Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación

Principios de Sistemas Operativos

Proyecto 1: The Bootloader

Profesor: Ernesto Rivera Alvarado

Estudiantes:

Daniel Blanco 2016139325
Carlos Adán Arguello Calderón
Gerardo Villalobos Villalobos 201143253

Grupo: 40

Primer Semestre, 2020

Contenido

Arquitectura x86

- La arquitectura conocida como x86 comienza en 1978 con los procesadores Intel 8086/88 (aunque previamente habían aparecido el 4004, el 8080 y el 8085.). Estos procesadores pertenecían a la arquitectura IA-16(Intel Architecture 16 bits).
- Durante toda su evolución, desde 1978, Intel ha mantenido la compatibilidad binaria con los procesadores precedentes. Esta compatibilidad se *rompe* con los procesadores Itanium e Itanium2, con arquitectura IA-64, que son totalmente incompatibles con sus predecesores.
- El (80)386 fue el primer procesador de Intel con un juego de instrucciones de 32 bits (IA-32). Tanto los operandos como el direccionamiento en memoria utilizan 32 bits, por lo tanto, el 386 tiene un espacio de direccionamiento de 4GB.
- El 386 también es el primero en introducir una MMU la paginación, con un tamaño de página fijo de 4KB. Los procesadores anteriores ya poseían un esquema de segmentación, que en el 386 se puede obviar utilizando un esquema de memoria plano.

Características Básicas

- La arquitectura x86 es de longitud de instrucción variable, de tipo registro memoria y diseño CISC.
- El espacio de direcciones lineal es de 4GB, aunque la memoria física puede llegar hasta los 64GB en algunos modelos, con acceso desalineado y almacenamiento Little-endian.
- Un programa normal dispone de 8 registros de propósito general de 32 bits, 6 registros de segmento de 16 bits, un registro de estado EFLAGS y un puntero de instrucción EIP, ambos de 32 bits. Dichos registros se pueden acceder desde las operaciones de propósito general, compuestas por las instrucciones de aritmética entera, las instrucciones de control de flujo, las de operaciones con bits y con cadenas de bytes, y las instrucciones de acceso a memoria.
- Un conjunto de 8 registros de coma flotante de 80 bits, un conjunto de 8 registros MMX y XMM, de 64 y 128 bits respectivamente, para realizar operaciones SIMD.
- Un conjunto de recursos para el manejo de la pila y la invocación de subrutinas.
- El SO dispone además de puertos E/S, registros de control, de manejo de memoria, de depuración, de monitorización, etc

Modos de Funcionamiento:

Modo Real

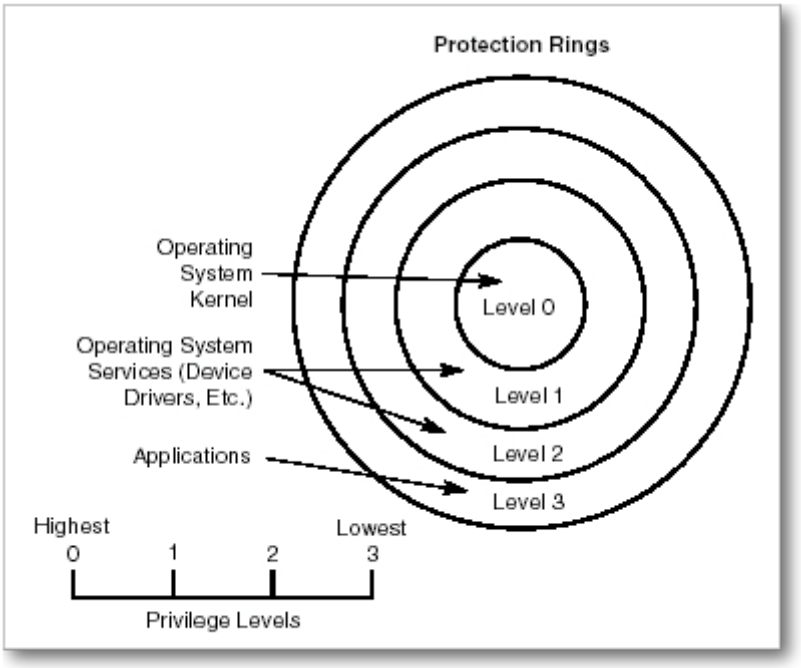
Es un modo de compatibilidad con el 8086, salvo por la capacidad de cambiar a modo protegido. El procesador siempre comienza su ejecución en este modo (tras el arranque o un reset).

