

Tema 1.3

Sistema Operativos (SSOO)

Introducción a Linux

Índice

- Introducción a Linux
- GNU/Linux y sus distribuciones
- Arranque del ordenador
- Arranque secundario
- Arranque del SO

Introducción a Linux

Algunas cuestiones importantes:

- ¿Quién usa Linux?
- ¿Qué es Linux?
- ¿Qué es Ubuntu, Debian, Red Hat, Suse...? ¿Cuántos Linux hay?
- ¿Puedo usarlo en casa para escuchar música, ver películas, chatear, etc.? ¿Echaré algo en falta?
- ¿Por qué tendría que cambiarme a Linux?
- ¿Cómo se aprende Linux?

GNU/Linux

- El sistema operativo GNU es un sistema completo de software libre compatible con Unix. El término GNU proviene de «**GNU No es Unix».**
- Richard Stallman escribió el anuncio inicial del Proyecto GNU en setiembre de **1983**.
- Al principio de los años ochenta, casi todo el software era privativo, lo que significa que tenía dueños que prohibían e impedían la cooperación entre usuarios. De ahí que fuera necesario el Proyecto GNU.
- Si no existe un sistema operativo libre, ni siquiera se puede comenzar a usar un ordenador sin recurrir al software privativo. Así, el primer elemento en la agenda del software libre tenía que ser un sistema operativo libre.
- El sistema operativo era compatible con Unix porque el diseño en general ya estaba probado y era portable, y porque la compatibilidad facilitaba a los usuarios de Unix el cambio de Unix a GNU.

GNU/Linux

- Un sistema operativo similar a Unix incluye un núcleo, compiladores, editores, procesadores de texto, software de correo, interfaces gráficas, bibliotecas, juegos y muchas otras cosas.
- En 1990 ya se contaba con los componentes principales excepto uno, el núcleo.
- En 1991, Linus Torvalds creó Linux basándose en **Minix**, una pequeña implementación de UNIX para PC, el cual fue escrito totalmente en C por A. Tannenbaum.
- Linus anunció su primera versión oficial 0.02, el 5 de Octubre de 1991. En 1992 se convirtió en software libre.

Fuente original: <https://www.gnu.org/distros/free-distros.html>

Linux

- Consta básicamente del Kernel, el shell y el compilador gcc.
- Cuando se habla de Linux, realmente se hace referencia al **Kernel** o núcleo del sistema. El resto de aplicaciones que corren sobre ese kernel (formando usualmente una distribución) debería llamarse GNU / Linux.

La idea de usar un pingüino fue de Alan Cox, uno de los desarrolladores del kernel de Linux. Larry Ewing se encargó de dibujarlo usando GIMP, en el año 1996.
Su nombre es Tux (Torvalds Linux)












Fuente original: <https://www.gnu.org/distros/free-distros.html>

Distribuciones 100% GNU/Linux

- La palabra «libre» en «software libre» se refiere a libertad, no al precio. Se puede o no pagar un precio por obtener software de GNU.
- Una vez que se obtiene el software, se tienen cuatro libertades específicas para usarlo:
 - La libertad de ejecutar el programa como se desee;
 - La libertad de copiar el programa y dárselo a amigos o compañeros de trabajo;
 - La libertad de cambiar el programa como se desee mediante el acceso completo al código fuente
 - La libertad de distribuir una versión mejorada, ayudando así a construir la comunidad (si se redistribuye software de GNU, se puede cobrar una tarifa por el acto físico de efectuar la copia, o bien regalar copias)



