Arranc BIOS vs UEFI

Actualment podem trobar al mercat PCs que proporcionen de fàbrica un dels següents dos tipus d'arrencada: l'arranc per BIOS i l'arranc per UEFI. El 1º tipus és el clàssic però el 2º és el el futur. Expliquem-los:

*__BIOS__ (Basic Input Output System) : Firmware específic emmagatzemat de fàbrica en un chip de la placa base (concretament, a la seva memòria ROM). Bàsicament és un conjunt de funcions i rutines de baix nivell que serveixen per reconèixer de manera fonamental els components més importants del PC (memòria, discos, alguns perifèrics, etc). És el primer software que s'executa en encendre's l'ordinador, abans de qualsevol altra cosa.

La BIOS ofereix un panell de control des d'on es poden modificar diversos aspectes del seu funcionament. Aquest panell de control només està accessible durant els primers segons de l'arranc pitjant una tecla determinada (que pot ser diferent segons el fabricant de placa base...aquesta tecla se sol indicar en els primers missatges que apareixen per pantalla just en arrencar però sol ser la tecla SUPR. Altres tecles possibles poden ser ESC, F1, F2, F8, F9, F10, F11 o F12). En aquest panell de control es pot canviar, per exemple l'hora del rellotge intern sistema (alimentat per una pila botó quan no li arribi corrent elèctric extern) o bé l'ordre d'arrencada entre els diferents dispositius disponibles (discos durs, CD/DVDs, xarxa, etc). Aquests canvis es poden guardar per tal de què afectin a les arrencades següents

NOTA: Concretamente hay una rutina incluida en la BIOS que es la que se pone en marcha la primera de todas: es la llamada POST (Power-On Self Test), que no es más que una comprobación y inicialización eléctrica para asegurarse que todos los componentes de la placa funcionan correctamente y que todo está en condiciones de funcionar. Este proceso es el responsable de todos los mensajes que nos aparecen en el tiempo que transcurre justo desde que arrancamos el ordenador hasta que vemos el mensaje de arranque del sistema operativo que tengamos instalado en nuestro disco duro. En esos mensajes se puede observar el reconocimiento del tamaño total de la memoria RAM instalada en el sistema, el modelo de microprocesador instalado y su frecuencia, los discos duros detectados,etc. Si en este proceso se detectara algún error grave (la memoria RAM está defectuosa, no se encuentra tarjeta gráfica,etc),, la BIOS mostrará un mensaje de error por pantalla, pero si el error es anterior a la puesta en funcionamiento del sistema de video (es decir, antes del primer punto), entonces emitirá una serie de pitidos por el altavoz del sistema que indican el tipo de error. Según la combinación de pitidos largos y cortos, se podrá saber el tipo de error detectado. Esta combinación de pitidos no es estándar: cada fabricante de BIOS tiene una combinación propia. Para saber los codigos POST de error de tu BIOS particular, deberás ir a la web del fabricante, o bien, consultar la web http://www.bioscentral.com.

NOTA: Después del POST, el siguiente paso es cargar en memoria los controladores de dispositivos para los componentes básicos del sistema (teclado, disco duro, controladores placa base, etc). De hecho, en realidad podemos definir la BIOS como un conjunto de programas muy pequeños cuya tarea principal es interpretar los datos provenientes de cada uno de los dispositivos del PC y convertirlos en comandos que pueda utilizar la CPU. Es decir, por sí misma, la CPU es incapaz de comunicarse con los otros componentes del sistema: la CPU sólo dispone del juego de instrucciones que es capaz de utilizar. Por lo tanto, la BIOS, tal como su nombre indica, ofrece una interfaz de muy bajo nivel para que la CPU pueda reconocer y trabajar con los distintos dispositivos presentes en el sistema. En resumen, la BIOS es una especie de almacén de “puertas” diferentes donde la CPU elige una concreta para comunicarse al nivel más hardware con un determinado componente del sistema.

