

# Trabajo Práctico Especial

Arquitectura de Computadoras

Alumno: Rivas, Leandro Matias

Legajo: 51274

Cuerpo Docente: Valles, Santiago Raul

Merovich, Horacio Victor

Nizzo Mc Intosh, Augusto

# Indice

|  |   |
|--|---|
| El kernel .....                              | 3 |
| Manual de usuario .....                      | 3 |
| Respecto al mouse .....                      | 4 |
| Savescreen .....                             | 4 |
| Implementación del mouse: dificultades ..... | 4 |
| Sobre el teclado, video y otras funciones .  | 5 |
| Real Time Clock (RTC) .....                  | 5 |

## El kernel

Para la realización del kernel, se respetó todos los aspectos necesarios para poder separar lo que es el kernel space del user space. Se intentó de que solamente el kernel sea capaz de comunicarse con los drivers tanto de teclado como del mouse y, por su parte, el shell interactúa con el usuario.

También, se buscó que el código esté lo más modulado posible, no solo por tener una mejor organización y lectura del mismo, sino que esto permite en un futuro poder mejorarlo si afectara gravemente su funcionamiento.

Las interrupciones fueron un tema importante que debió estudiarse con detenimiento para poder lograr que los periféricos funcionen, en especial aquellos que se encontraban en el PIC2 (slave), dado que para que este funcione se debe desenmascarar el IRQ2 que se encuentra en el PIC1. Por último, para la realización de este TPE, se consultó en diversas páginas código, que al ser sumadas al trabajo fueron indicados sus fuentes.

## Manual de usuario

Aquí se listan los comandos presentes en el kernel.

**savescreen**: permite determinar el tiempo (en segundos) en el que aparecerá el salvapantalla. Por Default, está seteado en 10 segundos.

**hello world**: imprime en pantalla el mensaje "Hello World!".

**clear**: limpia la pantalla.

**time**: muestra la hora y el día del sistema.

**set time**: permite al usuario determinar la hora, minutos y segundos del sistema.

**help**: imprime la lista de comandos que el usuario puede ejecutar. Esta misma lista de comandos aparece cada vez que se haga click izquierdo o derecho con el mouse.

## Respecto al mouse

El kernel permite al usuario hacer uso del mouse. El puntero del mouse está representado por un cuadrado verde que se mueve cada vez que el mouse se mueve. Al interrumpir el mouse, envía los datos de los botones, si fueron presionados o no, como también de su desplazamiento en x e y, permitiendo actualizar la posición cuadrado verde. Al iniciar el kernel, el mouse se encuentra en el centro de la pantalla.

Previamente, el dispositivo tuvo que ser inicializado, es decir, prepararlo para que pueda ser capaz de enviar los paquetes de datos (3 bytes).

## Savescreen

El salvapantalla, por default, aparece cada 10 segundos. Estando en ejecución, se muestran diversos colores, uno detrás de otro, que permanecen en pantalla durante cierto tiempo.

El tiempo en el que tarda en el salvapantalla en mostrar puede ser configurado mediante el comando *set time*.

Para que se muestre, debe transcurrir el tiempo definido y no debe presionar ninguna tecla como tampoco ningún botón del mouse ni moverlo. Estas acciones producen interrupciones que lo desactivan, mostrando el estado previo del shell a su aparición.

## Implementación del mouse: dificultades

Este ha sido uno de los aspectos más desafiantes del trabajo práctico. Se buscó la información necesaria sobre su funcionamiento, para luego poder implementar su código.

Al no poder diseñar un código para inicializar el mouse de forma adecuada, se optó por buscar un código que lo hiciera en lenguaje C o assembler.

Varios fragmentos de código se han probado sin éxito hasta encontrar aquél que no mostrara ningún error en la terminal o simplemente reiniciar permanentemente el kernel.

Otro inconveniente encontrado al utilizar QEMU o VirtualBox es que no retornaban los 3 bytes de datos, sino 2. Por otro lado, al ejecutar el kernel desde un pendrive directo sobre la PC, el mouse funcionaba correctamente, por lo que cada vez que deseaba realizar un cambio al código y ver su debida ejecución, debía grabar el kernel actualizado y reiniciar la PC.

Este fue el feature último a implementarse por lo que su código no está prolijamente presentado. De no ser por el tiempo acotado, éste podría ser mejorado.

## Sobre el teclado, video y otras funciones

En este trabajo práctico se ha utilizado parte del código de mi trabajo práctico del cuatrimestre anterior. El código que permite el funcionamiento del teclado, video y las funciones del archivo lib.c fueron agregados. Se los ha adaptado para los requisitos actuales como también se los ha tratado de mejorar sus implementaciones.

## Real Time Clock (RTC)

No solo es utilizado para mostrar la hora del sistema cuando el usuario lo pida, sino también para determinar cuándo es momento de mostrar el salvapantalla. Para ello, cada vez que el usuarios settea el tiempo, internamente se guarda la hora (hh:mm:ss) en la que hizo. Luego esa hora es comparada con la actual en el handler del timer tick, que interrumpe permanentemente. La dificultad estuvo en que la interrupción es muy rápida que no da tiempo a que la hora cambie por lo que la actual y la previamente guardada eran la misma haciendo que la implementación del código del salvapantalla funcione incorrectamente. Este problema pudo ser solucionado.