FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Organización de Computadoras (66.20)

Conjunto de Instrucciones MIPS

Cristian González 94719

Darius Maitia 95436

Índice

1. Introducción

El “Juego de la Vida”de Conway es un autómata celular, diseñado por el matemático británico John Conway en 1970. Se trata básicamente de una grilla en principio infinita, en cada una de cuyas celdas puede haber un organismo vivo, en cada iteración del juego, se va redefiniendo si un organismo (una celda de la grilla) cambia de estado (viva o muerta).
Este cambio de estado depende de sus vecinos (las posiciones alrededor de este) que define con ciertas reglas. Para eso se nos pide implementar una función `vecinos()` que observa los vecinos y nos devuelve la cantidad de vecinos “vivos”.
Esta funcionalidad será implementada en Assembler MIPS 32, pero también nuestro programa contara con su versión implementada totalmente en C.

1.1. Instrucciones

Acá se describe los pasos para ejecutar el juego. Se cuenta con los archivos:

- JuegoDeLaVida.c
- vecinos.S

Los primeros casos consiste en compilar ambos archivos:

- `gcc -ggdb -Wall -c vecinos.S`
- `gcc -ggdb -Wall -c JuegoDeLaVida.c`

Se procede a linkear y crear un ejecutable.

- `gcc vecinos.o JuegoDeLaVida.o -o conway`

Para la ejecución del programa se debe realizar la siguiente linea:

```
conway i M N inputfile [-o outputprefix]
```

1.2. Código Assembler

```
#include <mips/regdef.h>

        .align 2                #alinear por tandas de 2^2=4 bytes
        .text
        .ent vecinos
        .globl vecinos

        ################VARIABLES INTERNAS##################
        #int columnaFinal ---  s1
        #int filaFinal    ---  s2
        #columnaInicial   ---  s3
        #filaInicial      ---  s4
        #vecinos          ---  s5
        #y =  s6
        #x =  s7
        ####################################################

        ################PARAMETROS##########################
        #unsigned char *a       ---  a0

        #unsigned int columna  ---  a1
        #unsigned int fila     ---  a2
        #unsigned int cantFilas---  a3
        #unsigned int cantCols --- 48( $fp) ->  t7
        ####################################################

vecinos:subu  sp, sp,32

        sw      $fp,28(sp); #Guardo la direccion de memoria de fp en memoria
        sw      gp,24(sp);  #
        move    $fp,sp;     #uso fp,

        ######INICIALIZAR VARIABLES######
        subu  s4, a2,1          #filaInicial = fila - 1;
        subu  s3, a1,1          #columnaInicial = columna - 1;
        addu  s2, a2,1          #filaFinal = fila + 1;
        addu  s1, a1,1          #columnaFinal = columna + 1;
        add  s5, zero, zero     #vecinos = 0;
        lw  t7,48($fp)          #cantCols ->  t7

        ######ACOTAR POSICIONES #########
# Referencias : https://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Mips/cond.html
#               Patterson: apendice A: A-64 "Branch Instructions";
#               Patterson: capitulo 2.7 "Instructions for making decisions"

 IF1:   bgez  s4, IF2                   #if (filaInicial < 0) {
        add  s4, zero, zero             #  filaInicial = 0;
                                        #}
                                        # Resumen:
                                        # "Si filaInicial es mayor o igual a cero
                                        # saltar a IF2 y sino ejecutar cuerpo"

 IF2:   bgez  s3, IF3                   #if (columnaInicial < 0) {
```

```
        add   s3, zero, zero           #  columnaInicial = 0;
                                       #}

 IF3:   subu  t0, a3,1                 #t0 = cantFilas - 1
        ble   s2, t0,IF4               #if (filaFinal > cantFilas - 1) {
        add   s2, zero, t0             # filaFinal = cantFilas - 1;
                                       #}
                                       # Resumen:
                                       # "Si filaFinal es menor o igual a cantFilas - 1
                                       # saltar a IF4 y sino ejecutar cuerpo"

 IF4:   subu  t0, t7,1                 #t0 = cantCols - 1
        ble   s1, t0,LOOP              #if (columnaFinal > cantCols - 1){
        add   s1, zero, t0             #  columnaFinal = cantCols - 1;
                                       #}

 LOOP: addi  s6, s4,-1                 #s6 = filaInicial - 1 (porque luego incremento
                                       #cuando comienza el ciclo y arranca en filaInici
 FOR1:  addu s6, s6,1                  #y++
        bgt  s6, s2,RETURN             #si y > filaFinal ir a FIN
        addu  s7, s3,-1                # s7 = columnaInicial - 1 (mismo motivo que ante
        j FOR2

 IF5:   multu  s6, t7                  #y*cantCols
        mflo   t1                      #almacenar los 32 bits menos significativos
        add  t1, t1, s7                #y*cantCols + x
        addu t3,t1,a0                  #sumo el desplazamiento mas la direccion que hay
        lbu   t2,(t3)                  #a[y * cantCols + x]
        add  s5, s5, t2                #vecinos += t2
        j FOR2

FOR2:   addi s7, s7,1                  #x++
        bgt  s7, s1,FOR1               #si x > columnaFinal ir a FOR1
                                       #en esta parte cabe aclarar que por DeMorgan
                                       #!(x == columna && y == fila) equivale a (x!=colu

        bne  s7, a1,IF5
        beq  s6, a2,FOR2               #No se cumple ninguna de las 2 condiciones
        j IF5

RETURN: addu v0,s5,zero                #coloco en ra lo que hay en s5
        lw $fp,28(sp)
        addi sp,sp,32                  #restauro el valor del stack pointer
        j ra                           #retorno el valor
        .end vecinos
```

