

---

# UBA FACULTAD DE INGENIERÍA

66.20 Organización de Computadoras

## Trabajo Practico

2<sup>do</sup> Cuatrimestre 2016

### Grupo 1

**Integrantes:**

Federico Rodriguez	93336
fedelonghi@hotmail.com	
Ezequiel Dufau	91985
fedelonghi@hotmail.com	
Pablo Ascarza	89711
fedelonghi@hotmail.com	

# Índice

<b>1. Enunciado</b>	<b>2</b>
<b>2. Introducción</b>	<b>2</b>
<b>3. Utilización</b>	<b>2</b>
3.1. Compilación y Ejecución . . . . .	2
3.2. Documentación de Parámetros . . . . .	2
3.3. Documentación de Errores . . . . .	3
3.4. Algunas Aclaraciones . . . . .	3
3.5. Ejemplos de Uso . . . . .	3
<b>4. Implementación</b>	<b>4</b>
<b>5. Documentación Diseño e Implementación</b>	<b>4</b>
5.1. Diagrama del Stack de Vecinos . . . . .	4
<b>6. Corridas de Prueba</b>	<b>4</b>
<b>7. Conclusiones</b>	<b>4</b>
<b>8. Código Fuente</b>	<b>4</b>

## 1. Enunciado

## 2. Introducción

El presente trabajo tiene como objetivo familiarizarse con el conjunto de instrucciones *MIPS-32* y el concepto de ABI. Para el cumplimiento de este objetivo se desarrolló un programa que simula el “*Juego de la Vida*” de *Conway* según lo detallado en el enunciado.

La implementación se realizó en el lenguaje de programación C. Además se desarrolló una porción en assembler *MIPS-32* que luego será detallada.

El programa fue desarrollado para correr sobre una plataforma *NetBSD* / *MIPS-32* mediante el emulador *GXEmul*.

## 3. Utilización

El programa fue implementado para que cumpliera con los requisitos pedidos por el tp. En las siguientes secciones se detallarán los diferentes aspectos para la ejecución del programa.

### 3.1. Compilación y Ejecución

1. Descargar el archivo fuente “conway.c”
2. Compilar el archivo (por ejemplo con gcc:  

```
gcc -Wall -c ‘conway.c’  
gcc -Wall -o ‘conway’ ‘conway.c’
```

)
3. Ejecutar el programa con: `./conway i M N input [-o output]`

### 3.2. Documentación de Parámetros

- `i` es la cantidad de simulaciones que queremos realizar.
- `M` y `N` especifican las dimensiones de la matriz sobre la cual queremos simular.
- `input` es el nombre del archivo que contiene las coordenadas de las celdas vivas e identifica el estado inicial de la matriz.
- `-o` es un parámetro opcional que especifica que se utilizará el nombre `output` como prefijo de los nombres de los archivos pbm generados. En caso de no existir este parámetro tomara como prefijo `input`.
- `-V` o `--version` muestra la versión del programa.
- `-h` o `--help` muestra la ayuda.

### 3.3. Documentación de Errores

A continuación se detallan los errores y su significado:

- **Actions Count must be a positive integer:** El primer parámetro tiene que ser un entero positivo.
- **Number of rows must be a positive integer:** El número de filas tiene que ser un entero positivo.
- **Number of columns must be a positive integer:** El número de columnas tiene que ser un entero positivo.
- **Invalid parameter:** Para el caso de `-V`, `-h` y `-o`. El parámetro no coincide con estos valores (o sus equivalentes).
- **Wrong number of parameters:** Hay parámetros de mas o de menos (se pasó un número de parámetros distinto a 1 o 6).
- **Error while opening input file:** No se pudo abrir el archivo de entrada.
- **Error while opening output file: [nombre\_de\_archivo]:** No se pudo abrir el archivo de salida con el nombre *nombre\_de\_archivo*

### 3.4. Algunas Aclaraciones

- Las imágenes pbm generadas se guardan en la carpeta imágenes.
- Todos los errores se imprimen directamente a `stderr`.

### 3.5. Ejemplos de Uso

Para ver la documentación:

```
./conway -h
```

Para ver la informacion sobre la version:

```
./conway -V
```

Para generar un tablero de  $100 \times 50$  a partir del archivo `glider` y realizar 20 iteraciones:

```
./conway 20 100 50 glider
```

Los archivos pbm generados por el comando anterior seran nombrados de la forma: “glider\_N.pbm”.

Para generar un tablero de  $20 \times 30$  a partir del archivo **pento**, realizar 10 iteraciones y que los archivos pbm generados tengan como prefijo el nombre **jorge**:

```
./conway 10 20 30 pento -o jorge
```

Los archivos pbm generados por el comando anterior seran nombrados de la forma: “jorge\_N.pbm”.

## **4. Implementación**

## **5. Documentación Diseño e Implementación**

### **5.1. Diagrama del Stack de Vecinos**

## **6. Corridas de Prueba**

## **7. Conclusiones**

## **8. Código Fuente**