



Proyecto 2 (12%) El Juego de la Vida

Objetivos

- Familiarizarse con uso de procesos y de hilos.
- Desarrollar una aplicación con procesos cooperantes.
- Diseñar esquemas de comunicación.
- Resolver problemas de región crítica.

Descripción general del problema

El Juego de la Vida fue desarrollado por J.H. Conway en 1970. Aunque parezca un juego, realmente es un modelo de una población de células que interactúan con su entorno. En el núcleo del modelo se encuentra una cuadrícula que representa el "mundo".

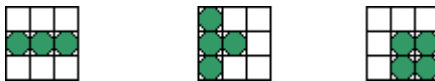
Cada cuadrado puede estar ocupado por una célula viva o una muerta. La ejecución del juego produce una serie de generaciones donde cada una de ellas es un mundo derivado del anterior.

Las células interactúan entre si usando las siguientes reglas:

- Una célula muerta con exactamente 3 vecinos vivos se convierte en una célula viva (nacimiento)



- Una célula viva con 2 ó 3 vecinos vivos permanece viva (supervivencia)



- En todos los demás casos, una célula muere o permanece muerta (sobrepoblación o soledad)



Para mayor información acerca del juego revisen la página: [Conway's Game of Life](http://conwaylife.com/) .

Implementación

Se desea que Ud. implemente el Juego de la Vida usando un esquema maestro/esclavos. El

proceso/hilo maestro será el primero y los esclavos serán los creados por el maestro. La comunicación entre procesos será a través de pipes y entre hilos usando memoria compartida respectivamente. La cuadrícula que representa el mundo será repartida entre los distintos esclavos.

El juego será invocado de la siguiente forma:

```
$ juego_p <n_procesos> <n_generaciones> <n_visualizacion> <archivo>
$ juego_h <n_hilos> <n_generaciones> <n_visualizacion> <archivo>
```

donde

n_procesos	número de procesos esclavos que realizarán la simulación
n_hilos	número de hilos esclavos que realizarán la simulación
n_generaciones	número de generaciones o mundos a ser generados
n_visualizaciones	cada cuántas generaciones se desea ver el mundo generado
archivo	nombre del archivo que contiene el mundo inicial

El maestro crea n esclavos y lee de <archivo> el tamaño del mundo a simular (número de filas y columnas).

Cada esclavo lee de <archivo> la porción del mundo que le corresponde simular y almacena dicha información. Note que ellos no deben leer la información que le corresponde a otro esclavo. A cada uno le corresponde un número *equitativo* de filas que dependerá del tamaño de la cuadrícula y del número de esclavos involucrados.

Cada etapa de la simulación consta de los siguientes pasos:

1. **Intercambio:** Cada esclavo envía y recibe de sus vecinos la información acerca de las filas de la cuadrícula que le interesan. Esta etapa involucra comunicación
 - entre proceso que deberá ser implementada por medio de pipes
 - entre hilos que deberá ser implementada por medio de memoria compartida
2. **Generación:** Se actualizan las celdas de la cuadrícula dependiendo del estado actual usando las reglas arriba mencionadas

La simulación se repite <n_generaciones> y cada <n_visualizaciones> el padre debe mostrar el mundo en una forma clara.

Formato de <archivo> con el mundo Inicial

- La primera línea del archivo contiene *nfilas ncol*

<i>nfilas</i>	entero que indica el número de filas en la cuadrícula
<i>ncol</i>	entero que indica el número de columnas en la cuadrícula
- Las siguientes *nfilas* líneas del archivo tienen la forma:

$$C_{i0} C_{i1} C_{i2} \dots C_{i\text{ncol}-1}$$

donde C_{ij} es igual a cero si hay una célula muerta en la celda ij , y es igual a 1 si hay una célula viva

Su aplicación debe cumplir con las siguientes restricciones:

- Los procesos deben comunicarse entre sí usando pipes no nominales o señales.
- Los hilos deben comunicarse entre sí usando memoria compartida o señales.
- Los hilos pueden usar semáforos, los procesos no pueden usarlos.

Informe

Realice un informe en el que muestre y justifique

- los problemas de condición de carrera encontrados y cómo fueron resueltos.
- El esquema de comunicación entre los procesos/hilos

Recomendaciones

Entender bien el problema antes de diseñar su propuesta.

- Lea la información dada por las páginas de manual del sistema sobre las funciones que debe usar.
- Diseñe **toda** su solución antes de comenzar a programar.
- Programe en **forma estructurada** esto le permitirá **re-usar** código entre las dos soluciones.
- Vaya documentado su código a medida que lo vaya generando.
- Trabaje en forma ordenada!
- Si tiene alguna duda, con tiempo aclárela directamente con su profesor.
- **Siga la guía de estilo publicada.**

Entrega del proyecto:

Realización: Individual

Digital: Hasta las 11:59pm del sábado de semana 10.

Cada estudiante debe colocar en un archivo que con nombre carnet.tar.gz:

- los fuentes de su tarea (archivos .c y .h). Estos deben estar bien identificados, documentados y estructurados. Este conjunto será usado para compilar y correr su proyecto.
- Un archivo de nombre *codigo.pdf* con los fuentes de su tarea. Exactamente la misma versión que los .c y .h. Éste será usado para evaluar el código de su tarea.
- El Makefile que pueda compilar/enlazar su aplicación.
- Un informe explicando manejos de regiones críticas y las gráficas de desempeño.