

INTRODUCCIÓN

Todos los que en algún momento hemos usado alguna **distribución linux** o celular con **android**, hemos interactuado sin saberlo con versiones del **kernel linux**. Aunque en la actualidad las características son más complejas y requieren en muchos casos una revisión más extensa para poder entender los aspectos técnicos.

Si tu interes es empezar con el desarrollo del **kernel linux** muchas veces tendras que regresarte un poco para poder entender el por que se han ido agregaron ciertas características, si ese es tu interes creo que este compendio te ayudara muchísimo.

Como ya sabras **Linux** es un sistema operativo derivado de **UNIX** que ha ganado una gran popularidad en los ultimos años, con sus ya *25 años* de vida conserva una arquitectura y estilo similares a los de cualquier otro **UNIX**; de hecho la compatibilidad ha sido una de los principales objetivos del diseño del proyecto. Aunque su diseño es mucho más actual que algunos **UNIX** ha sido gracias a que la comunidad de desarrolladores que ha venido agregando nuevas características y solucionado un gran numero de Errores al pasar del tiempo.

EN EL PRINCIPIO DE LOS TIEMPOS

UNIX es el perfecto ejemplo del trabajo que se puede efectuar cuando se encauzan todas las energias a la busqueda de un ideal tecnologico. Cuando *AT&T* distribuye casi libremente en 1974 el código fuente del sistema operativo a la Universidades por que, entre otras razones, no ve ningun futuro economico a su producto. No parece dudar del entusiasmo de los estudiantes, profesores e investigadores de la Computacion. Esta primer comunidad pasara mucho tiempo modificando y mejorando el producto, subiendo todas la novedades a *AT&T* para que se integre al producto oficial. Tras el cambio de licencia en 1978, la energia de la comunidad se encauzo hacia el proyecto Universitario **BSD**, dejando al **UNIX** comercial de *AT&T*.

Los primeros ordenadores eran escencialmente herramientas de investigacion en manos de universitarios y necesidades militares. En los laboratorios de investigacion, los programas circulaban como las ideas: libremente. Era absolutamente normal que un programa desarrollado por un equipo de programadores o investigadores se distribuyera a otros equipos de otras universidades y a cualquier otro lugar donde hiciera falta.

Y nada habia de raro en que este programa fuera modificado por otro equipo, y asi sucesivamente. A dia de hoy, cuando un ilustre matematico demuestra un teorema dificil, publica el resultado de sus investigaciones en obras especializadas con el fin de ayudar al progreso de la ciencia. Todo el mundo tiene acceso a ello.

Pero el universo de la computación ha seguido otros caminos. Pese a ser una ciencia, el futuro de las investigaciones no se circunscribe al mundo Universitario.

Rapidamente las empresas vieron el inmenso interes de automatizar varias de sus tareas, como la contabilidad, los pagos, etc. Con la compra de los primeros grandes ordenadores de gestion, se necesitaron programas. Estos programas tuvieron que ser protegidos como secretos industriales, habia

nacido una nueva industria: el desarrollo de programas. Con su entrada en la dinamica de las grandes empresas, la Computacion perdio rapidamente la inocencia y se hizo mucho menos libre. Se empezo a hablar de licencias, impuestos, tasas, prohibición de copia.

EL PROYECTO GNU Y LA FSF

Es importante resaltar que **Linux** no seria lo que es sin el proyecto llamado **GNU**.

A principios de 1984, **Richard Stallman**, en aquella época empleado en el *AI Lab* del **MIT**, abandonó su trabajo para comenzar el proyecto **GNU**. **Stallman** se consideraba un desarrollador de los que gozaban compartiendo sus inquietudes tecnológicas y su código. Veía con desagrado cómo su negativa a firmar acuerdos de exclusividad y no compartición le estaban convirtiendo en un extraño en su propio mundo, y cómo el uso de ~~software propietario~~ en su entorno le dejaba impotente ante situaciones que antes podía solventar fácilmente.