NOTA: La última tarea que realiza la BIOS es ir a buscar (allí donde se le haya indicado, que normalmente será un disco duro local pero no tiene por qué) un programa muy especial llamado "gestor de arranque" (boot loader) y ejecutarlo. A partir de aquí la BIOS finaliza su trabajo y el "testigo" pasa a este programa, del cual hablaremos más adelante

*__UEFI__ (Unified Extensible Firmware Interface) : Firmware emmagatzemat de fàbrica en un chip de la placa base (concretament, a la seva memòria ROM) que pretén substituir al ja antic estàndard BIOS. Per tant, és igualment el 1r software que s'executa en encendre's l'ordinador, abans de qualsevol altra cosa. Igualment, també ofereix un panell de control (accessible només durant els primers segons de l'arranc) dins del qual es poden modificar diversos aspectes del seu funcionament.

Els avantatges d'usar UEFI en comptes de BIOS són vàries: un sistema UEFI pot, per exemple, reconèixer discos durs i particions més grans, el seu panell de control sol ser més intuïtiu gràcies a ser gràfic i possibilitar l'ús del ratolí, pot usar funcionalitats criptogràfiques i d'autentificació per xarxa, suporta extensions emmagatzemades en medis no-volàtils, ofereix un entorn "shell" per poder executar aplicacions EFI tals com utilitats de diagnòstic, reparació, actualització, etc. A més, UEFI és independent de l'arquitectura de la CPU (no com la BIOS, que depenia de la x86(_64)) així que és extensible a altres tipus de màquines com les d'arquitectura ARM.

NOTA: L'especificació original (anomenada EFI) va ser dissenyada per Intel però posteriorment va ser assumida per la resta de fabricants de plaques base del món i s'hi va afegir la "U" d'"unified".

NOTA: A la pràctica, moltes UEFI ofereixen un "BIOS Legacy mode" (escollible des del seu panell de control) que permet "transformar" el seu comportament per tal de què actuin com si fossin BIOS clàssiques. Però no en parlarem.

Suposant que volem tenir instal.lat algun sistema operatiu dins d'alguna partició d'un disc dur local del nostre PC, tot seguit explicarem pas a pas el seu procés d'arrencada segons si tenim una màquina BIOS o UEFI:

Arranc per BIOS

*Requisit previ: el disc dur ha de tenir una taula de particions de tipus "msdos"
NOTA: El tipus de taula de particions es pot establir amb l'aplicació gnome-disks, gparted o similars. Cal tenir en compte que (re)establint-ho, però, s'esborraran totes les dades que hi poguessin haver al disc en qüestió.

*Quan un sistema operatiu s'instal.la en una partició, a més de gravar tots els seus fitxers allà, fa això de forma permanent per les seves següents arrencades :

-Converteix a "activa" (és a dir, "arrencable") aquesta partició.
NOTA: Aquesta característica es pot consultar/afegir/treure manualment amb l'aplicació gnome-disks, gparted o similars

-Grava en el primer sector del disc dur (qué és un lloc fora de qualsevol partició) un programa especial anomenat "gestor d'arranc" (boot loader), la tasca del qual és simplement trobar i arrencar el sistema operatiu ubicat a la partició activa (això vol dir a la pràctica que, quan s'instal.la un sistema operatiu, sempre s'autoimposarà com el sistema a arrencar per defecte a la màquina). El gestor d'arranc que se sol instal.lar amb les distribucions Linux és un anomenat Grub2, però n'hi ha més a escollir com per exemple Syslinux. El gestor d'arranc que s'instal.la amb els sistemes Windows és un anomenat NTLDR.

NOTA: Els sectors dels discs durs són el mínim tros que es pot llegir/escriure pel hardware (són com els "àtoms" d'informació que a la pràctica es poden manipular). Actualment el seu tamany és de 512 bytes. Com que el primer sector del disc dur (el nº0) és força especial perquè és on es troba el gestor d'arranc, té fins i tot un nom propi: Master Boot Record (MBR)

NOTA: Al MBR no es troba només el gestor d'arranc (que és un programa) sinó un conjunt de bits anoments "taula de particions" els quals bàsicament, ofereixen tota la informació necessària per a què el gestor d'arranc reconegui quantes, quines, on estan i de quin tamany són les particions existents al disc dur i sàpiga trobar la que és l'activa. És a dir, el gestor d'arranc bàsicament el que fa és llegir la taula de particions i actuar en conseqüència.

NOTA: El fet de què el MBR sigui tan petit fa que el tamany de la taula de particions sigui limitat. Això fa que només s'hi puguin definir fins a quatre particions (les anomenades particions "primàries"). Si se'n volguessin tenir més, es pot fer però amb un "truc" que consisteix en repartir la taula de particions dins de les pròpies particions en forma de llista anidada; és aquí quan apareix el concepte de "partició extesa" i "particions lògiques".

