Universidad de Buenos Aires - FIUBA

**66:20 Organización de Computadoras**

Trabajo práctico 1: Conjunto de instrucciones MIPS

Fonseca, Matias (98591)

Bacigaluppo, Ivan (98064)

Irrazabal, Diego (98125)

8 de Mayo del 2018

Diseño e implementación

En este TP el objetivo es familiarizarse con el conjunto de instrucciones de MIPS y la ABI, para esto se nos dieron varios archivos que conforman el programa que dibuja el conjunto Julia y sus vecindades. Se nos pide modificar el programa para que el computo del fractal tenga soporte nativo para NetBSD/pmax y desarrollar la lógica del computo del fractal en assembly MIPS. Para esto debemos reescribir la función mips32.plot().

Se implemento las siguientes funciones:

* Inttostr : recibe un int y obtiene el string de este.
* print\_p2: imprime “P2” por pantalla
* print\_number: recibe un int y lo imprime por pantalla.
* print\_barran: imprime “\n”.
* print\_header: imprime los encabezados, utilizando print\_numbre, print\_p2 y print\_barran.
* get\_shades: recibe los parámetros para el computo de los fractales. Realiza el computo de los fractales, los imprime y devuelve uno si todo salió bien o cero si hubo un error.

En todas estas funciones utilizamos stacks frame para guardas las variables que se utilizan entre ellas.

Compilación

Para el proceso de compilación de la función implementada en Mips, se utilizó Gxemul para emular un procesador MIPS R3000 con una imagen de sistema operativo NETBSD 3.0.

Este se ejecutó con soporte X11, mediante la generación de un túnel entre el HostOS y el GuestOS con SSH/SSHD.La transferencia de archivos (los necesarios para compilar el tp) se hizo con la utilidad SCP.

Para realizar la compilación de mips32 plot.S junto con el main.c y demás archivos necesarios se utilizó el makefile provisto por la cátedra. Para esto se emite el comando “make” , obteniendo de esta manera el ejecutable.

El archivo de salida se puede enviar al sistema operativo HostOS mediante SCP

así corroborar el resultado del mismo.

Código MIPS

Conclusion

Al final de este TP pudimos observar los usos de cada distinta herramienta. Vimos como gxemul nos posibilita levantar una imagen virtual de otro sistema operativo dentro de nuestra propia máquina. Esta imagen sea NetBSD que utilizamos para poder trabajar con otras arquitecturas que nuestro sistema operativo no permitía, como en particular assembly MIPS. De esta manera pudimos implementar en MIPS el computo de los fractales, donde aprendimos el uso del stack frame y el conjunto de instrucciones de MIPS.