve correctamente.
2. Aprovecha la interfaz gráfica de la práctica anterior. Pierde el mínimo tiempo posible en ella.
3. No esperes hasta el último día. Seguramente la profe no podrá echarte una mano el día de antes de la entrega.

Ítem 6: Qué hay que entregar

El código

Debe cumplir con los siguientes requisitos:

1. Debe compilar (si no compila, no se corrige y se da por no entregado).
2. Los trenes deben moverse de la forma en que se indica y debe visualizarse en el padre.
3. Debe controlarse el paso a nivel. Los trenes no deben colisionar y se debe controlar por medio de semáforos. No valen esperas activas ni otro tipo de solución.
4. Punto excelencia (para llegar a 10): Incorpora un tren más horizontal y otro tren más en vertical y marca otro punto de conflicto que controles con semáforos.

Haz un *zip* del proyecto o de los proyectos que hayas utilizado para implementar la práctica.

Procura que el código esté lo más comentado posible.

Da las indicaciones que creas necesarias para ponerlo en marcha.

Memoria (por favor, muy breve)

1. Indica qué procesos se utilizan en tu práctica y cuál es la jerarquía.
2. Indica qué se guarda en la memoria compartida. Desde dónde se accede en el padre y desde dónde se accede en los hijos.
3. Indica cómo has inicializado los semáforos y cuándo se utilizan. Indica dónde se hacen los wait y los signal.
4. Conclusiones: Qué has aprendido con esta práctica, qué te ha costado más, cualquier opinión que consideres relevante...