

Sistemas informáticos

Arranque

Semana 22

Abril 2023

Cruz García, Iago



[Introducción](#)

[Sistema de arranque](#)

[BIOS](#)

[EFI/UEFI](#)

[Master Boot Record \(MBR\)](#)

[Bootloaderr](#)

[Kernel](#)

[Conclusiones](#)

Introducción

Hemos hecho un pequeño salto de parte física o *hardware* a parte lógica o *software* cuando estudiamos los Sistemas Operativos. Vamos a regresar parcialmente a una zona intermedia: sistema de arranque, formato de los datos, etc.

Esta semana veremos el arranque del Sistema Operativo, que es lo que ocurre al encender un computador y cómo es capaz una máquina de pasar de nada a poder visualizar los datos en pantalla.

Sistema de arranque

Vamos a ver que ocurre desde el momento que damos la señal de encendido en nuestro ordenador hasta que el Sistema Operativo se muestra en la pantalla.

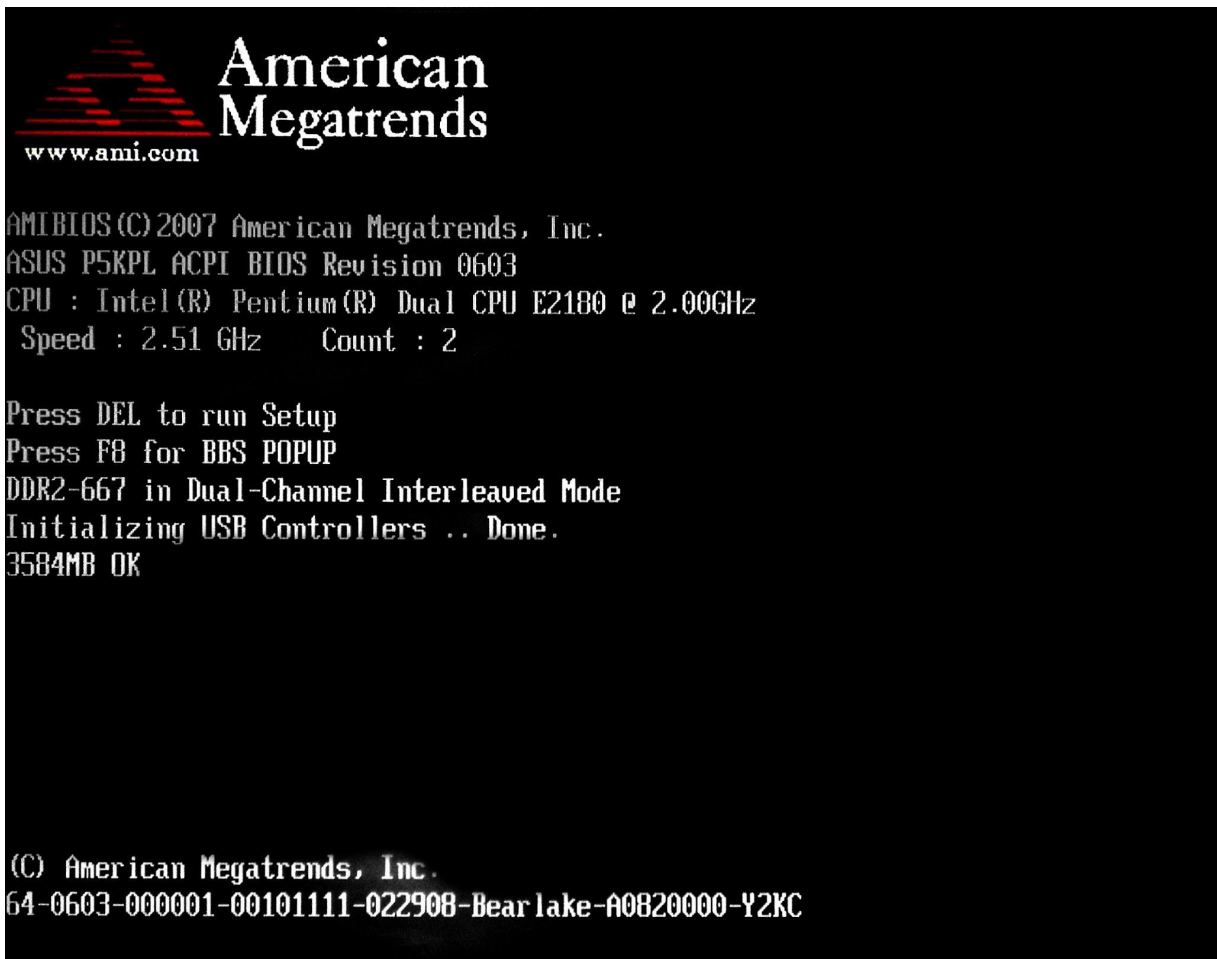
Para comenzar, debemos conocer lo que es la **BIOS**

BIOS

De **Basic Input/Output System** o Sistema básico de Entrada/Salida, es el *firmware* (“software” integrado en el componente físico electrónico para su manejo) que permite el manejo del hardware del equipo al Sistema Operativo. Este se encuentra físicamente dispuesto en la placa base, generalmente dispuesto en el **punto sur o southbridge**, se suele identificar por una pila de botón, pues se localiza cerca de la misma.