Su idea al abandonar el **MIT** era construir un sistema de software completo, de propósito general, pero completamente libre. El sistema (y el proyecto que se encargaría de hacerlo realidad) se llamó **GNU** (*acrónimo recursivo*, **GNU's Not Unix**). Aunque desde el principio el proyecto **GNU** incluyó en su sistema software ya disponible (como *Tex*, o más adelante, el sistema *X Window*), había mucho que construir. **Richard Stallman** comenzó por escribir un compilador de **C** (**GCC**) y un editor (**Emacs**), ambos aún en uso (y muy populares) hoy día.

Desde el principio del proyecto **GNU**, **Richard Stallman** estaba preocupado por las libertades que tendrían los usuarios de su software.

Estaba interesado en que no sólo los que recibieran los programas directamente del proyecto **GNU**, sino cualquiera que lo recibiera después de cualquier número de redistribuciones y (quizás) modificaciones, siguiera disfrutando de los mismos derechos (modificación, redistribución, etc.). Para ello, escribió la licencia **GPL**, probablemente la primera licencia de software diseñada específicamente para garantizar que un programa fuera libre en este sentido. Al mecanismo genérico que utilizan las licencias tipo **GPL** para conseguir estas garantías, **Richard Stallman** lo llamó **copyleft**, que hoy día es el nombre de una gran familia de licencias de software libre.

Richard Stallman también fundó la **Free Software Foundation (FSF)** con el fin de conseguir fondos para el desarrollo y la protección del **software libre**, y sentó los fundamentos éticos del **software libre**, con documentos como **The GNU Manifesto** y **Why Software Should Not Have Owners**.

Desde el punto de vista técnico, el proyecto **GNU** fue concebido como un trabajo muy estructurado y con metas muy claras. El método habitual estaba basado en grupos relativamente pequeños de personas (habitualmente voluntarios) que desarrollaban alguna de las herramientas que luego encajarían perfectamente en el rompecabezas completo (el sistema **GNU**). La modularidad de **UNIX**, en la que se inspiraba el desarrollo, encajaba perfectamente en esta idea.

El método de trabajo generalmente implicaba el uso de Internet, pero ante la escasa implantación de aquellos días, la **Free Software Foundation** también vendía cintas en las que grababa las aplicaciones, siendo probablemente uno de las primeras organizaciones en beneficiarse económicamente (aunque de manera bastante limitada) de la creación de **software libre**.

A principios de la década de **1990**, unos seis años después de su nacimiento, el proyecto **GNU** estaba muy cerca de tener un sistema completo similar a **Unix**. Aun así, hasta ese momento todavía no había producido una de las piezas fundamentales: el **kernel** del sistema (el núcleo del sistema operativo que se relaciona con el hardware y permite que todo funcione). Sin embargo, el software de **GNU** era muy popular entre los usuarios de las distintas variantes de **Unix**, por aquella época el sistema operativo más

usado en las empresas. Además, el proyecto **GNU** había conseguido ser relativamente conocido entre los profesionales informáticos, y muy especialmente entre los que trabajaban en universidades. En esa época, sus productos ya gozaban de una merecida reputación de estabilidad y calidad.

A continuación te mostrare una tabla con los proyectos que **GNU** considera son proyectos **software libre**.

			PROYECTOS GNU
<i>3dldf</i>	1:2	2:2	3:2
<i>anubis</i>	1:3	2:3	3:3
<i>autoconf</i>	1:4	2:4	3:4
<i>bash</i>	1:5	2:5	3:5
<i>bool</i>	1:6	2:6	3:6
<i>ccscript</i>	1:7	2:7	3:7
<i>clisp</i>	1:8	2:8	3:8
<i>coreutils</i>	1:9	2:9	3:9
<i>dc</i>	1:10	2:10	3:10
<i>diction</i>	1:11	2:11	3:11
<i>easejs</i>	1:12	2:12	3:12
<i>enscript</i>	1:13	2:13	3:13
<i>foliot</i>	1:14	2:14	3:14
<i>fribidi</i>	1:15	2:15	3:15
<i>gcide</i>	1:16	2:16	3:16
<i>gleem</i>	1:17	2:17	3:17
<i>gnash</i>	1:17	2:17	3:17
<i>gettext</i>	1:17	2:17	3:17

