Universidad de los Andes
Departamento de Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional
Semestre 2024-10

Caso 1
Manejo de la Concurrencia

En este caso vamos a simular el Juego de la Vida. El juego de la vida es un autómata celular que sirve, entre otras tareas, para simular la evolución de una población a lo largo del tiempo y a partir de una situación inicial definida y de acuerdo con unas reglas de evolución precisas.

Funcionamiento:

En el juego original, el tablero de juego es infinito, pero para nuestro caso utilizaremos una matriz cuadrada de NxN posiciones (N es un dato de entrada). Cada celda debe aplicar las reglas de evolución de acuerdo con el estado de sus celdas vecinas en la matriz. Cada celda tiene dos estados: “viva” o “muerta” y todas evolucionan al mismo tiempo (hacia el siguiente turno) en función de los estados de las celdas vecinas (en el turno actual), siguiendo las siguientes reglas (ajustadas para estas condiciones):

- Nace: Si una celda muerta tiene exactamente 3 celdas vecinas vivas "nace" (es decir, al turno siguiente estará viva).
- Muere: una celda viva puede morir por uno de 2 casos:
    - Sobrepoblación: si tiene más de tres celdas vivas alrededor.
    - Aislamiento: si no tiene celdas vivas alrededor.
- Vive: una celda se mantiene viva si tiene entre 1 y 3 celdas vecinas vivas a su alrededor.

Al inicio se define el número de generaciones (turnos) que se van a simular y, en cada turno, las celdas informan a sus vecinas su estado y cada celda calcula el nuevo estado para el turno siguiente. Para ello, cada celda necesitará enviar la información de su estado a todas sus vecinas y, a su vez, recibir el estado de todas sus vecinas para actualizar el suyo propio.

El comportamiento de cada celda se debe implementar con threads y su comunicación por medio de buzones, teniendo en cuenta las siguientes reglas:

Todas las celdas deben evolucionar por turnos: en un momento todas las celdas se encuentran en la misma generación o turno (ninguna puede avanzar hasta que todas completen el cálculo de su estado para el turno en el que se encuentran).

- El thread principal tendrá la responsabilidad de mostrar el resultado final de la evolución del estado de todas las celdas del tablero una vez se han simulado el número de generaciones especificadas.
- Cada celda tendrá asociado un buzón con capacidad de x mensajes. x se calcula como el número de fila de la celda en el tablero más 1 (p.e. la celda ([3][2] tiene un buzón con capacidad de 4 mensajes).

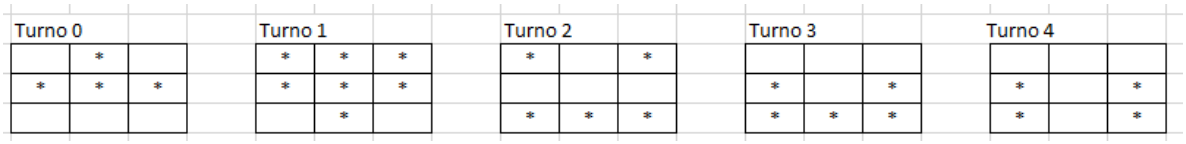
- Al enviar el estado a una celda vecina se debe hacer sobre el buzón del destinatario y con espera pasiva.
- Al recibir el estado de una celda vecina, se debe hacer sobre el buzón propio con espera semi-activa. (Nota: sleep no es apropiado para espera semi-activa).
- La comunicación con el thread principal la puede implementar como mejor lo considere, pero en el informe debe justificar su selección.

El estado inicial se debe cargar de un archivo que se especifica por consola. El número de generaciones a simular también debe especificarse por consola. La primera línea del archivo debe especificar el número de filas (y columnas) del tablero, las siguientes líneas especifican el estado de cada celda: true significa viva, false en caso contrario. A continuación, un ejemplo de archivo de entrada y la secuencia de evolución que generaría:

Archivo de entrada:

3
false,true,false
true,true,true
false,false,false

Evolución de tableros



Condiciones de Entrega:

- En un archivo .zip entregar el código fuente del programa, y un documento Word explicando el diseño (diagrama de clase) y funcionamiento del programa, así como la validación realizada. En particular, para cada pareja de objetos que interactúan (para cada pareja thread-thread o thread-objeto), explique cómo se realiza la sincronización, así como el funcionamiento global del sistema. El nombre del archivo debe ser: caso1_*login1_login2_login3*.zip
- El trabajo se realiza en los grupos definidos en el curso para este caso. No debe haber consultas entre grupos.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes).
- Habrá una coevaluación del trabajo en grupo. Cada miembro del grupo evaluará a sus dos compañeros y las notas recibidas por un estudiante ponderarán la nota final de caso para ese estudiante.
- En el parcial se incluirá una pregunta sobre el desarrollo de alguna de las funcionalidades del caso.
- El proyecto debe ser entregado por Bloque Neón por uno solo de los integrantes del grupo. **Al comienzo del documento Word, deben estar los nombres y carnés de los integrantes del grupo**. Si un integrante no aparece en el documento entregado, el grupo podrá informarlo posteriormente, sin embargo, habrá una penalización: la calificación asignada será distribuida (dividida de forma equitativa) entre los integrantes del grupo.

- Tenga en cuenta las reglas establecidas en el programa del curso sobre la obligatoriedad de trabajar con el grupo asignado.
- **La solución se debe entregar por BloqueNeon a más tardar el 19 de febrero, 2024 a las 23:55 (p.m.)**

Cronograma Propuesto:

5 feb.:	Lectura del enunciado
6 feb. :	Entrega del contrato de trabajo en grupo
10 feb. :	Arquitectura general de la solución (diagrama de clases) y estrategia para los procesos
15 feb.:	Implementación + pruebas
19 feb.:	Informe y entrega
22. feb:	Realizar Coevaluación