Entonces, ¿qué trabajo hace la BIOS? Tiene 2 funciones principales:

- **Comprueba** el estado de los componentes electrónicos del equipo a nivel físico: si están conectados correctamente, si la corriente es correcta, etc. Esto se conoce como **POST** o “Power On Self Test”



Ejemplo de POST(Fuente: [Wikipedia](https://es.wikipedia.org/wiki/BIOS))

- **Realizar la carga** del sistema de arranque del disco o memoria no volátil en memoria principal. Veremos en apartados posteriores que esto se denomina comúnmente **MBR** o Master Boot Record.

Además de estas dos funciones básicas, también permite **modificar o gestionar** estos elementos físicos a su nivel más básico. Generalmente accesible presionando la tecla definida por el fabricante de la placa base

(F2, F8 o Delete/Supr son teclas relativamente comunes). La interfaz que brinda la BIOS permitirá modificar los voltajes de los componentes, velocidad del procesador, orden de arranque de los sistemas de almacenamiento, etc.

EFI/UEFI

Con el paso del tiempo, nace una tecnología a raíz de la BIOS, **EFI** o Extensible Firmware Interface o Interfaz de firmware extensible, de la empresa fabricante de procesadores Intel. Esta se ve reemplazada por una versión abierta conocida como **UEFI** o Unified Extensible Firmware Interface.

Esta trae consigo una serie de cambios importantes acompañando la evolución tecnológica:

- Permite la arquitectura de 64 bits o 32 bits.
- Soporta programación en C de sus funciones, sin necesidad de reprogramar electrónicamente el chip.
- Es en sí misma un "Sistema Operativo" en miniatura, pues permite conexión de red, el uso del ratón o cambio de idioma.

Por el resto, las funciones que realizaba la BIOS son idénticas.

Master Boot Record (MBR)

Una vez la BIOS/UEFI comprueba el correcto funcionamiento del equipo en POST, realiza una llamada lanzando una señal a los dispositivos de almacenamiento en el orden indicado previamente. Esta señal sólo será leída por la primera unidad de almacenamiento que disponga de una

partición (localizada en la primera sección del disco o en los primeros bits de almacenamiento) conocida como **MBR** o **Master Boot Record**, una partición que almacena, como mínimo unos 512 bytes de información.

Esta señal enviará o localizará **MBR** en memoria principal para ser ejecutada, cuya única función es llamar al **Bootloader** o cargador de arranque del Sistema Operativo en cuestión (**Boot manager** en Windows o **GRUB** en UNIX/Linux).

Entonces, ¿por qué no llamar al cargador de arranque directamente? La respuesta es relativamente lógica y sencilla: la llamada tiene que ser estándar e independiente del Sistema Operativo. Como cada S.O tiene su estructura lógica, particiones, formatos de disco, etc., se creó un estándar de partición de arranque, **MBR**, que contendrá la información del S.O que se va a cargar, por lo que la **BIOS/UEFI** no tiene que conocer información específica de cada Sistema Operativo, si no que solo lanza una señal básica de carga.

Bootloader

Ya sabemos que el **bootloader** o cargador de arranque depende de cada Sistema Operativo. Pero tienen una cosa en común, pues independientemente de cual, en este paso será cuando se carga el Kernel del Sistema para poder realizar la conexión con los sistemas de Entrada/Salida del computador.

Una de las grandes ventajas de **UEFI** es poder realizar la señal de arranque directamente a **Bootloader**. Pero a pesar de ser **posible**, como muchos equipos actuales (sobretudo en el sector industrial y en el sector administrativo) todavía operan con **BIOS**, la secuencia descrita todavía está en vigor.

Kernel

Una vez se realiza la señal de carga de **Kernel**, casi de forma inmediata se cargan una serie de procesos importantes del Sistema Operativo, siendo la más importante la capa de abstracción física (**HAL**, Hardware Abstract Layer) que será la que permite la conexión con el entorno físico.

A su vez, una vez termina la carga del Kernel, este llamará a los programas o rutinas especificadas por el Sistema Operativo en orden de prioridad: sistema de ficheros, librerías de acceso a componentes, entorno gráfico, red, etc.

Conclusiones

Vemos que el sistema de arranque es una secuencia compleja de elementos que interactúan entre ellos, comenzando desde la señal de encendido hasta llegar a visualizar los primeros atisbos de nuestro Sistema Operativo. Observamos que el estándar de arranque es algo estático además, sin

novedades importantes desde **UEFI** en 2005, permanece en una especie de estasis debido a la sencillez y estandarización por parte de la industria.

<https://www.grs-software.de/sims/bios/phoenix/pages/>