



Mumuki Query Learning

Un Runner de SQL para el Proyecto Mumuki

8 de julio de 2015

Trabajo de Inserción Profesional
Tecnicatura Universitaria en Programación

Leandro Di Lorenzo

Coordinador: Ing. Fernando Dodino

Universidad Nacional de Quilmes

Roque Sáenz Peña 352, Bernal

Buenos Aires, Argentina (B1876BXD)

Tel. (+54 11) 4365 7100 | Fax (+54 11) 4365 7101

<http://www.unq.edu.ar/> | info@unq.edu.ar

1. Introduccion

Cuando se enciende la computadora, el motherboard recibe energía y se prende. Al encenderse, el mother inicializa su firmware y trata de hacer andar el CPU. Si todo anduvo bien el CPU comienza a andar. En un sistema multi-procesador o multi-core se selecciona un CPU (al azar) como procesador de arranque (bootstrap processor, o BSP) que ejecuta el BIOS (Basic Input Output System) y el código de inicialización del kernel del sistema operativo.

El BIOS es el primer programa que se ejecuta en la computadora, su propósito fundamental es iniciar y probar el hardware del sistema y cargar un gestor de arranque o un sistema operativo desde algún dispositivo de almacenamiento.

La mayor parte de los registros del CPU tienen valores definidos en el arranque, incluyendo el EIP (puntero de instrucciones) que guarda la dirección de memoria de la instrucción que se está ejecutando por la CPU. La CPU luego comienza a ejecutar código del BIOS, que inicializa algunos de los componentes de hardware de la máquina. Hecho esto, la BIOS lanza el POST (Power on self test) que chequea varios componentes de la computadora.

POST es un proceso de verificación e inicialización de los componentes de entrada y salida en un sistema de cómputo que se encarga de configurar y diagnosticar el estado del hardware.

El POST involucra una serie de tests e inicializaciones, incluyendo descubrimiento de recursos -interrupciones, rangos de memoria, puertos de entrada/salida- para los dispositivos PCI. Los BIOS modernos que siguen las especificaciones ACPI (Advanced Configuration and Power Interface) construyen unas tablas de datos que describen los dispositivos en la computadora; estas tablas son usadas luego por el kernel.

Después del POST, el BIOS intenta bootear un sistema operativo, que debe estar en alguna parte: discos rígidos, CD/DVDs, USBs, u otros dispositivos de almacenamiento. El orden en el que el BIOS busca el dispositivo de booteo es configurable por el usuario. Si no hay ningún dispositivo de booteo el BIOS termina con un mensaje de error del estilo «No hay disco de sistema o error de disco»; esto sucede por ejemplo cuando el disco rígido está roto y no hay otro dispositivo de booteo. Si en cambio el BIOS encuentra un dispositivo de almacenamiento en funcionamiento, entonces permite continuar con el procedimiento de booteo.

Ahora el BIOS lee el sector de los primeros 512 bytes del disco rígido. Este sector se llama MBR (Master Boot Record) y normalmente contiene 2 componentes principales: un pequeño programa de inicio específico de un sistema operativo en el comienzo, seguido de la tabla de particiones del disco. El BIOS a esta altura ya no toma parte de este proceso, sino que simplemente carga los contenidos del MBR en memoria y apunta el EIP a la primera posición donde alojó esos contenidos, para que se comience a ejecutar el programa que se debería haber traído del MBR.

2. Motivación
3. Presentación
4. Alcance
5. Tecnologías
6. Trabajo futuro
7. Conclusiones

8. Referencias

Referencias

- [1] How computers boot
<http://duartes.org/gustavo/blog/post/how-computers-boot-up>