

Un Modelo de Referencia para la Generación de Informes

José Ignacio Sbruzzi, *Padrón Nro. 97.452*

jose_sbruzzi@hotmail.com

Leandro Huemul Desuque, *Padrón Nro. 00.000*

dirección de e-mail

Grupo Nro. 0 - 2do. Cuatrimestre de 2006

66.20 Organización de Computadoras

Facultad de Ingeniería, Universidad de Buenos Aires

Resumen

Se desarrolló un programa en C que simula el Juego de la Vida de Conway, habiéndose implementado una parte específica en assembly MIPS. El objetivo del presente trabajo fue además familiarizarse con el entorno GXEmul y con LaTeX.

1. Introducción

Inicialmente, se implementó en C el Juego de la Vida de Conway. El Juego de la Vida es un autómata celular descrito por John Conway en 1970 ???. Inicialmente fue publicado como un juego matemático recreacional. El juego es "de cero jugadores."^{en} el sentido de que no es necesario propiamente jugar, sino que lo único que hace el jugador es decidir la configuración inicial y observar cómo evoluciona el sistema. Se rige por reglas sencillas: las células nacen, sobreviven o mueren dependiendo de la cantidad de vecinos vivos que tienen, tal como descrito en el enunciado de este trabajo práctico. Así, es un sistema simple, y una buena elección si el objetivo final no es construir un sistema elaborado sino familiarizarse con el software necesario para ello.

Posteriormente, se reprogramó una parte del programa en assembly MIPS, para esto se utilizó el emulador GXEmul. GXEmul emula arquitecturas de computadoras y permite correr sistemas operativos sobre ellos ???. De esta manera, se compiló y se corrió el programa en assembly MIPS pese a no disponer de un equipo MIPS.

2. desarrollo

2.1. Documentación del código C

2.1.1. vecinos

`vecinos` es una función que, a partir de la matriz y de una celda, determina cuántos vecinos vivos (es decir, casilleros ocupados) tiene una celda. De esta manera, la función `siguienteMatriz` puede definir dónde suceden nacimientos, muertes y supervivencias.

2.1.2. siguienteMatriz

`siguienteMatriz` genera, a partir de una matriz (es decir, un estado), la matriz que le sigue, según las reglas del juego.

2.1.3. El resto acá...

asdasd

2.2. Documentación del código assembly

...

2.3. Dificultades

La construcción de una imagen en formato PBM presentó algunas dificultades ya que se usó PBM en vez de plain PBM, y en la descripción del formato PBM no había ejemplos de archivos PBM (sí los había de plain PBM). Un punto relativamente problemático del formato PBM es que cada pixel está asociado a un bit. Para evitar esta dificultad, se decidió que cada celda correspondería a un cuadrado de 8x8 pixeles, simplificando el fragmento del programa que guarda las imágenes.

3. Resultados

3.1. Medidas de tiempo de ejecución

3.2. Corridas de prueba

A continuación se detalla el resultado de las corridas de prueba de glider, pento y sapo para 10 operaciones en una matriz de 20 por 20 tal como pedido.

3.2.1. Glider

Glider es una configuración del autómata que se mueve por la pantalla sin destruirse: atravieza un ciclo de cuatro estados, al cabo de los cuales regresa a la configuración inicial, pero desplazada. En la figura ?? se muestra la configuración inicial del glider junto con 10 iteraciones. El glider se mueve, en cada ciclo, una celda hacia abajo y una celda a la derecha, como puede notarse en la figura ??.

↗	↖	↘	↙
↗	↖	↘	↙
↗	↖	↘	

Figura 1: Desde la esquina izquierda superior: 1:Estado inicial descrito en el archivo glider. 2 a 11: Iteraciones 1 a 10



Figura 2: El movimiento del glider: se superponen la iteración 4 y 8

3.2.2. Pento

Pento es un patrón que se estabiliza luego de cierta cantidad de iteraciones. El resultado final varía según el tamaño de la matriz. Para una matriz de 20 por 20, se muestra la configuración inicial, las primeras 10 iteraciones y el resultado final estable, alcanzado en la iteración 60, en la figura ?? . En la imagen ?? puede verse la configuración final de Pento para una matriz de otro tamaño.



Figura 3: Desde la esquina superior izquierda: 1: estado inicial de Pento, 2 a 11: primeras 10 iteraciones, 12: estado final luego de 60 iteraciones. El tamaño de la matriz es 20x20



Figura 4: La configuración final de Pento en una matriz de 30 por 30.

3.2.3. Sapo

El patrón Sapo, al igual que Glider, atravieza un ciclo, pero de 2 estados (no de 4), y no se desplaza. En la figura ?? se muestran el estado inicial y cuatro iteraciones de Sapo. No se muestran más iteraciones porque los estados se repiten constantemente.

		
estado inicial	Iteración 1	Iteración 2
		
	Iteración 3	Iteración 4

Figura 5: La configuración final de Pento en una matriz de 30 por 30.

4. Conclusiones

HACERRRRRRR Se presentó un modelo para que los alumnos puedan tomar como referencia en la redacción de sus informes de trabajos prácticos.

Referencias

- [1] Gardner, Martin. “Mathematical Games - The fantastic combinations of John Conway’s new solitaire game ‘life’ ” Scientific America, 223. pp. 120-123. ISBN 0-89454-001-7. Archivado del original en https://web.archive.org/web/20090603015231/http://ddi.cs.unipotsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelif/ConwayScientificAmerican.htm . Consultado en septiembre 2016.
- [2] Sitio web de GXemul <http://gxemul.sourceforge.net/> . Consultado en septiembre 2016.