NOTA: De fet, que el MBR sigui tan petit fa que sovint el propi codi del boot loader tampoc no pugui ubicar-se sencer dins seu sinó que només hi estigui el "tros" suficient per començar, guardant-se la "2ªpart" de codi dins de la partició activa.

*Si s'instal.len diferents sistemes operatius, els gestors d'arranc es van matxacant un rera l'altre dins del MBR (i la partició activa va canviant cada cop). Aquí poden haver-hi diferents casos:

-**Instal.lar Windows i després Linux:** El Grub respecta el NLTDR integrant "dins seu" tota la informació sobre l'arranc dels dos sistemes i mostrant-la en forma de menú. Aquest menú serà el que permeti a l'usuari escollir quin sistema vol arrencar.

-**Instal.lar Linux i després Windows:** El NTLDR matxaca el Grub com si no hagués existit mai, de forma que es passarà a arrencar directament Windows. La partició de Linux però, romandrà intacta. La manera de poder tornar a arrencar Linux és reinstal.lant el Grub al MBR, reproduïnt llavors el cas anterior (amb un LiveCD de qualsevol Linux es pot fer això : ho veurem a classe més endavant)

-**Instal.lar Linux i després un altre Linux**: El Grub nou respecta el Grub anterior integrant "dins seu" tota la informació sobre l'arranc dels dos sistemes i mostrant-la en forma de menú

-**Instal.lar Windows i després un altre Windows:** El NTLDR nou respecta el NTLDR anterior integrant "dins seu" tota la informació sobre l'arranc dels dos sistemes i mostrant-la en forma de menú NOMÉS si el NTLDR nou és més modern que el NTLDR anterior. Si no fos el cas, estaríem en un cas similar al descrit en segon lloc, i la manera de reparar aquesta situació és reinstal.lant el NTLDR modern un altre cop.

*La funció del gestor d'arranc és trobar a la partició activa una parella kernel+initrd i executar-la, a més de passar-li un conjunt de paràmetres que defineixen aquesta execució. D'entre els possibles paràmetres que es poden passar al kernel, ún d'imprescindible és el paràmetre "root", el qual indica la partició (/dev/sdxn) que el kernel haurà de muntar a la partició arrel ("/") -això implica que haurà de poder reconèixer el seu sistema de fitxers via l'arxiu initrd adient- perquè allà es troba la resta del sistema, començant pel procés INIT (també anomenat PID nº1), el qual, actualment, a la majoria de distribucions actuals, és el programa Systemd. Normalment aquesta partició "arrel" sol coincidir amb la partició activa.

Initrd viene de “initial ramdisk”. Un disco RAM inicial es un pequeño sistema de archivos que es cargado en la memoria RAM y montado cuando el kernel arranca, antes de que la partición raíz donde se halla el resto del sistema sea montada (normalmente en "/"). La razón para usar un initrd es porque allí residen un conjunto de módulos -"plugins"- del kernel que deben ser cargados antes del montaje de esa partición raíz. Esos módulos son necesarios, principalmente, para dotar al kernel de la capacidad de reconocer el sistema de archivos usado precisamente por la partición raíz (ext4, jfs, xfs), o el controlador al que el disco duro está ligado (SCSI, RAID, etc.). Podemos entender, pues, que el archivo "initrd" es como una "bolsa de drivers" que el kernel necesita para trabajar con el/los dispositivo/s de almacenamiento externo donde se supone que reside el resto del sistema a arrancar. Como hay tantas opciones diferentes disponibles en los kernels modernos de Linux, no resulta práctico intentar hacer un kernel inmenso que las cubra todas o muchos kernels para cubrir las necesidades de cada uno o: es mucho más flexible armar un kernel ligero genérico y configurar diferentes módulos (ubicados en diferentes archivos initrd) para él. Por otro lado, the initramfs only needs to contain the modules necessary to access the root filesystem; it does not need to contain every module one would ever want to use because the majority of modules will be loaded later on by udev, during the init process.

