

ST0254 Organización de computadores

Entrega Final, Práctica 3: Manipulación de archivos de imágenes usando Assembler en Linux

Santiago Palacio Gómez, *Estudiante, EAFIT*,
 Pablo Velásquez Manrique *Estudiante, EAFIT*,
 y Santiago Zubieta Ortiz, *Estudiante, EAFIT*

Resumen—Entrega Final de la práctica 3 de Organización de Computadores.

Index Terms—Organización de Computadores, ST0254, Assembler, Universidad EAFIT



1. INTRODUCCIÓN

ESTE reporte corresponde a la entrega final de la práctica 3 de Organización de Computadores, en la Universidad EAFIT, Manipulación de imágenes usando Assembler en Linux.

Profesor Jose Luis Montoya Pareja.

22 de Noviembre de 2013

2. ESTADO DEL PROYECTO

El proyecto se dio por concluido, llegando exitosamente a varios de los objetivos propuestos por el enunciado dado por el profesor.

3. DIFICULTADES/PROBLEMAS

1. Se perdió mucho tiempo tratando de tener las cosas funcionando. La máquina virtual del profesor puso problemas (Ubuntu 13.04), también un Ubuntu 12.04 de 64 bits virtualizado, y un Ubuntu 11.04 de 32 bits nativo en un computador viejo. Finalmente optamos por virtualizar un Ubuntu 12.04 de 32 bits, y aun así tocó instalar esta máquina virtual 2 veces

por problemas en la compilación de las librerías que no entendimos su causa.

2. Realizar el cambio de pixeles al recorrer el arreglo de datos en assembler. Pudimos con 3 ciclos anidados mover los iteradores correctamente, pero al momento de hacer el intercambio mandar el punto de origen y el de destino a c++ para que allí se hiciera el intercambio, debido a que de la manera que teníamos pensada implementar la recuperación de un pixel a partir del apuntador al inicio del arreglo, y sumarle un offset, funciona al sumarle un valor fijo pero no un valor contenido en un registro/label.
3. En un principio pudimos llamar exitosamente desde asm procedimientos definidos por nosotros mismos en un .c, pero posteriormente apenas hacíamos un include de las librerías de manipulación de imágenes, todo se enloquecía y no dejaba linkear al momento de compilar

4. SOLUCIONES

1. Instalar una y otra vez máquinas virtuales y todas las dependencias necesarias hasta

- que todo se diera exitosamente, principalmente eran errores que se daban al momento de realizar el make de OpenCV
2. No encontramos solucion al problema de no realizar la operacion de intercambio de pixeles en asm sino en c++, por lo que la unica funcion externa que definimos que no este encargada de cargar la imagen, guardar la imagen, u obtener las dimensiones de la imagen, es la de a partir de un valor p1 de un arreglo y un valor p2, pasar un pixel a otro.
 3. Lo del problema del linkeo al momento de incluir librerias de OpenCV vimos que se debia a que estabamos tratando de linkear como un archivo .c, y en las librerias incluidas estabamos usando cosas de c++, por lo que toco linkear el archivo como un c++, y los procedimientos que se llaman desde asm definirlos como externos y de c para evitar problemas de compatibilidad.

5. BIBLIOGRAFÍA

1. Argumentos en NASM, <http://www.dreamincode.net/forums/topic/285550-nasm-linux-getting-command-line-parameters/>
2. Guia instalacion OpenCV, <http://desarrollophpsenior.wordpress.com/2012/09/19/instalar-opencv-2-4-2-en-ubuntu-12-04-lts/>
3. Compilar un archivo que incluya OpenCV, <http://stackoverflow.com/questions/10094692/opencv-error-undefined-reference-to-cvloadimage-ubuntu>
4. Linkear un archivo .o de NASM y .o de C++ a un ejecutable unico, permitiendo llamar funciones del C++ en el NASM <http://stackoverflow.com/questions/5612492/how-to-link-elf32-assembly-and-c-files>
5. Referencia de OpenCV <http://docs.opencv.org/modules/refman.html>
6. Referencia de NASM y ejemplos <http://www.nasm.us/doc/nasmdoc3.html>
7. CheatSheet de NASM para x86 <http://www.cs.uaf.edu/2006/fall/cs301/support/x86/>