

# Sistemas informáticos

Arranque Windows y Linux

Semana 23

Abril 2023

Cruz García, Iago



[Introducción](#)

[Windows](#)

[BIOS](#)

[MBR](#)

[Boot manager](#)

[Boot loader](#)

[Kernel](#)

[Linux](#)

[BIOS](#)

[MBR](#)

[GRUB](#)

[Kernel](#)

[Initdr](#)

[RUN](#)

# Introducción

La semana pasada echamos un vistazo al arranque de un computador independientemente de su Sistema Operativo disponible. Ahora vamos a ver los arranques en un entorno más concreto: Windows y Linux.

Además, vamos a trabajar con sus respectivos cargadores de arranque para comprobar que opciones de configuración están disponibles.

# Windows

Vamos a ver el arranque primero de Windows. Concretamente las versiones de Windows 10 en adelante.

## BIOS

Como sabemos, al encender el computador, se inicia la BIOS gracias a la pila y comienza el proceso de verificación de los componentes o **POST (Power On Self Test)**. Una vez el proceso se completa, se lanza la señal de carga a **MBR** o **Master Boot Record**, cargándolo desde el primer sector del disco a memoria principal.

## MBR

Seguidamente, se ejecuta el código de arranque, buscando la partición activa (cuál es la que contiene el MBR) buscando el cargador de arranque o **boot loader**, llamado **boot manager** en Windows, ejecutando la aplicación "bootmgr", localizada en la carpeta raíz del Sistema Operativo (generalmente C:\). No tiene una localización fija en todas las versiones. En las últimas de Windows 10 encontramos diferentes bootmgr (para diferentes arquitecturas de procesadores y S.O) en "C:\Windows\WindowsSxS"

## Boot manager

Windows divide en dos tareas la carga del arranque de sus sistema. El primero el **boot manager**. Su trabajo será leer el **BCD** o Boot Configuration Data, el fichero que tiene la información sobre el gestor de arranque de Windows: Tiempo de espera, pantalla de selección de Sistema Operativo, sistema por defecto...

## Boot loader

Una vez se carga la información del BCD en memoria, lanza **Windows Boot Loader**, el cargador de arranque específico de sistema de Microsoft. En concreto, localizado en la carpeta "C:\Windows\System32\winload.exe", este se encargará de cargar el Kernel en memoria.

## Kernel

El **Kernel** ahora cumplirá una serie de tareas. Será el engranaje principal de cualquier Sistema Operativo, siendo en sistemas de Microsoft, **Windows NT Os Kernel** lo primero que se carga como tal de nuestro sistema. Este cargará primero la capa de abstracción de hardware o **HAL** y a continuación los drivers de los dispositivos y montará los sistemas de ficheros para poder tener acceso a nuestros dispositivos de almacenamiento. Por último lanzará el **SMSS o Session Manager SubSystem**, el programa que se encarga de manejar las sesiones de los usuarios, administrador, etc.

Este lanzará otro subsistema, el encargado de la interfaz gráfica **wind32k.sys**, creará los subprocesos de sesiones, lanzará los servicios de inicio de sesión (aquellos marcados como "BOOT\_READY").

Por último, se cargarán los scripts o elementos de inicio de sesión indicados por el usuario, incluidos aquellos programas propios de Windows como explorer.exe.

Todos estos procesos llevan a la ya conocida pantalla de escritorio de los sistemas Windows, cargado y listo para ser usado.

## Linux

Si bien Windows tiene un sistema de cargado muy fragmentado, a nivel de Kernel sobretodo, Linux fragmenta las tareas de otra forma, algo más directa y menos unilateral, provocando que el arranque sea funcional rápidamente aunque la protección contra errores se vea afectada, pues la carga del sistema podría ser incorrecta o tener algún error pero el Sistema Operativo se cargará igualmente, aún faltando funciones o estando erradas.

Concretamente analizaremos la secuencia de arranque de un sistema Ubuntu 22.04.

## BIOS

Al igual que en los sistemas Windows, la BIOS es el primer elemento en cargar al encender nuestro computador. E igual que antes, lanza la señal a **MBR** para cargar el arranque

## MBR

En este punto ya encontramos una diferencia. El sector de arranque de Linux se considera un fichero en sí mismo, lo podemos localizar en `"/dev/sda"`. Este contiene 512 bytes de información:

- La información de la carga en los primeros 446 bytes.
- La tabla de particiones en los 64 bits siguientes.
- Los 2 últimos bytes son para la validación de la carga.

La tarea del **MBR** será la carga de **GRUB**.

## GRUB

El cargador de arranque de Linux, **GRUB o Grand Unified Bootloader**, que será el encargado de permitir la modificación de qué sistema lanzar, qué opciones de carga tenemos, etc. Se puede modificar manualmente, localizado por defecto en `"/boot/grub/grub.cfg"`, modificable en `"/etc/default/grub"` o con la aplicación [grub-customizer](#).

Actualmente se utiliza la versión [GRUB2](#), mejorada del primero, pero por legado sigue tomando el primer nombre.

Una vez seleccionado el sistema a cargar, GRUB lanzará el **Kernel**.

## Kernel

Será el encargado, como en Windows, de administrar la capa abstracta para comunicarse con el hardware. Pero a diferencia del anterior sistema, en Linux no cargará los drivers el mismo. Como hemos dicho, todo en Linux es un fichero, eso incluyen los propios dispositivos, por lo que el Kernel tiene la tarea de montar el sistema de ficheros antes de nada y procede a ejecutar el programa “/sbin/init” conocido también como **Initrd**, de **Initial RAM Disk**.

Es un programa que montará un sistema de ficheros virtual para que el Kernel pueda leer los ficheros antes de montar el propio sistema. Permitirá entonces cargar los sistemas de drivers de los elementos físicos para su comunicación, además de lanzar dicho programa.

## Initdr

Mientras el sistema se monta y el Kernel realiza sus tareas, **Init** ejecutará **Runlevel o Run**, el programa que se encargará de lanzar todos los programas. A la hora de lanzar este **Run**, se puede hacer en diferentes niveles en el fichero “/etc/inittab”:

- 0: Apagado



- 1: Usuario Único
- 2: Multiusuario sin NFS (compartir ficheros)
- 3: Multiusuario completo
- 4: Sin uso
- **5: Multiusuario con interfaz gráfica. (Por defecto en Ubuntu)**
- 6: Reinicio.

Cuando lance **Run**, también le dirá en qué nivel se está ejecutando.

## Run

**Run** tomará el nivel de **Initdr** y accederá al directorio correspondiente `"/etc/rc.d/rc'nivel'.d/".` En dicho directorio estará los programas indicados para el inicio del sistema (marcados con una S de Startup) y los programas que se usan en el cierre o apagado (marcados con una K o Kill). Los números en los programas indican el orden de carga. Si ambos comparten un número, la carga se realizará arbitrariamente o al azar.

```

K01speech-dispatcher  S01cups-browsed      S01plymouth
S01acpid              S01dbus              S01pulseaudio-enable-autospawn
S01anacron            S01gdm3              S01rsync
S01apport             S01grub-common       S01saned
S01avahi-daemon       S01irqbalance        S01spice-vdagent
S01bluetooth          S01kerneloops        S01unattended-upgrades
S01console-setup.sh  S01named             S01uidd
S01cron               S01open-vm-tools     S01whoopsie
S01cups               S01openvpn

```

Diferentes programas de inicio y fin de sistema