Fuente original: <https://www.gnu.org/distros/free-distros.html>

Distribuciones 100% GNU/Linux

Distribución	Breve descripción
	Dragora GNU/Linux-Libre, una distribución GNU/Linux independiente basada en la idea de simplicidad.
	Dyne:bolic, una distribución de GNU/Linux que pone especial énfasis en la edición de audio y vídeo. Es una distribución «estática», que normalmente se ejecuta desde un CD autónomo. Debería utilizarse sin conexión a la red, ya que no recibe actualizaciones de seguridad.
	gNewSense, una distribución de GNU/Linux basada en Debian, con el patrocinio de la FSF.
	Guix System, una distribución GNU/Linux avanzada basada en GNU Guix, un gestor de paquetes puramente funcional para el sistema GNU.
	Hyperbola GNU/Linux-libre, una distribución enfocada en la simplicidad y con soporte a largo plazo, basada en Arch GNU/Linux.
	Parabola GNU/Linux-libre, una distribución basada en Arch que prioriza la gestión sencilla del sistema y de los paquetes.
	PureOS, una distribución de GNU basada en Debian enfocada en la privacidad, la seguridad y la comodidad.
	Trisquel, una distribución de GNU/Linux basada en Ubuntu y destinada a las pequeñas empresas, usuarios domésticos y centros educativos.
	Ututo S, una distribución de GNU/Linux 100 % libre. Fue el primer sistema GNU/Linux completamente libre reconocido por el Proyecto GNU.

Distribuciones 100% GNU/Linux – Ligeras

- Distribuciones para dispositivos con recursos limitados, como por ejemplo un *router* inalámbrico.
- Estas distribuciones no son autónomas sino que deben poder ser desarrolladas y compiladas sobre una de las distribuciones libres completas de la lista anterior, posiblemente con el auxilio de herramientas de desarrollo libres distribuidas junto a la misma.

Distribución	Breve descripción
 The libre embedded GNU/Linux distro	libreCMC es una distribución GNU/Linux embebida para dispositivos con recursos muy limitados. Aunque está pensada principalmente para <i>routers</i> , ofrece soporte para una amplia gama de dispositivos y situaciones de uso. En 2015, LibreWRT se fusionó con libreCMC.
 ProteanOS	ProteanOS es una nueva distribución, pequeña y veloz, para dispositivos embebidos. Posee una funcionalidad de configuración que permite configurar los paquetes binarios en el momento de la compilación o en tiempo de ejecución, según los diferentes tipos de hardware y usos.

Fuente original: <https://www.gnu.org/distros/free-distros.html>

Distribuciones GNU/Linux – Sin avalar

Excepto donde se indica, ninguna de las distribuciones que se incluyen en la siguiente lista respeta las pautas en al menos dos aspectos importantes:

- No adoptan ninguna política para incluir únicamente software libre, ni para eliminar el software que no sea libre cuando se detecta. La mayoría de estas distribuciones no tienen en absoluto una **política clara sobre qué software aceptan o rechazan**. Hay distribuciones que sí tienen una política, pero lamentablemente no son lo suficientemente estrictas, como se explica a continuación.
- El núcleo que distribuyen (en la mayoría de los casos, Linux) incluye **blobs: piezas de código compilado distribuidas sin el código fuente**, generalmente firmware para hacer funcionar algún dispositivo.

Distribuciones GNU/Linux – Sin avalar

Arch GNU/Linux

Canaima

CentOS

Debian GNU/Linux

Fedora

Gentoo GNU/Linux

Mandriva GNU/Linux

Manjaro GNU/Linux

Mint GNU/Linux

openSUSE

Red Hat GNU/Linux

Slackware

SteamOS

SUSE GNU/Linux Enterprise

Tails

Ubuntu GNU/Linux

Otros:

Android

Sistema BSD

Chrome OS

Haiku

/e/

LineageOS

ReactOS

<https://www.gnu.org/distros/common-distros.html>

SO para la asignatura



debian



MINIX 3



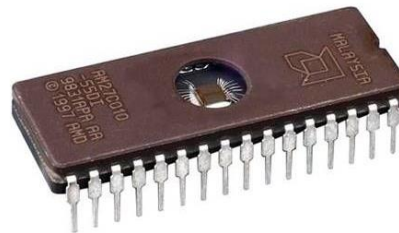
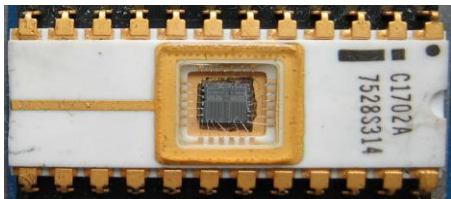
[Descargar SO](#)