<i>gnu-c-manual</i>	1:17	2:17	3:17
<i>gnubiff</i>	1:17	2:17	3:17
<i>gnue</i>	1:17	2:17	3:17
<i>gnulib</i>	1:17	2:17	3:17
<i>gnupg</i>	1:17	2:17	3:17
<i>gnusound</i>	1:17	2:17	3:17
<i>gnuzilla</i>	1:17	2:17	3:17
<i>greg</i>	1:17	2:17	3:17
<i>gsl</i>	1:17	2:17	3:17
<i>guile</i>	1:17	2:17	3:17
<i>guix</i>	1:17	2:17	3:17
<i>health</i>	1:17	2:17	3:17
<i>hyperbole</i>	1:17	2:17	3:17
<i>intlfonts</i>	1:17	2:17	3:17
<i>kopi</i>	1:17	2:17	3:17
<i>libextractor</i>	1:17	2:17	3:17
<i>libmicrohttpd</i>	1:17	2:17	3:17
<i>libunistring</i>	1:17	2:17	3:17
<i>lispintro</i>	1:17	2:17	3:17
<i>make</i>	1:17	2:17	3:17
<i>mediagoblin</i>	1:17	2:17	3:17
<i>mit-scheme</i>	1:17	2:17	3:17

<i>nana</i>	1:17	2:17	3:17
<i>octave</i>	1:17	2:17	3:17
<i>pexec</i>	1:17	2:17	3:17
<i>powerguru</i>	1:17	2:17	3:17
<i>qexo</i>	1:17	2:17	3:17
<i>reftex</i>	1:17	2:17	3:17
<i>screen</i>	1:17	2:17	3:17
<i>shtool</i>	1:17	2:17	3:17
<i>speex</i>	1:17	2:17	3:17
<i>superopt</i>	1:17	2:17	3:17
<i>termutils</i>	1:17	2:17	3:17
<i>tramp</i>	1:17	2:17	3:17
<i>uucp</i>	1:17	2:17	3:17
<i>websocket4j</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>8sync</i>	1:17	2:17	3:17
<i>autoconf-archive</i>	1:17	2:17	3:17
<i>apl</i>	1:17	2:17	3:17
<i>bayonne</i>	1:17	2:17	3:17
<i>bpel2owfn</i>	1:17	2:17	3:17
<i>cflow</i>	1:17	2:17	3:17
<i>cobol</i>	1:17	2:17	3:17

<i>cpio</i>	1:17	2:17	3:17
<i>ddd</i>	1:17	2:17	3:17
<i>diffutils</i>	1:17	2:17	3:17
<i>ed</i>	1:17	2:17	3:17
<i>eprints</i>	1:17	2:17	3:17
<i>fontopia</i>	1:17	2:17	3:17
<i>g-golf</i>	1:17	2:17	3:17
<i>gcl</i>	1:17	2:17	3:17
<i>gforth</i>	1:17	2:17	3:17
<i>glib</i>	1:17	2:17	3:17
<i>gnat</i>	1:17	2:17	3:17
<i>gnu-crypto</i>	1:17	2:17	3:17
<i>gnubik</i>	1:17	2:17	3:17
<i>gnufm</i>	1:17	2:17	3:17
<i>gnumach</i>	1:17	2:17	3:17
<i>gnupod</i>	1:17	2:17	3:17
<i>gnuspeech</i>	1:17	2:17	3:17
<i>goptical</i>	1:17	2:17	3:17
<i>grep</i>	1:17	2:17	3:17
<i>gslip</i>	1:17	2:17	3:17
<i>guile-dbi</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17

<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17
<i>xhippo</i>	1:17	2:17	3:17

A finales de la década de los ochenta, principios de los noventa, el proyecto **GNU** contaba con una gama básica de utilidades y herramientas que permitían tener el sistema operativo al completo. Ya entonces muchas aplicaciones libres eran las mejores en su campo (utilidades **UNIX**, compiladores), siendo especialmente interesante el caso de **X Window**. Sin embargo, para terminar el rompecabezas faltaba una pieza esencial: el núcleo del sistema operativo. El proyecto **GNU** estaba buscando esa pieza con un proyecto llamado **Hurd**, que pretendía construir un **kernel** con tecnologías modernas.