Primer Proyecto: Concu Voley

75.59 - Técnicas de Programación Concurrente I

Objetivo

El objetivo de este proyecto consiste en desarrollar una aplicación que simulará un torneo de *beach voley*.

Reglamento

Las personas llegan, de a una, a una playa donde se desarrolla un torneo de beach voley.
El torneo se realiza en una modalidad algo extraña:

1. Los participantes compiten en parejas en partidos de 2 contra 2.
2. Los partidos se juegan en sets, siendo posible jugar de 3 a 5 sets (en voley no existen los partidos en empate). Cada participante obtiene:
   a) Partido ganado 3 sets a 0 ó 3 sets a 1: 3 puntos para cada participante del equipo ganador, 0 para los perdedores.
   b) Partido ganado 3 sets a 2: 2 puntos para cada participante del equipo ganador, 1 punto para cada participante del equipo perdedor.
3. Cada participante puede competir en hasta k partidos, con la condición de que no puede jugar dos partidos con el mismo compañero.

Los partidos se desarrollan en un predio dentro de la playa con capacidad máxima de M personas. Todas las personas en el predio compiten en el torneo, una vez que completan los k partidos se van, con excepción del caso en el cual no puedan encontrar pareja, en cuyo caso abandonan el predio por su cuenta.

Las personas pueden entrar y salir del predio en cualquier momento. Las personas al formar una pareja se anotan y esperan que se libere una cancha para poder jugar. Las parejas se comienzan a formar sólo luego de que arranque el torneo. El torneo arranca cuando haya 10 o más personas en el predio. Cada partido dura un tiempo aleatorio.

El predio contiene $F \cdot C$ canchas, ubicadas en F filas y C columnas. Cada partido se desarrolla en una cancha al azar, suponiéndolas indistinguibles.

En la playa la marea puede subir o bajar. Hay un sistema de alarmas en el control que se encarga de monitorear el estado de la marea. Al subir hasta cierto punto, suena una alarma y se inhabilitan las F canchas correspondientes a una columna. Al bajar a cierto punto, suena otra alarma y se habilitan nuevamente F canchas. Se estima que la marea no puede cubrir más de $C - 1$ columnas, y que sube y baja habilitando y deshabilitando solo una columna de canchas a la vez.

Si se inhabilitan las canchas, los partidos que allí se desarrollan se cancelan y es como si no se hubiesen jugado.

Hay dos personas encargadas de organizar el torneo, una que lleva un detalle de todos los partidos jugados en el día (parejas y resultado) y otra que se encarga de mantener la tabla de resultados y publicarlos periódicamente en el sitio web de la playa.

Requerimientos Funcionales

Los requerimientos funcionales son los siguientes:

1. Debe poder configurarse sin necesidad de recompilar el código:
   a) Las dimensiones del predio: las F filas y C columnas.
   b) La capacidad máxima de personas (M).
   c) La cantidad máxima de partidos por participante k.
2. Al finalizar el torneo, se debe comunicar quién es el campeón. Será la persona que obtenga mayor cantidad de puntos (serán varias personas, en caso de empate en mayor cantidad de puntos).

Requerimientos no Funcionales

Los siguientes son los requerimientos no funcionales de la aplicación:

1. El proyecto deberá ser desarrollado en lenguaje C o C++, siendo este último el lenguaje de preferencia.
2. La simulación puede no tener interfaz gráfica y ejecutarse en una o varias consolas de línea de comandos.
3. El proyecto deberá funcionar en ambiente Unix / Linux.
4. La aplicación deberá funcionar en una única computadora.
5. El programa deberá poder ejecutarse en "modo debug", lo cual dejará registro de la actividad que realiza en un único archivo de texto para su revisión posterior.
6. Las facilidades de IPC que se podrán utilizar para la realización de este proyecto son las que abarcan la primera parte de la materia, es decir, hasta el primer parcial. Dichas facilidades son:
   a) Memoria compartida
   b) Señales
   c) Pipes y fifos
   d) Locks
   e) Semáforos

   Cualquier otra facilidad queda expresamente excluida para este proyecto.

Tareas a Realizar

A continuación se listan las tareas a realizar para completar el desarrollo del proyecto:

1. Dividir el proyecto en procesos. El objetivo es lograr que la simulación esté conformada por un conjunto de procesos que sean lo más sencillos posible.
2. Una vez obtenida la división en procesos, establecer un esquema de comunicación entre ellos teniendo en cuenta los requerimientos de la aplicación. ¿Qué procesos se comunican entre sí? ¿Qué datos necesitan compartir para poder trabajar?
3. Tratar de mapear la comunicación entre los procesos a los problemas conocidos de concurrencia.
4. Determinar los mecanismos de concurrencia a utilizar para cada una de las comunicaciones entre procesos que fueron detectadas en el ítem 2. No se requiere la utilización de algún mecanismo específico, la elección en cada caso queda a cargo del grupo y debe estar debidamente justificada.
5. Realizar la codificación de la aplicación. El código fuente debe estar documentado.

Entrega

La entrega del proyecto comprende lo siguiente:

1. Informe, se deberá presentar impreso en una carpeta o folio y en forma digital (PDF) a través del campus
2. El código fuente de la aplicación, que se entregará únicamente mediante el campus

La entrega en el campus estará habilitada hasta las 19 hs de la fecha indicada oportunamente.

El informe a entregar debe contener los siguientes items:

1. Detalle de resolución de la lista de tareas anterior.
2. Diagrama que refleje los procesos, el flujo de comunicación entre ellos y los datos que intercambian.
3. Diagramas de clases realizados.
4. Diagrama de transición de estados de un jugador.