

# Trabajo Práctico 1:

## MIPS

Duhalde, Agustín, *Padrón Nro. 97.063*  
agusnduha@gmail.com

Suárez, Emiliano, *Padrón Nro. 78.372*  
emilianosuarez@gmail.com

Fecha Entrega: 10/05/2016

1er. Cuatrimestre de 2016  
66.20 Organización de Computadoras – Práctica Martes  
Facultad de Ingeniería, Universidad de Buenos Aires

## Abstract

Se implementó parte del programa anterior en lenguaje **Assembly MIPS**, que multiplica matrices cuadradas de números reales, representados en punto flotante de doble precisión.

## 1 Introducción

En este Trabajo Práctico se pretende familiarizarse con el lenguaje ensamblador MIPS.

Se utilizará *GXemul* para simular una máquina *MIPS* corriendo una versión reciente del sistema operativo *NetBSD*.

El programa implementado, muestra por *stdout* u *output file*, el resultado de multiplicar dos matrices cuadradas (puede ser mas de una operación), previamente leídas por *stdin* o *input file*.

## 2 Diseño e Implementación

Se debió respetar la interfaz del TP anterior, tanto para la estructura de las matrices, como para la implementación de ciertos métodos. Se implemento en MIPS las funciones `print_matrix` y `matrix_multiply`

## 3 Comando para compilar el programa

Para compilar el programa se debe abrir una terminal en la carpeta donde están alojados los archivos fuentes (*src/*) y se ejecuta el siguiente comando:

```
../src$ make
```

Para generar el ejecutable *tp0*.

**make:** se encargara de compilar los archivos generando el ejecutable.

Con el Makefile pudimos integrar la parte del codigo programda en MIPS con la parte programada en C.

## 4 Casos de Prueba

Algunos de los casos de pruebas realizados, pueden observarse a continuación:

Se creó un archivo *Makefile* que permite compilar tanto la versión en *MIPS*, como en la versión implementada completamente en *C*.

En el siguiente caso de prueba, se puede observar el contenido del archivo de entrada *varias.txt* y la salida guardada en el archivo *out.txt* (previamente vacío), utilizando los operadores *j* y *l*, respectivamente:

```

root@:~/2016/tp0# cat varias.txt
3 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
3 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6
2 1 2 3 4 5 6 7 8
2 1 1 1 1 1 1 1 1
root@:~/2016/tp0# cat out.txt
root@:~/2016/tp0# ./tp0 < varias.txt > out.txt
root@:~/2016/tp0# cat out.txt
3 84 90 96 201 216 231 318 342 366
3 15 15 15 30 30 30 45 45 45
2 19 22 43 50
2 2 2 2 2

```

En esta prueba, el programa recibe por *stdin* la salida del comando *cat example.txt*, a través del pipe (—). La salida, es almacenada en *out.txt*:

```

root@:~/2016/tp0# cat example.txt
2 1 2 3 4 1 2 3 4
3 1 2 3 4 5 6.1 3 2 1 1 0 0 0 1 0 0 0 1
root@:~/2016/tp0# cat out.txt
root@:~/2016/tp0# cat example.txt | ./tp0 > out.txt
root@:~/2016/tp0# cat out.txt
2 7 10 15 22
3 1 2 3 4 5 6.1 3 2 1
root@:~/tp/tp1/src# cat prueba2.txt
3 1 -2 3 4 5 0.6 7 8 9 10 11.2 12 -13 -14 -15 16 17 18
root@:~/tp/tp1/src# cat prueba2.txt | ./tp1 > out.txt
root@:~/tp/tp1/src# cat out.txt
3 84 90.2 96 -15.4 -15 -16.2 110 119.4 126

```

## 5 Conclusiones

El presente trabajo permitió la familiarización con las herramientas de compilación de código C y código assembly en un entorno que emula la arquitectura MIPS 32. Fue un buen primer contacto con el lenguaje ensamblador MIPS y con NetBSD. Nos ayudó a conocer la relación entre C y Assembly.

### 5.1 Bibliografía

## References

- [1] *The NetBSD project*  
<http://www.netbsd.org/>
- [2] *MIPS*  
[https://en.wikibooks.org/wiki/MIPS\\_Asembly](https://en.wikibooks.org/wiki/MIPS_Asembly)