Arranque del ordenador i

- Cada Pentium contiene (en su tarjeta madre) un programa conocido como **BIOS** (Basic Input Output System, Sistema básico de entrada y salida) del sistema.
 - El **BIOS** contiene software de E/S de bajo nivel, incluyendo procedimientos para leer el teclado, escribir en la pantalla y realizar operaciones de E/S de disco, entre otras cosas.
- Hoy en día está contenido en una RAM tipo flash que es no volátil pero el SO puede actualizarla cuando se encuentran errores en el BIOS.
- La **BIOS** (*firmware*) realiza la inicialización básica del hardware (**POST**: encendido y autocomprobación (**power on self test**)) y pasa el control del sistema al siguiente paso.
 - Primero hace pruebas para ver cuánta **RAM** hay instalada y si el **teclado** junto con otros dispositivos básicos están instalados y responden en forma correcta.
 - Empieza explorando los buses ISA y PCI para detectar todos los dispositivos conectados a ellos.

Arranque del ordenador ii

- Un firmware es un software que maneja físicamente al hardware durante el arranque.
- Está cargado en la memoria ROM (Read-Only Memory) del ordenador, no en el disco duro.
- Read-only memory (ROM) es un tipo de memoria no volátil que se usa en equipos y otros dispositivos electrónicos.
- Permite solo la lectura de la información y no su escritura.
- Las ROM modernas sí se pueden reprogramar y se llaman EPROM (Erasable Programmable Read-Only Memory)
- Entre ellas también las flash ROM.



Arranque del ordenador iii

- El **BIOS** determina el dispositivo de arranque, para lo cual prueba una lista de dispositivos almacenada en la memoria CMOS.
 - El usuario puede cambiar esta lista si entra a un programa de configuración del BIOS, justo después de iniciar el sistema.
 - Por lo general, se hace un intento por arrancar: Disco flexible, Unidad de CD, USB, Red, HDD, otros.
- La BIOS busca el dispositivo de inicio y pasa el control al primer sector físico del dispositivo conocido como Master Boot Record o **MBR**.
- Los dos primeros sectores del disco duro (los primeros 512 bytes) contienen el **MBR**. El **MBR** es básicamente una tabla de particiones cuyas entradas son las particiones primarias y lógicas que hay en el disco duro.
- Este sector contiene un programa que por lo general examina la tabla de particiones al final del sector de arranque, para determinar qué partición está activa.

Arranque secundario

- Después se lee un cargador de arranque secundario de esa partición. Este cargador lee el sistema operativo de la partición activa y lo inicia.
- El sistema operativo consulta al BIOS para obtener la información de configuración.
- Para cada dispositivo, comprueba si tiene el driver correspondiente. De no ser así, pide al usuario que inserte un CD-ROM que contenga el driver (suministrado por el fabricante del dispositivo).
- Una vez que tiene los drivers de todos los dispositivos, el sistema operativo los carga en el **kernel**. Después inicializa sus tablas, crea los procesos de segundo plano que se requieran, y arranca un programa de inicio de sesión o GUI.

Arranque secundario

Cargador de arranque secundario.

- El cargador de arranque (bootloader), en nuestro caso GRUB, es el encargado de la segunda fase del proceso de arranque.
- GNU GRUB (GNU GRand Unified Bootloader) es un gestor de arranque múltiple, desarrollado por el proyecto GNU que nos permite elegir qué sistema operativo arrancar de los instalados.
- Grub se puede instalar en el MBR o en los primeros sectores de una partición primaria activa (es decir, arrancable), identificada como /boot
- Si tengo varios sistemas operativos y quiero que Grub sea capaz de arrancar todos, debo instalarlo en el MBR.
- Este /boot puede estar en la misma partición que la / o en otra partición (igual que /home por ejemplo, que lo separamos de /)

Particiones para el arranque

- **Partición primaria:** partición que si está activa, puede ser usada para arrancar un sistema operativo.
- **Partición extendida:** partición física que no se puede usar a menos que sea dividida en particiones lógicas.
- **Particiones lógicas:** subdivisiones de la extendida.

