

Trabajo Práctico 1: Sistemas Operativos

Farizano, Juan Ignacio

Mellino, Natalia

1. Motivación

El Juego de la Vida de Conway es un autómata celular creado por el matemático John Horton Conway en 1970. Se trata de un juego de cero jugadores, por lo tanto, su evolución está determinada por el estado inicial y no necesita ninguna entrada de datos posteriormente.

El tablero de juego es una grilla con cuadrados que se extiende en el infinito en todas las direcciones, por lo tanto, cada célula tiene 8 células vecinas. Dichas células tienen sólo dos estados: vivas o muertas. El juego evoluciona en cada ciclo teniendo en cuenta las siguientes reglas:

- Una célula muerta con exactamente 3 vecinos vivos, revive.
- Una célula viva con 2 o 3 células vecinas vivas, sigue viva. En cualquier otro caso esta muere por soledad o superpoblación.

Así, en cada ciclo, surge una nueva generación de células.

2. Implementación

Para nuestra implementación en C, como no podemos disponer de un tablero infinito, las fronteras del tablero se comparten, es decir, el borde superior del tablero es el borde inferior del tablero y el borde izquierdo del tablero es el borde derecho.

Los estados de las células están simbolizados con 'X' y 'O', donde 'X' significa que la célula está viva y 'O' que la célula está muerta.

El tablero inicial es leído desde un archivo donde en la primera línea se detalla el número de ciclos que se desean realizar, el número de filas y el número de columnas, en ese orden. Si tenemos n filas, en las siguientes n líneas del archivo estarán las filas del tablero con el estado inicial de las células.

Mediante la utilización de hilos distribuiremos el trabajo que conlleva recorrer el tablero y calcular los estados de las células: utilizaremos tantos hilos como unidades de procesamiento haya disponibles. Si la cantidad de hilos llegara a ser mayor que la cantidad de filas, cada hilo trabajará con una sola fila, reduciendo así la cantidad de estos. En caso contrario, las filas que le corresponden a cada hilo son distribuidas de la forma más equitativa posible. En la mayoría de los casos puede que algunos hilos tengan, a lo sumo, una fila más que el resto de los hilos ya que no se pudo asignar a cada hilo la misma cantidad de filas.

Una vez que tenemos la carga del tablero distribuida, cada hilo realizará lo siguiente:

- Por cada fila recorremos las columnas, calculando cuál será el estado siguiente de cada célula. Para esto utilizaremos una función auxiliar que toma una sola posición i, j del tablero y revisa sus vecinos; luego aplica las reglas y decide si la célula vive o muere.

- Una vez que un hilo terminó de calcular el estado siguiente de sus células, este debe esperar a que los demás hilos terminen de hacer lo mismo, por lo tanto utilizaremos una barrera, que hará que dicho hilo espere a que terminen todos los demás el ítem anterior y luego continúe con la siguiente sección del código.
- Cuando todos los hilos terminaron, debemos pasar el nuevo tablero a otro ya que necesitaremos siempre tener el anterior para poder crear el siguiente. Cada hilo pasará sus filas a la sección correspondiente del tablero y nuevamente deberá esperar a que los demás hilos terminen de escribir sus filas, para así poder pasar a la siguiente iteración donde calcularemos la siguiente generación de células. En este paso también utilizaremos una barrera, así todos los hilos se esperan.
- Este proceso es repetido tantas veces como ciclos haya ingresado el usuario. Una vez finalizados los ciclos, el tablero final es escrito en un archivo de salida y el usuario podrá verlo.

3. Compilación y ejecución

Para compilar basta con poner en la terminal el comando:

```
$ make
```

y luego para ejecutar simplemente ponemos:

```
$ ./simulador.o fileName.game
```