

Arquitectura de Computadoras

TPE - Especial

Informe

54308 – Francisco Bartolomé

52056 – Juan Marcos Bellini

55291 – Julián Rodríguez Nicastro



Diseño

El sistema está dividido en dos partes, por un lado el KernelSpace y por otro, UserLand.

El KernelSpace es el espacio encargado de interactuar directamente con el hardware. Este se comunica mediante drivers con los distintos periféricos de la computadora. Por ejemplo, puede imprimir en pantalla, o reproducir sonidos.

UserLand, en cambio, es la pieza de software encargada de interactuar con el usuario. Este corre programas con menores niveles de privilegios. Debido a que no puede acceder al hardware de manera directa, UserLand accede al KernelSpace únicamente mediante interrupciones de software (INT 80h). Sin embargo, puede correr cualquier tipo de código que no requiera de accesos directos a hardware.

Si bien este tipo de diseño es más costoso debido al reiterado uso de interrupciones para acceder al KernelSpace, de esta manera se logra un sistema más seguro, ya que el usuario jamás podrá acceder directamente al hardware.

SystemCalls

Como fue mencionado anteriormente, el UserLand y el KernelSpace se comunican mediante el uso de la interrupción de software INT 80h. Esta interrupción es una entrada en la IDT, y será usada únicamente por el UserLand. El proceso consta de dos etapas: la etapa de usuario y la etapa del kernel. La primera etapa comienza llamando a funciones wrapper que ordenan los parámetros. Luego se llama a una función que tiene como única instrucción INT 80h, comenzando la parte de kernel.

En el kernel esta interrupción se maneja en un handler que deriva, en base a los parámetros, a la rutina correspondiente. El camino es el siguiente:

syscallWrapper (C) → syscaller (C) → int 80 (asm) → syscallHandler(asm) → handler(C) → rutinaWrapper(C) → rutina (asm ó C)

Al terminar la rutina de interrupción correspondiente, la función syscallHandler, para volver al contexto de usuario, utiliza la instrucción correspondiente (IRETQ), y el control vuelve a UserLand.

Se cuenta con las siguientes system calls. Para empezar, la función **read** sirve para leer información de los distintos dispositivos. Por ejemplo, se puede leer del teclado, del timer tick, o del video (por ejemplo, en qué posición se encuentra escribiendo actualmente). Esta system call es utilizada, por ejemplo, por la función **getchar** de la librería estándar provista.

Luego se tiene **write**. Write escribe datos en los distintos dispositivos. Sirve, por ejemplo, para imprimir en pantalla, o para habilitar el speaker (se le escribe la frecuencia a la que debe vibrar). En la librería estándar, **printf** y **putchar**, utilizan esta system call para realizar la tarea correspondiente.

Existen distintas variedades de reads y writes. Por ejemplo, existe un read que, a medida que vaya leyendo caracteres del teclado, los va imprimiendo en pantalla. Se podría utilizar para evitar la doble interrupción de software (leer y escribir lo leído) al pedir que se ingresen datos por teclado. Para conocer todas las variedades, ver el archivo *syscalls.c*, en el directorio *kernel*.

Drivers

Video

El driver de video tiene acceso a la posición de memoria en donde se guardan los caracteres que se verán en la pantalla, así como su color y fondo. Este se encarga de mostrar en pantalla los caracteres que se le soliciten, además de borrar la pantalla cuando se lo desee.

Sonido

Este driver se encargará de reproducir sonidos mediante el uso del PC Speaker. Las dos funciones que cumple son reproducir un “beep” con determinada frecuencia, y apagar el sonido.

Teclado

Este driver se encarga de encolar scancodes de las teclas presionadas y soltadas en un buffer interno. Esto se realiza a través de la interrupción de teclado asignada en la posición 21h de la IDT.

TimerTick

Este river maneja las interrupciones del timer, llevando la cuentas veces fue interrumpida. Esto se realiza en una variable de tipo entero de 64 bits, permitiendo que la cantidad de interrupciones guardados sea suficientemente alta. Esta se utilizará para tener noción del tiempo. La interrupción se setea en la posición 20h de la IDT.

Shell

La consola se ejecuta inmediatamente de que se inicia el Kernel, ésta correrá hasta que se cierre el kernel. La consola se encarga de pedir comandos a los usuarios y luego llamar diversas funciones dependiendo del comando ingresado por el usuario, en caso de que el comando sea válido.

Los comandos válidos son:

- help: muestra los comandos disponibles.
- clear: borra la pantalla.
- piano: simula un piano utilizando el teclado, como si fueran teclas del piano.
- beep: realiza un pequeño sonido.
- musicplayer: reproduce canciones.
- echo: repite el texto que ingresa el usuario.

En caso de que el comando no sea válido, se le informará al usuario aconsejando hacer uso de "help".