2. Documentacion de Corridas

Contenido del archivo `pento`:

```
cat pento
3 5
3 6
4 4
4 5
5 5
```

Contenido del archivo `sapo`:

```
cat sapo
5 3
5 4
5 5
4 4
4 5
4 6
```

Contenido del archivo `glider`:

```
cat glider
5 3
5 4
5 5
3 4
4 5
```

2.1. Corrida glider

Comando ejecutado:

```
./conway 10 20 20 glider -o salida
```

Salida stdin:

```
Grabando salida_1.pbm
Listo
Grabando salida_2.pbm
Listo
Grabando salida_3.pbm
Listo
Grabando salida_4.pbm
Listo
Grabando salida_5.pbm
Listo
Grabando salida_6.pbm
Listo
Grabando salida_7.pbm
Listo
Grabando salida_8.pbm
Listo
Grabando salida_9.pbm
Listo
Grabando salida_10.pbm
Listo
```

Archivos:

```
cat salida_1.pbm
P1 20 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
cat salida_5.pbm
P1 20 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
cat salida_10.pbm
P1 20 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

2.2. Corrida sapo

Comando ejecutado:

```
./conway 10 20 20 sapo -o salida_sapo
```

Salida stdin:

```
Grabando salida_sapo_1.pbm
Listo
Grabando salida_sapo_2.pbm
Listo
Grabando salida_sapo_3.pbm
Listo
Grabando salida_sapo_4.pbm
Listo
Grabando salida_sapo_5.pbm
Listo
Grabando salida_sapo_6.pbm
Listo
Grabando salida_sapo_7.pbm
Listo
Grabando salida_sapo_8.pbm
Listo
Grabando salida_sapo_9.pbm
Listo
Grabando salida_sapo_10.pbm
Listo
```

Archivos:

```
cat salida_1.pbm
P1 20 20
P1 20 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
   cat salida_sapo_5.pbm
P1 20 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
cat salida_sapo_10.pbm
P1 20 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

2.3. Corrida pento

Comando ejecutado:

```
./conway 10 20 20 pento -o salida_pento
```

Salida stdin:

```
Grabando salida_pento_1.pbm
Listo
Grabando salida_pento_2.pbm
Listo
Grabando salida_pento_3.pbm
Listo
Grabando salida_pento_4.pbm
Listo
Grabando salida_pento_5.pbm
Listo
Grabando salida_pento_6.pbm
Listo
Grabando salida_pento_7.pbm
Listo
Grabando salida_pento_8.pbm
Listo
Grabando salida_pento_9.pbm
Listo
Grabando salida_pento_10.pbm
Listo
```

Archivos:

```
cat salida_1.pbm
P1 20 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
cat salida_pento_5.pbm
P1 20 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
   cat salida_pento_10.pbm
P1 20 20
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

En caso de que no especifique archivo de salida:

```
./conway 10 20 20 pento
Grabando pento_1.pbm
Listo
Grabando pento_2.pbm
Listo
Grabando pento_3.pbm
Listo
Grabando pento_4.pbm
Listo
Grabando pento_5.pbm
Listo
Grabando pento_6.pbm
Listo
Grabando pento_7.pbm
Listo
```

```
Grabando pento_8.pbm
Listo
Grabando pento_9.pbm
Listo
Grabando pento_10.pbm
Listo
```

3. Definiciones del Juego

- Se realizo el juego de tal forma que si un vecino de la posición i,j está fuera de rango de nuestra grilla, este vecino está muerto.

4. Stack

```
|//////////////////////////////|
|/////////STACK DE LA//////////|
|///////////FUNCION////////////|
|//////////ANTERIOR////////////|
|——————————————————————————————|
|              fp              |
|——————————————————————————————|
|              sp              |
|——————————————————————————————|
|                              |
|——————————————————————————————|
|                              |
|——————————————————————————————|
|                              |
|——————————————————————————————|
|                              |
|——————————————————————————————|
|                              |
|——————————————————————————————|
```

5. Conclusiones

En este trábajo practico se llego a qué Assembly representa un interesante recurso de programación, en el que se hace necesario el conocimiento del funcionamiento interno del procesador así como de el armado del stack entre funciones llamadas y llamadoras.
El cómo una función comparte sus recursos - en este caso el quinto argumento - por medio del stack. También sobre la preservación de una estado correcto cómo la preservación de valores de ciertos registros, como la dirección de memoria de stack pointer.