

Informe

TPE Arquitectura de Computadoras

Sistema Operativo:

MaritoBaracOS V1.1

Integrantes:

- Bezdjian, Alejandro
- Gómez, Jorge
- Sakuda, Eugenia

Fecha de Entrega:

02 de Diciembre de 2014

Introducción

En este informe se detallan la toma de decisiones, inconvenientes y mejoras posibles para el desarrollo del Trabajo Práctico especial de la materia Arquitectura de Computadoras. El nombre del sistema operativo desarrollado es MaritoBaracOS V1.0, y cuenta con diversas funcionalidades, como la impresión de la información de la BIOS, la reproducción de sonidos y el manejo de tres excepciones, entre otras.

Decisiones

Al momento del diseño del trabajo, se tomaron en cuenta los conocimientos aprendidos en la materia, tanto en la parte teórica como práctica. La principal rama de diseño se encarga en separar el "User Space" con el "Kernel Space". El kernel, en conjunto con las funciones de drivers, se encarga del manejo de los recursos del sistema, y provee funciones para el manejo de los mismos. El usuario no interactúa directamente con los recursos, sino que siempre lo hace a través del kernel.

Excepciones elegidas

- ❖ Para el ítem de manejo de excepciones, se eligieron las excepciones siguientes:
 - Zero Divide Exception: Excepción generada al momento de realizar una división por cero.
 - General Protection Fault: Excepción lanzada cuando se lanza una excepción no descrita en la IDT.
 - Debug Exception: Excepción lanzada cuando un debugger marca un breakpoint.
- ❖ Se utilizó el modo texto de la placa de video.
- ❖ Para la IDT se decidió no mover las interrupciones que se superponen con las excepciones, debido a la falta de tiempo, y que no requeríamos de ninguna que se superponga.
- ❖ Se decidió que en las primeras 11 líneas se muestre el logo del Sistema Operativo.

Inconvenientes

Durante el transcurso del desarrollo del trabajo se encontraron diversos inconvenientes.

Inconvenientes Operativos:

- ❖ No se lograba ejecutar el sistema operativo en ninguna computadora con el uso del pendrive, ni en ninguna máquina virtual utilizando el archivo .iso. Para esto utilizamos los siguientes programas:
 - Multisystem: Nos facilitó la creación de la imagen para crear un pendrive booteable.
 - Genisoimage: Nos facilitó la creación de la imagen de cd.
- ❖ No se logró utilizar el qemu y bochs en mac y windows.
- ❖ El VirtualBox nos generó inconvenientes al momento de utilizar las máquinas virtuales por problemas de los headers del kernel de linux.

Inconvenientes de Implementación:

- ❖ No se logró poder eliminar las interrupciones del timer tick al momento de escribir un comando en consola, lo que a veces genera que cuando se presiona una tecla no se ejecute la interrupción, y por lo tanto no la agregue al buffer de teclado.
- ❖ Se tuvo dificultades para encontrar la información de la BIOS, pero se logró obtener la misma utilizando el SMBIOS (System Management BIOS), el cuál no esta disponible para computadoras muy viejas.
- ❖ No se pudo implementar las excepciones que realmente queríamos, en lugar de la excepción de debug queríamos utilizar Invalid Opcode.
- ❖ Se encontró un error en diferentes ocasiones, al momento de imprimir en pantalla.

Mejoras Posibles

- ❖ Se deberían mover las interrupciones que se superponen con las excepciones en la IDT.
- ❖ Se podría editar el shell para que busque los programas en un array de programas, que serían los programas instalados, y no realizar un string compare por cada uno de ellos.
- ❖ Sería bueno que el driver de video permita el scroll de la pantalla.
- ❖ Se podría implementar funciones que permitan agregar configuraciones a los sonidos, como la duración o agregar la escala musical completa.
- ❖ Se debería implementar mayor seguridad al Sistema Operativo.

Programas de Terceros

Para el desarrollo y testeo del sistema se utilizaron los siguientes programas y herramientas:

Gcc, Nasm, Ld, Bochs, Geniso, MultiSystem, Sublime, VirtualBox, Bitbucket.

Código e información de Terceros

Para la realización de este trabajo, se tomaron extractos de código de fuentes externas para las siguientes funciones o encabezados:

- Cursor.asm: seteo del cursor.
- Libasm.asm: reproducción de un sonido.
- Libc.c: acceder a la información de la BIOS.
- Panic.c: chequeo de la pila.
- Stdlib.c: la comparación de strings.
- Keyboard.c: Tabla de correspondencias entre caracter y código ascii.
- Kc.h: estructuras para el manejo de la SMBios.
- Stdarg.h: manejos de argumentos variables.

Además se contó con información de los siguientes sitios:

- wiki.osdev.org
- El Lenguaje de Programación C - Kernighan and Ritchie
- Material teórico brindado por la cátedra

Notas de la Versión 1.1:

Updates:

- Se mejoró la impresión de la IDT, imprimiendo el offset, el selector y el acceso. También se detalla si esta la programamos nosotros o no en el sistema.
- Se actualizó el comando about, imprimiendo en el logo los datos del grupo.
- Se modificó el sonido de las excepciones por defecto, y el orden en que se ejecuta al momento de realizar una excepción.
- Se agregó la tabla del bloqueo de mayúsculas al teclado.
- Actualizado help.
- Agregada la impresión de una alerta si el comando a ejecutar no existe.
- Agregado comando exit que simula la salida del sistema operativo.

Fixes:

- Corregidas las funciones maths y dolarcalculator, ahora validan el ingreso de datos.
- Corregidos errores menores en stdlib (No tomaba el caracter si el código ascii era el del número nueve en el caso de que se quiera ingresar un número).
- Se corrigieron errores en la función de piano.