Trabajo Práctico Especial

Arquitectura de Computadoras
Informe

Pablo Diaz Uboe, Iñaki Lanusse

Primer Cuatrimestre 2013

Desarrollo y decisiones de diseño

Lo primero que se hizo fue obtener varios de los programas recomendados por la cátedra para el desarrollo del TPE, como por ejemplo Bochs y VirtualBox, y aprender lo básico de su funcionalidad como para el testeo de todo lo implementado para el TP.

Se tomó como base del trabajo el kernel de prueba otorgado por la cátedra, el cual sirvió como referencia a la hora de programar.

El primer objetivo que se buscó fue implementar las interrupciones int08h (timer tick) y int09h (interrupciones de teclado). Se agregaron al IDT, y luego definimos el teclado como una matriz de caracteres.

Se quiso poder imprimir en pantalla un string a manera de verificar el correcto funcionamiento de lo visual del TP.

Una vez logrado ésto, se empeñó en escribir las primitivas __read y __write, y utilizarlas en todas las funciones ya implementadas que la requieran. En un principio, la idea era separar Kernel Space y User Space mediante el uso de la int80h, pero a falta de una solución a tiempo, se optó por permitir llamadas directas a las primitivas sin la intervención de dicha interrupción.

Ya lograda la funcionalidad básica del kernel, se trabajó en el diseño de las funciones de la biblioteca estándar stdio.h y algunas de string.h, es decir, se implementó putchar, getchar, printf y scanf para su uso en el TP.

Luego, se trabajó en la lectura y ejecución de comandos por parte del shell. Las primeras implementadas fueron help y time, seguido de echo, y por último, lspci, cputemp y test.

A continuación, se listarán detalles más puntuales de cada parte del TP que presentaban la necesidad de una breve explicación.

Shell

El shell es una estructura usada para representar la consola. Dentro de ella hay un vector de 80x25x2, que representa la pantalla en modo texto. También cuenta con un cursor que apunta a la posición donde escribirá/borrará. El shell es el encargado de llevar a cabo los llamados a las rutinas correspondientes para cada comando; es decir, el shell es el intérprete de comandos.

Teclado

El teclado es una matriz de caracteres de 88x2, con un buffer para almacenar lo que se escribe. Al generarse la interrupción de teclado, por medio de la primitica read, se lee del puerto 60h del mapa de I/O el *scancode* correspondiente a la tecla ya sea que se apretó o soltó (makecode o breakcode), y se busca en la matriz según corresponda para guardarse en el buffer.

Rutina de prueba

Para probar el indicador de temperatura, pensamos en una rutina que calcule todos los números primos hasta una determinada cifra. Para llamar a dicha rutina, se debe invocar en el Shell el comando “test”.

Comando lspci

El comando lspci, fue el que mas problemas nos ocasionó. Para empezar, encontramos escasa información en Internet y de poca utilidad. Nos costó lograr conseguir los datos de los dispositivos. Luego, tuvimos dificultades con la magnitud de la base de datos de pcidatabase.com y no pudimos arreglarlo, por lo que optamos por incluir solo unas pocas entradas. Este problema nos tuvo hasta ultimo momento y fue el ultimo que se

resolvió.

Temperatura

Otro problema que enfrentamos fue la lectura de la temperatura. Para este caso, la información disponible en Internet también nos fue de poca utilidad y además, solo funciona en maquinas Intel ya que el registro que leemos, solo esta disponible en dichas maquinas (y no en todas). Por este motivo, no funciona en maquinas virtuales y, además como la PC en la que testeamos tiene procesador AMD, tampoco funcionaba en aquella corriendo nativamente. En conclusión, esta función no la pudimos probar extensamente.

Fuentes de investigación

Para el desarrollo del TP, se utilizó como fuente la página de Internet wiki.osdev.org, pero también se consultó ocasionalmente stackoverflow.com. Para la implementación de la biblioteca estándar, se siguieron las especificaciones de Kernighan y Ritchie de su libro, The C Programming Language.