**Tutorial 7**

Hasta el momento solo hemos podido ver como los bootloaders funcionan y hemos realizado un bootloader simple de dos etapas. En este tutorial ahondamos un poco mas en el mundo de los 32 bits de los procesadores y mas que todo el modo protegido. En Real Mode no se tiene ningún tipo de protección de memoria con el cual si se cuenta en modo protegido. Como estamos limitados a utilizar registros del procesador de 16 bits, también estamos limitados a utilizar solo 1MB de RAM. Cuando entramos en modo protegido, obtenemos protección de memoria, acceso a registros de 32 bits, memoria virtual, 4 modos de operación del procesador (Ring 0, 1, 2, y 3) y acceso hasta 4GB de memoria RAM.

La familia de procesadores de x86 de Intel utiliza la arquitectura Van Neumann y el cual establece que un sistema de computadoras tiene tres componentes esenciales : CPU, Memoria y E/S. Estos tres componentes están conectados a través de un bus de sistema. Es por este mismo canal que el CPU interactúa con la memoria RAM y los componentes del hardware. Estos componentes de hardware son mapeados a una dirección en memoria en la cual se le pueden enviar comandos y recibir información. El bus de sistema esta compuesta de 3 buses : un bus de datos, un bus de direcciones, y el bus de control.

En el bus de datos pasan cada uno de los bits de información que van desde y hacia el procesador. Un procesador de 32 bits tiene un bus de datos de 32 bits, siendo 1 bit una línea del bus. El bus de direcciones sirve para referenciar y acceder a un espacio en memoria. En los procesadores viejos solo se tenia espacio a un bus de direcciones de hasta 20 líneas (2^20 = 1MB) por lo tanto solo podían acceder a 1MB de memoria. Cuando la línea de los procesadores x86 fue creciendo, se fueron agregando líneas a el bus de direcciones para así pasar del limite de memoria de 1MB.

Dentro de un sistema computacional x86 existen adicionalmente otros controladores que ayudan al procesador a llevar a cabo las tareas del sistema. Esta el controlador de memoria que se encarga de poder escribir bytes de datos en la memoria RAM, controlar el contenido de datos en cada una de las DIMM o tarjetas de memoria, saber en que ubicación de memoria esta mapeada cada unidad de RAM y como acceder a ella. Por otro lado esta el controlador de I/O o de E/S, que se encarga de “mapear” los dispositivos de hardware a cierta dirección en memoria. Esta es la única forma de comunicarse con los dispositivos en modo protegido por lo que es necesario que se programen adecuadamente los números de los puertos a nivel de kernel.

Dentro de un procesador van codificadas muchas instrucciones especiales que ejecutan una cierta tarea. Por ejemplo, el procesador x86 cuenta con 4 registros de uso general para poder llevar a cabo tareas de estas instrucciones. Hay instrucciones especificas para cargar ciertas tablas como el GDT (Global Descriptor Table), LDT (Local Descriptor Table) y el IDT(Interrupt Descriptor Table). Otras instrucciones sirven para parar el procesador, hacer llamados a interrupciones, mover o copiar memoria de un registro a otro o a una dirección en memoria, etc. Un registro especial llamado IP (Instruction Pointer) lleva el contado de la siguiente instrucción que se ejecutara en el procesador. Hay otros registros que llevan conteo de segmentos de memoria, estados de procesador, etc.