

# Trabajo Práctico Especial

## Segundo Cuatrimestre 2014

### Objetivo

Implementar un micro *kernel* que implemente administre los recursos de hardware de una computadora y muestre características del modo protegido de Intel.

### Enunciado

Implementar un micro *kernel* booteable por GRUB, utilizando cómo ejemplo el TP5 de la guía de ejercicios. Este módulo debe administrar los recursos de hardware y proveer una API para que aplicaciones de usuarios puedan utilizar estos recursos.

Se deben definir dos espacios claramente separados, uno *kernel space* y el otro *user space*. El espacio del *kernel* interactuará directamente con el *hardware* mediante *drivers*, mientras al mismo tiempo proveerá funciones al *user space*. El espacio de usuario no debe acceder por ninguna razón al hardware, sólo a través del espacio de *kernel*. En la **Figura 1** se puede ver un ejemplo de los módulos que conforman el espacio de usuario y los módulos que conforman el espacio de kernel.

Se deberá definir una API con la cual el espacio de usuario puede acceder a las funciones del kernel. Este acceso deberá ser implementado a través de la interrupción de software 80h, ya que se supone<sup>1</sup> que estos dos módulos se encuentran en distintos espacios de memoria. Dicha API deberá basarse en la API de Linux, que está comprendida por funciones cómo *read* y *write*.

Además se deberá definir un set de funciones para interactuar con dicha API, a saber, deberá formar parte al equivalente en Linux a la biblioteca estandar de C. En la cual, se deberá estar basado. Por ejemplo, deberá contar con funciones como *scanf*, *printf*, *putChar*, *getChar*.

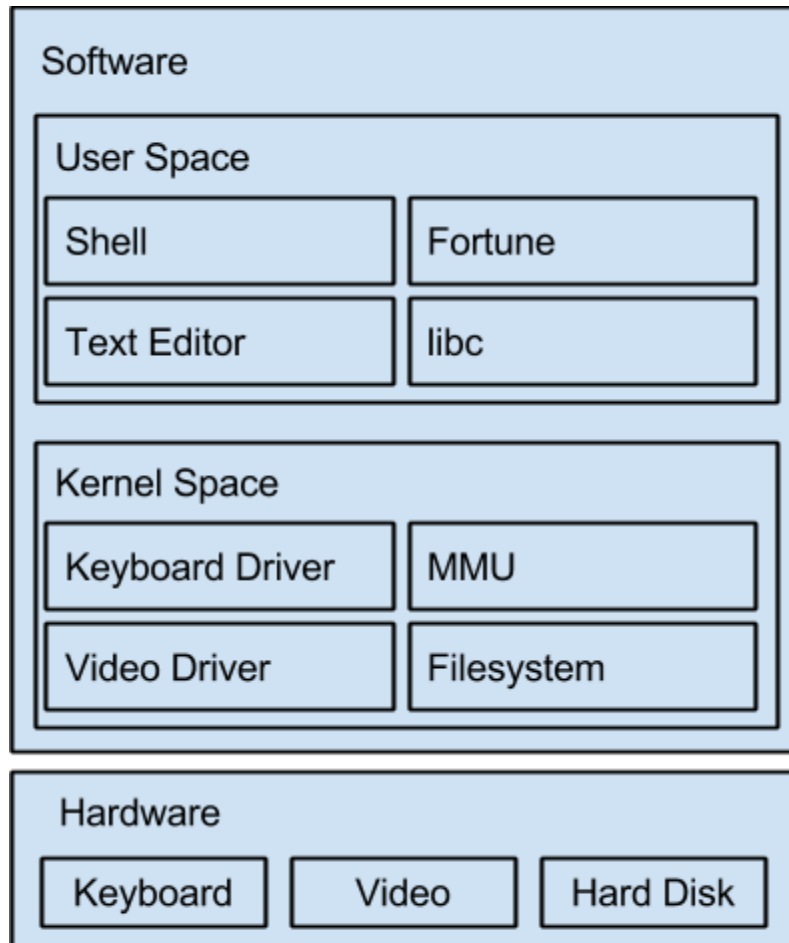
La implementación del sistema deberá contar con los siguiente:

- Interprete de comandos: que demuestren el funcionamiento del *kernel*. Deberá tomar dichos comandos por entrada estándar y mostrar el resultado de la ejecución por salida estándar.
- Programas:
  - Un programa que muestre el contenido de la IDT (infoIDT)

---

<sup>1</sup> Para simplificar el desarrollo del trabajo práctico.

