

Universidad de Buenos Aires - FIUBA  
66.20 Organización de Computadoras  
Trabajo práctico 1: conjunto de instrucciones MIPS  
2º cuatrimestre de 2016

\$Date: 2016/10/02 21:23:43 \$

## 1. Objetivos

Familiarizarse con el conjunto de instrucciones MIPS y el concepto de ABI, extendiendo un programa que resuelva el problema descrito en la sección 4.

## 2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

## 3. Requisitos

El trabajo deberá ser entregado personalmente, en la fecha estipulada, con una carátula que contenga los datos completos de todos los integrantes, un informe impreso de acuerdo con lo que mencionaremos en la sección 5, y con una copia digital de los archivos fuente necesarios para compilar el trabajo.

## 4. Descripción

Se trata de un modificar un programa que dibuje el conjunto de Julia y sus vecindades introducido en el *TP0* [1], en el cual la lógica de cómputo del fractal deberá tener soporte nativo para MIPS32 sobre NetBSD/pmax.

El código fuente con la versión inicial del programa, se encuentra disponible en [2]. El mismo deberá ser considerado como punto de partida de todas las implementaciones.

### 4.1. Soporte para MIPS

El entregable producido en este trabajo deberá implementar la lógica de cómputo del fractal en assembly MIPS32, con soporte nativo para NetBSD/pmax.

Para ello, cada grupo deberá tomar el código fuente de base para este TP, [2], y reescribir la función `mips32_plot()` sin cambiar su API. Esta función está ubicada en el archivo `mips32_plot.c`.

## 4.2. Casos de prueba

El informe trabajo práctico deberá incluir una sección dedicada a verificar el funcionamiento del código implementado. Para ello, será necesario escribir pruebas orientadas a probar el programa completo, ejercitando los casos más comunes de funcionamiento, los casos de borde, y también casos de error.

## 4.3. Compilación

El código fuente provisto por la cátedra provee los makefiles necesarios para compilar el ejecutable a partir de la versión en C con el archivo `mips32_plot.c`. Para poder compilar el código desarrollado deberán cambiar la definición en el archivo `Makefile.in` la línea número 6:

```
SRCS = mips32_plot.c main.c mygetopt_long.c
```

por

```
SRCS = mips32_plot.S main.c mygetopt_long.c
```

Luego deberán invocar la siguiente secuencia de comandos para limpiar los archivos temporales y generar los nuevos Makefiles:

```
$ make clean
$ make makefiles
$ make
```

## 4.4. Detalles de la implementación

Para optimizar los accesos a las llamadas a servicio del sistema (`syscalls`), deben utilizar un buffer de  $n$  bytes para escribir los datos de salida para luego ser enviados al archivo de salida. El tamaño  $n$  debe ser parametrizable mediante un `#define`.

## 5. Informe

El informe, a entregar en formato impreso y digital<sup>1</sup> deberá incluir:

- Documentación relevante al diseño e implementación del código desarrollado para adaptar el programa. Incluir el diagrama de *stack frame* de las funciones implementadas en MIPS32.
- Documentación relevante al proceso de compilación: cómo obtener el ejecutable a partir de los archivos fuente. Especificar modificaciones realizadas a los archivos provistos por la cátedra si es que los hubo.
- Las corridas de prueba, con los comentarios pertinentes.<sup>2</sup>
- El código fuente, en lenguaje C (y MIPS32 donde corresponda)
- Este enunciado.

---

<sup>1</sup>En CD, DVD o memoria flash.

<sup>2</sup>Las pruebas provistas deben ejecutarse correctamente en NetBSD sobre MIPS32 sin modificación alguna.

## 6. Fecha de entrega

La fecha de vencimiento será el Martes 01/11.

## Referencias

- [1] Trabajo Práctico 0, 2do cuatrimestre de 2016.  
<https://groups.yahoo.com/neo/groups/orga-comp/files/TPs/tp0-2016-2q.pdf>.
- [2] Código fuente con el esqueleto del trabajo práctico.  
[https://drive.google.com/open?id=0B93s6e6NY\\_j1TFV2TFBqbUNKZ3M](https://drive.google.com/open?id=0B93s6e6NY_j1TFV2TFBqbUNKZ3M).