Informe Experiencia 1

Sistemas Operativos
INF-246

Carlos Jara Almendra 201773038-2
Bastián Solar Vargas 201773003-k

1 Introducción

La experiencia comenzó con la descarga del repositorio con lo necesario para emular un sistema en QEMU. Con ello se vincularon las funciones del gdb con la maquina virtual de QEMU para poder hacer uso del *debuger*. Hecho esto se estudió el primer comando de la ejecución, se analizó el bootloader y algunas instrucciones que este realizó, se modificó uno de los archivos constructores del bootloader para entender el actuar, se experimentó con la consola de QEMU para obtener más información del *kernel* y del sistema, y por último se modificó el código encargado de mostrar en pantalla para mostrar números en octales.

2 Desarrollo

1.- ¿Por qué hay un jump en la primera instrucción?

Respuesta:

Antes que nada, se reconoce un *jump* en la primera instrucción pues lo primero en ejecutarse, desde el bloque 0x0000fff0, es un *ljmp* a 0x0000e056. Esto ocurre pues el CPU al iniciar no sabe dónde la *BIOS* comienza ni cuanto espacio ocupa, por lo que tienen cargado de fábrica leer esa *dirección espacial* en específico, la cual corresponde a un bloque del *ROM* con el comando *jmp* necesario para redirigirse a la sección del *ROM* correspondiente de la *BIOS* para comenzar el procedimiento real de inicio del sistema y comprobación de hardware.

2.1.- Considere las instrucciones que están entre la 3ra instrucción (*xor*) y la 6ta (*mov*) del bootloader ¿Qué funcionalidad cumplen estas instrucciones?

Respuesta:

Haciendo uso de los *Hints* entrgados logramos determinar que éstas instrucciones de máquina corresponden parte inicial de la creación de la imagen del bootloader para entregarle el poder al *kernel*, en específico referencian a los **Data Segment Registers**, es decir, configura el *Data Segment*, el *Extra Segment* y el *Stack Segment*, reposiciona los segmentos a sus bloques físicos correspondientes para continuar con el *booteo*.

2.2.- ¿Qué es lo último que hace el bootloader?

Respuesta:

Así como se menciona en la introducción dada en la experiencia, lo último reailzado por el *bootloader* es entregarle el control al *kernel*. En específico, lo último que realiza es ejecutar la instrucción de máquina ***call***, llamando a una dirección de memoria en específico con el fin de comenzar la ejecución del *kernel* y entregarle el control del sistema.

3.- Después realizar los pasos del inciso 3 ¿A qué se debe lo ocurrido debido al cambio?

Respuesta:

Al hacer el cambio causamos que el sistema no se iniciara. Esto se debe a que cambiamos la dirección de memoria que llamaba la **BIOS** con intención de iniciar el *Bootloader*. Como ésta no apuntaba al Bootloader se producía un loop infinito a causa de una mala redirección.

4.- ¿En qué dirección de memoria inicia el kernel?

Respuesta:

Mediante el comando *kerninfo* en consola del *Qemu*, como también se comprobó siguiendo las últimas instrucciones que realizaba el bootloader, hallamos que el *kernel* inicia en la dirección de memoria 0x0010000c y finaliza en la dirección 0x00112944.