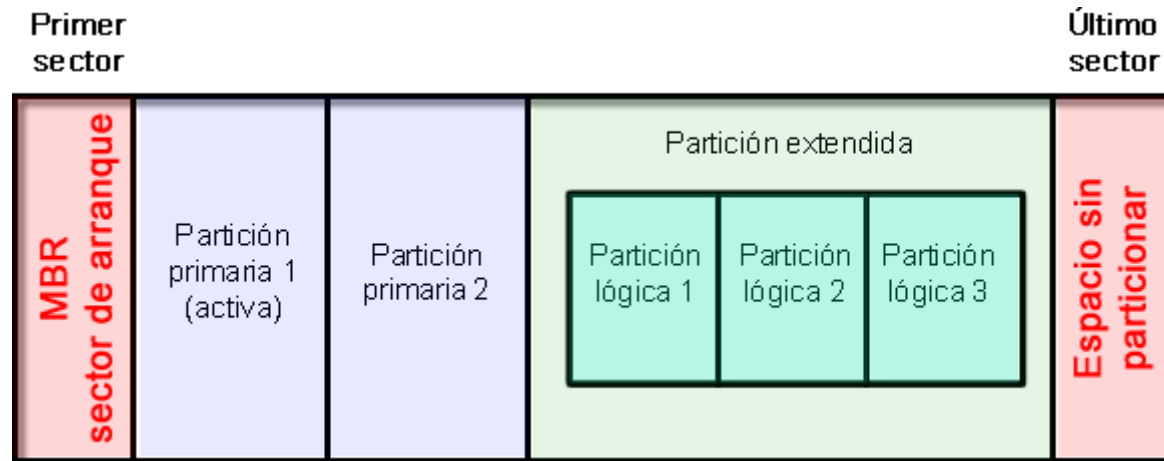


Imagen tomada de: <http://elsitioandres.blogspot.com/2011/09/particion-y-multiple-instalacion-de.html>

Particiones para el arranque

- Se pueden tener hasta **cuatro particiones primarias** o **tres primarias más una extendida**.
- La extendida se puede dividir en hasta **128 particiones lógicas**.
- Podemos indicar que el sistema se inicia desde una partición primaria activa del disco duro primario.
- Solo puede haber una partición primaria marcada como activa.
- **Partición primaria:** partición que si está activa, puede ser usada para arrancar un sistema operativo.
- La partición **/** puede estar en una primaria (activa o no) o incluso en una lógica.
- Pero **/boot** debe estar en una partición que sea arrancable (primaria activa).

Arranque del SO i

- Si se instala primero Windows y luego Linux, éste sobrescribe el gestor de arranque de Windows(BCD o Boot Configuration Data en Windows)
- No hay problema porque Grub reconocerá que están instalado Windows y meterá una entrada en el menú de arranque para poder arrancarlo
- En caso contrario no. Si instalo primero Linux y luego Windows, el gestor de arranque sobrescribe el Grub y luego no puede inicial Linux.
- Se carga el sistema de archivos raíz en disco duro.
- El programa **init** es ejecutado en primer lugar con el **PID=1** preparando el proceso de arranque principal para el comienzo de muchos procesos.
- Muchos de estos procesos son **demonios**.
 - **PID Proccess ID:** es el número entero usado por el kernel de un SO para identificar un proceso de forma unívoca.

Arranque del SO ii

- Un demonio (o daemon, **Disk And Execution **MONitor**) que generalmente se carga en memoria, y está esperando alguna señal (proveniente del núcleo o si es un servicio de red, un cliente que se conecte) para despertarse y ejecutar las funciones necesarias para tratarla.**
- Los daemons que tengamos arrancados **no ocupan la CPU mientras** no es estrictamente necesario.
- Es un proceso que se ejecuta en background (es decir, en segundo plano).
- Todos los demonios tienen un **shell script** situado en el directorio **/etc/init.d/** que permite iniciarlo, pararlo o ver su estado de ejecución.
- Iniciar/parar un demonio:
 - `sudo /etc/init.d/nombre_demonio {star, stop, restart}`
 - `Sudo service nombre_demonio {star, stop, restart}`