Modo Protegido

Es el modo de funcionamiento normal del procesador, en el que están disponibles todas sus características.

Anillos de Protección

Los procesadores de la arquitectura IA-32 tienen 4 niveles de privilegio (0-3), siendo el PL 0 el de mayor nivel de privilegio.



- El nivel de privilegio 0 (PL 0) es el que permite mayor acceso a los recursos del procesador. Es el utilizado por el SO, puesto que en él se pueden ejecutar todas las instrucciones del procesador y en él se tiene acceso a los puertos de entrada/salida.
- El nivel de privilegio 3 (PL 3) es el de acceso más restringido. Las aplicaciones normalmente se ejecutan en este nivel, en el que sólo se pueden ejecutar las instrucciones de propósito general y no se tiene acceso a los puertos de E/S. Existen instrucciones que otorgan el acceso a algunos puertos de E/S y que permiten ejecutar instrucciones tales como STI o CLI. Por supuesto, estas instrucciones no se

pueden ejecutar en el PL 3 sino que las ejecuta el S.O. en PL 0 y su efecto se mantiene al volver al proceso.
- El sistema operativo Linux sólo utiliza los niveles 0 y 3, que se denominan **modo núcleo** y **modo usuario**.
- El nivel de privilegio actual (CPL) se encuentra en los bits 0 y 1 de los registros CS y SS.
- Cuando el procesador está en el nivel de privilegio 0 tiene acceso a una serie de registros y estructuras de datos que permiten controlar el comportamiento del procesador, el manejo de memoria, etc.

Manejo de memoria en el modo protegido

- La arquitectura IA-32 en el modo protegido tiene un esquema de memoria denominado segmentación+paginación.
- A las direcciones lógicas (las emitidas por los procesos) se les aplica primero un esquema de segmentación con el que se obtiene una dirección lineal.
- Posteriormente, a la dirección lineal se le aplica el esquema de paginación, si está activado, para obtener la dirección **física**.
- Las tablas de descriptores utilizadas en la segmentación se acceden utilizando los registros GDTR y LDTR, y se indexan con los registros de segmento o mediante las entradas en la IDT.
- Aunque la paginación sí que puede desactivarse en la arquitectura IA-32, la segmentación está **siempre activa**.

Modelos de memoria

En el modo protegido existen dos posibles modelos de memoria:

- **Modelo plano**: Todos los segmentos se solopan apuntando a la dirección 0 y con el límite a 4GB. Es el modelo utilizado por Linux, ya que la segmentación **no** puede desactivarse.
- **Modelo segmentado**: Cada zona de memoria está contenida en un segmento, con unas características concretas, y del cual no se puede salir.

Además la arquitectura tiene un tipo de segmento especial, el **segmento de estado de tarea**, donde se mantiene la información sobre una tarea. La arquitectura IA-32 tiene su propio concepto de tarea implementado en el procesador y no tiene por que coincidir con el modelo utilizado por el SO.

El Bootloader

Insertamos un USB, determinamos nuestro dispositivo con el siguiente comando:

```
sudo lsblk
sudo fdisk -l
```

Tomamos el .img que queremos que se ejecute seguido del siguiente comando:

```
sudo dd if=proyecto1.img of=/dev/sdX
```

Luego, insertamos el USB en la computadora. Durante el arranque oprimimos la tecla para escoger el dispositivo de arranque, en nuestro caso escoger el USB. Usualmente es F12.

Escogemos el USB para que se ejecute y veremos la ejecución del programa que se guardó.

dd : Copia archivos. Es usado para copiar particiones completas o incluso clonar discos. Convierte y copia un fichero. Copia un fichero (de la entrada estándar a la salida estándar, por omisión) con un tamaño de bloque seleccionable por el usuario, a la par que, opcionalmente, realiza sobre él ciertas conversiones. Generalmente utilizado para realizar imágenes de discos/particiones.

Referencias

EcuRed. (s.f.). X86 - EcuRed. Recuperado 14 marzo, 2020, de https://www.ecured.cu/X86

Peréz, J. C. (s.f.). DSO. Tema 2. Introducción a la arquitectura x86. Recuperado 14 marzo, 2020, de http://sop.upv.es/gii-dso/es/t2-arquitectura/gen-t2-arquitectura.html

Navas, M. A. (2017, 23 noviembre). Procesadores x86 vs ARM: diferencias y ventajas principales. Recuperado 14 marzo, 2020, de https://www.profesionalreview.com/2017/11/26/procesadores-x86-vs-arm-diferencias-ventajas-principales/