Departamento de Informática
Universidad Técnica Federico Santa María

Universidad Técnica Federico Santa María
Campus Santiago San Joaquín
INF-245 - Arquitectura y Organización de Computadores
Primer Semestre 2017

Tarea 4

Cristian Navarrete 201573549-2 cristian.navarreteg@sansano.usm.cl
Juan Pablo Jorquera 201573533-6 juan.jorqueraz@sansano.usm.cl

1. Código Assembly (edo.s)

Inicialmente se define la entrada y un valor que luego se usara como cero ya que los flotantes no tienen un 0 definido.

A continuación se encuentran los prints que se usaran.

MAIN: simplemente se encargara de recibir el input y de luego mostrar el output. y setea $f32 como 0.0 para luego poder mover datos entre variables sin problemas.

EULERMEJORADO: Crea las condiciones iniciales y llama a LOOP.

PENDIENTE: Función utilizada por LOOP para calcular como su nombre lo indica la pendiente que sera utilizada luego para realizar los cálculos. Es importante mencionar que se utilizan $f16 y $f17 para intercambiar datos entre funciones ya que en flotantes no hay un estándar en cuanto a que variables se usan para que cosa.

LOOP: Se guardan el $ra en el stack para luego poder volver a este en el final del programa. Se realizan los cálculos según lo indicado en el enunciado y se compara x con xMax para determinar si salimos del loop, el resultado de la comparación se almacena en una variable especial en otro coprocesador y se puede utilizar bc1t para realizar saltos en caso de que sea verdad. Finalmente devolvemos el stack a su posición inicial para poder volver a MAIN.

2. Código en C (edo-c.c):

No hay cambios, es lo mismo que esta en assembly pero más bonito gracias a la syntax de C (nunca pensamos que diríamos esto)

Importante notar que el código es mucho más pequeño ya que cosas como el calculo de la pendiente y de yNext se puede hacer en una linea a diferencia de assembly donde se debe separar en sumas y multiplicaciones.

3. Assembly generado por C (edo-c.s):

Los códigos son completamente diferentes, esto es por que el código escrito por nosotros en assembly esta creado para MIPS, mientras que el compilador gcc crea el código assembly para una arquitectura x86 como lo es el macbook en el cual compilado.