Arranc per UEFI

*Requisit previ: el disc dur ha de tenir una taula de particions de tipus "gpt" (GUID Partition Table). Una conseqüència de fer servir aquest tipus de taula de particions és que ja no existeix el concepte de partició extesa ni el de partició activa: totes les particions són "primàries".
NOTA: El tipus de taula de particions es pot establir amb l'aplicació gnome-disks, gparted o similars. Cal tenir en compte que (re)establint-ho, però, s'esborraran totes les dades que hi poguessin haver al disc en qüestió.

*Quan un sistema operatiu s'instal.la en una partició, a més de gravar tots els seus fitxers allà, copia en una partició especial anomenada ESP (EFI System Partition) el gestor d'arranc pertinent (que ara és simplement un fitxer binari amb extensió ".efi", moltes vegades juntament amb el seu arxiu de configuració, que sol ser de text editable). És a dir, la ESP substituex ara al MBR però ara proporciona la possibilitat de tenir múltiples gestors d'arrancs instal.lats a la vegada sense que es "matxaquin" uns als altres ja que hi ha espai per tots. Això sí, només es podrà fer-se servir un durant l'arrencada; la forma d'escollir quin gestor es vol s'explicarà a l'apartat següent.
NOTA: Els gestors d'arranc que podem fer servir en un sistema UEFI o bé són una variant "UEFI" dels gestors d'arranc ja coneguts ("grub.efi", "syslinux.efi"...) o bé de nous no existents a sistemes BIOS, com ara "systemd-boot"

La ESP és una partició estàndar que pot existir prèviament o bé crear-se durant la instal.lació d'un sistema operatiu. No obstant, és obligatori marcar-la com a tal per a què la UEFI la pugui trobar; aquesta "marca", (que es pot aplicar amb aplicacions com *gnomedisks*, *gparted* o similars, o bé durant el mateix procés d'instal.lació del sistema operatiu a la pantalla de particionament del disc que ofereixi l'assistent d'instal.lació), consisteix en un identificador anomenat GUID que contenen totes les particions GPT però que en el cas de la ESP ha de tenir un valor molt concret: C12A7328-F81F-11D2-BA4B-00A0C93EC93B
A més a més, la ESP ha de tenir aquests altres requisits:

-Ha de tenir un tamany d'un parell de centenars de megabytes com a mínim

-Ha d'estar formatejada amb el sistemes de fitxers FAT32 (encara que moltes vegades només marcant-la com a "ESP" això ja es fa automàticament)

-Ha de tenir com a un punt de muntatge associat la carpeta /boot (antigament es feia servir /boot/efi). Dins d'ella apareixerà una subcarpeta anomenada EFI, a l'interior de la qual hi haura una subcarpeta per cada gestor d'arranc instal.lat

*Si s'instal.len diferents sistemes operatius, s'acumularan diferents gestors d'arranc a l'ESP; la selecció del gestor específic a utilitzar es pot fer llavors dins del panell de control de la pròpia UEFI. En realitat, el que es fa dins d'aquest panel de control és manipular les anomenades "variables EFI", variables que estableixen i guarden en una memòria hardware especial anomenada NVRAM la configuració actual de la UEFI (la qual, a més d'establir quin gestor d'arranc usar per defecte, defineix l'ordre en què es provaran si el gestor d'arranc previ i moltes altres coses més). Aquestes variables EFI també es poden manipular des d'un sistema Linux ja funcionant mitjançant la comanda *efibootmgr...* tal com veurem més endavant.

*La funció del gestor d'arranc és trobar una parella kernel+initrd i executar-la, a més de passar-li un conjunt de paràmetres que defineixen aquest arranc. Aquesta parella generalment estarà ubicada dins de la mateixa partició ESP (aquesta restricció depèn del gestor d'arranc usat). D'entre els possibles paràmetres que es poden passar al kernel, ún d'imprescindible és el paràmetre "root", el qual indica la partició (/dev/sdxn) que el kernel haurà de muntar a la partició arrel ("/") -això implica que haurà de poder reconèixer el seu sistema de fitxers via l'arxiu initrd adient- perquè allà es troba la resta del sistema, començant pel procés INIT (també anomenat PID nº1), el qual, actualment, a la majoria de distribucions actuals, és el programa Systemd. Aquesta partició arrel pot estar formatejada en qualsevol sistema de fitxers que el kernel+initrd pugui reconèixer.