- Un programa que pruebe la existencia de al menos 3 rutinas de atención de excepción.
- Un programa que obtenga la marca, la fecha y la versión del BIOS.
- Un programa que testé el altoparlante con distintos sonidos.
- Cuando se lance una excepción, se debe producir un sonido distinto por el altoparlante de manera configurable.



**Figura 1 - Ejemplo de Módulos**

Los requerimientos implican la implementación de drivers de teclado, de video y de mouse.

El sistema podrá ser desarrollado sobre un emulador, pero deberá funcionar sin ningún problema sobre una PC real con procesador de 32 bits.

## Integrantes

El grupo de trabajo podrá contar con un máximo de tres (3) integrantes. Aquellos estudiantes que estén recursando la materia, sólo podrán reutilizar el código entregado anteriormente si conservan el mismo grupo que entregó dicho código.

## Calificación

La nota del trabajo práctico consta de la evaluación de los entregables más la demostración de su funcionamiento en un coloquio oral. En dicho coloquio se podrá consultar a cualquier integrante cualquier porción de código cual es su implementación y funcionamiento. Además se podrán consultar conceptos teóricos de la materia.

A la hora de la presentación, todos los miembros deberán estar presentes, sin excepción.

En el caso de que se deba reentregar el TP por no estar aprobado en primera instancia, se penalizará a los integrantes del grupo con una disminución de 2 (dos) puntos sobre la nota final.

## Entregables

Cada grupo deberá entregar:

- CD booteable con el sistema implementado.
- Manual de usuario
- Informe del trabajo práctico, que explique el diseño elegido y sus justificaciones, pros y contras.

Todo el material deberá ser entregado en formato físico el día de la entrega y digital vía mail el día de la entrega a la casilla [arq-catedra@googlegroups.com](mailto:arq-catedra@googlegroups.com)

## Consideraciones

- El tipo de diseño y la forma de implementación serán discutidos entre el grupo y la cátedra durante las clases de laboratorio, dejando la posibilidad de modificar éste enunciado escrito, previo acuerdo entre el docente y los integrantes del grupo.
- Para la evaluación se tendrá en cuenta no sólo el funcionamiento del programa sino también la calidad del software y la documentación escrita pedida.
- Cualquier funcionalidad agregada al TP por sobre lo pedido deberá estar documentada apropiadamente.
- Cualquier código encontrado en Internet u otra fuente debe ser consultado con la

cátedra previo a la inclusión en el TPE. Toda inclusión de código de terceros deberá ser documentada con los enlaces o referencias correspondientes a su origen.

- Cualquier aclaración oral a cargo de la cátedra con respecto al enunciado del TP tiene la misma validez que el enunciado escrito.
- En caso de que se detecte plagio, el trabajo práctico se encontrará automáticamente desaprobado y el grupo se arriesga a sanciones académicas determinadas por la Secretaría Académica y el Consejo de Regencia, cómo lo establece el Reglamento Universitario bajo el concepto de Falta de Conducta Académica (7.4 Boletín General 2014).
  - Por tratarse de que todos los trabajos corren sobre la misma arquitectura, existirán muchos códigos equivalentes o similares. La cátedra tendrá estos elementos en cuenta, y en el caso de que surja una duda, esta deberá ser contestada por la cátedra.

### **Fechas de Entrega y Defensa**

La entrega del TPE es hasta el domingo 23 de Noviembre a las 23:59 hs. por correo electrónico dirigido al correo de la cátedra y en el Laboratorio de Informática al día siguiente en formato CD-ROM e impreso.

La defensa del TPE será el 24 de Noviembre a las 16 hs en el Laboratorio según orden establecido por la cátedra para los diferentes grupos.