Arranque del SO iii

- Desde Debian jessie (Debian 8) y las últimas versiones de Ubuntu, se cambió el arranque de **init** (al estilo Unix System V) por otra forma en la que quien gestiona todo, es un proceso llamado **systemd**.
- Este proceso **systemd** hace lo mismo que init
 - Álvaro: prueba el siguiente comando en la terminal de Ubuntu: **ps -- pid=1**
 - Explica a toda la clase para qué sirve este comando.
 - Cómo se te ve mucha disposición, prueba este comando: **arch**, explica qué muestra.
- **Systemd** es un conjunto de herramientas, librerías y servicios diseñados para llevar la administración y configuración central de los sistemas Linux e interactuar directamente con el núcleo de los mismos.
 - https://en.wikipedia.org/wiki/Category:Linux_distributions_without_systemd
 - Silvia: consulta el enlace anterior y comenta de qué trata.

Arranque del SO Debian/Ubuntu

- **Systemd** se encarga de arrancar todos los demás procesos durante el inicio del sistema.
- Cuáles se arrancan depende del nivel de ejecución en el que nos encontremos (también llamado **runlevel**)
- Por defecto, hay 7 runlevels posibles, desde el runlevel 0 hasta el runlevel 5, aunque esto es posible modificarlo.
 - **Init 0:**
 - **Init 1:**
 - **Init 2-4:**
 - **Init 5:**
 - **Init 6:**

Arranque del SO Debian/Ubuntu

- **Init 0:** ejecuta los procesos necesarios para parar el sistema
- **Init 1:** modo monousuario. Se ejecuta para realizar tareas de mantenimiento como root con unos servicios (o demonios) mínimos
- **Init 2-4:** modo multiusuario. Por defecto están todos configurados iguales (es decir, arrancan los mismos demonios), y están preparados por si queremos modificarlos (añadiendo o quitando demonios).
- **Init 5:** Multiusuario con entorno gráfico (sistema X Window)
- **Init 6:** Parecido al 0 pero sirve para reiniciar
- Los comandos **halt**, **reboot**, **shutdown** o **poweroff** lo único que hacen es llamar al nivel de ejecución 0 o 6

Arranque del SO Debian/Ubuntu

La forma como se organizan estos daemons en cada nivel de ejecución es muy simple:

- Cada nivel de ejecución tiene un directorio situado en `/etc/rcX.d/` donde la “X” es el número de runlevel.
- En estos directorios encontramos enlaces simbólicos a los shell scripts de los **daemons** situados en `/etc/init.d/`, que nos sirven para indicar al sistema si queremos iniciar o parar el daemon al que apuntan.
- Con el mismo nombre del enlace se identifica la acción a realizar: si el enlace empieza por “**S**” (**Start**) indicamos que queremos iniciar el daemon, mientras que se empieza por “**K**” (**Kill**) indica que queremos pararlo.
- Si el nombre no empieza por ninguna de estas letras, el sistema no hace nada con él.

Arranque del SO Debian/Ubuntu

- Después de esta letra se pone un número de 2 cifras entre “00” y “99”, que indica el orden de inicio o parada de los mismos.
- Este orden es importante, ya que algunos daemons necesitan que otros estén en ejecución antes de ser iniciados.
- Al cambiar de nivel de ejecución, el sistema inspeccionará los daemons del directorio correspondiente y empezará, primero, parando los daemons indicados y después iniciará los demás.
- Consulta por ejemplo:
 - `ls /etc/rc0.d`
 - `ls /etc/rc5.d`

Bibliografía

- **CARRETERO**, Jesús, **GARCÍA**, Félix, **DE MIGUEL**, Pedro, **PÉREZ**, Fernando. Sistemas Operativos: una visión aplicada. McGraw-Hill, 2001.
- **STALLINGS**, William. Sistemas operativos: aspectos internos y principios de diseño. 5ª Edición. Editorial Pearson Educación. 2005. ISBN: 978-84-205-4462-5.
- **TANENBAUM**, Andrew S. Sistemas operativos modernos. 3ª Edición. Editorial Prentice Hall. 2009. ISBN: 978-607- 442-046-3.