NOTA: La UEFI sol incorporar una característica anomenada "Secure boot" activada de fàbrica que pot fer que algunes distribucions de Linux no puguin arrencar des de LiveCD/LiveUSB i, per tant, que no es puguin ni tan sols instal.lar al sistema. En aquests casos, caldrà entrar a la UEFI i deshabilitar aquesta característica. En teoria, el "Secure boot" serveix per verificar l'autenticitat de les aplicacions EFI mitjançant una firma digital ja reconeguda per la UEFI (aquí està el problema: si la firma d'una aplicació EFI concreta no apareix a la "llista" de firmes reconegudes per la UEFI de la màquina que estem fent servir -o si directament l'aplicació EFI no està firmada, aquesta es negarà a executar-la. Per més informació, llegiu http://www.rodsbooks.com/efi-bootloaders/secureboot.html

NOTA: Els kernels Linux moderns són capaços d'executar-se directament per un gestor d'arranc UEFI. Això vol dir que poden actuar com si fossin una aplicació UEFI més (per això han de tenir l'extensió ".efi"), al mateix nivell que els gestors d'arranc o una shell EFI. Això a la pràctica vol dir que podríem configurar via les variables EFI com a programa directament a executar per la UEFI un determinat kernel (+ initrd), obviant completament l'ús de cap gestor d'arranc (però perdent, llavors, la capacitat de veure un menú d'opcions). A aquest tipus d'arranc més directe se'n diu "EFISTUB"

NOTA: També és possible que la configuració de la UEFI no apunti a cap aplicació EFI en particular dins de cap ESP sinó simplement tingui configurat "anar" a un disc. En aquest cas, serà la UEFI per ella mateixa qui buscarà en aquell disc alguna partició que sigui ESP i, si la troba l'aplicació EFI per defecte, que sempre s'anomena "BOOTX64.EFI" i l'espera trobar a la carpeta "EFI/BOOT" dins de l'ESP trobada.

El sistema d'arranc que utilitzen els fitxers iso (propi dels CDs/DVDs) és diferent del dels discos durs, encara que també es distingeix entre arrencada tipus BIOS i tipus UEFI.

Arranc per BIOS

Els dispositius òptics (CDs/DVDs) no disposen de cap MBR; això vol dir que el gestor d'arranc s'ha d'ubicar entre la resta de dades grabades. Per a què la BIOS pugui trobar i executar el gestor d'arranc en aquell lloc concret (el qual, insistim, està guardat dins una carpeta "normal") -és a dir, per a què el fitxer iso sigui arrencable- el format d'aquest fitxer iso ha de ser "especial": ha de ser de tipus "El Torito" (que és una variant del format ISO9660). L'estandard El Torito (l'especificació oficial del qual és a http://download.intel.com/support/motherboards/desktop/sb/specscdrom.pdf) precisament el que diu és en quin sector del CD/DVD s'ha de trobar el gestor d'arrencada.

Encara que el mecanisme d'arrencada en els CDs/DVDs i discos durs es troba en diferents llocs, afortunadament és possible afegir els dos mecanismes a un mateix fitxer iso: això són els fitxers iso "híbrids": fitxers iso que són El Torito i que a més contenen un MBR. Això permet fer servir el gravar el mateix arxiu iso en un CD/DVD o en un USB (que a efectes pràctics funciona com un disc dur) i aconseguir que arrenquin tots dos indistintament. Els fitxers iso de les distribucions més importants són híbrids.

Arranc per UEFI

Els dispositius òptics continuen utilitzant el format ElTorito per indicar al UEFI on es troba el gestor d'arrencada a utilitzar (que en aquest cas resulta ser el mateix lloc que es fa servir pels USB: la carpeta EFI/BOOT). En realitat, per aconseguir això, ElTorito fa un "truc" que consisteix en "enganyar" a l'UEFI fent-li veure que aquesta carpeta és una partició ESP embeguda dins del sistema matriu ISO9660. Aquest engany és possible gràcies a la presència del fitxer especial EFI/BOOT/images/efiboot.mgr. Així doncs, en un sistema UEFI els gestors d'arrencada poden ser els mateixos ja estiguem arrencant el fitxer iso gravat en un llapis USB o en un CD/DVD perquè tots estan ubicats en el mateix lloc dins de la carpeta EFI/BOOT.