1. Introducción al Software Libre

1.1. Los inicios. ¿Qué es GNU?

Para entender todo el movimiento del software libre, debemos situarnos a finales de la década de los sesenta, principios de los setenta. En aquellos tiempos las grandes compañías de ordenadores no daban el valor que hoy día se da al software. En su gran mayoría eran fabricantes de ordenadores que obtenían sus principales ingresos vendiendo sus grandes máquinas, a las que incorporaban algún tipo de sistema operativo y aplicaciones. Las universidades tenían permiso para coger y estudiar el código fuente del sistema operativo para fines docentes. Los mismos usuarios podían pedir el código fuente de *drivers* y programas para adaptarlos a sus necesidades. Se consideraba que el software no tenía valor por sí mismo si no estaba acompañado por el hardware que lo soportaba.

En este entorno, los laboratorios Bell (AT&T) diseñaron un sistema operativo llamado UNIX, caracterizado por la buena gestión de los recursos del sistema, su estabilidad y su compatibilidad con el hardware de diferentes fabricantes (para homogeneizar todos sus sistemas). Este último hecho fue importantísimo (hasta entonces todos los fabricantes tenían sus propios operativos incompatibles con los otros), ya que devino el factor que le proporcionó mucha popularidad.

Poco a poco, las grandes empresas empezaron a tomar conciencia del valor del software: primero fue IBM la que en 1965 dejó de dar el código fuente de su sistema operativo, a finales de los setenta Digital Research empezó a vender el suyo, etc. Este hecho hizo que todas las compañías se dieran cuenta de que el software podía ser muy rentable y les podía aportar grandes beneficios. A partir de este hecho, la mayoría de empresas empezaron a poner reticencias a dejar el código fuente de sus programas y sistemas operativos y empezaron a vender sus programas como un valor añadido a su hardware. En este entorno cada vez más cerrado, **Richard Stallman** (que trabajaba en el MIT, Massachusetts Institute of Technology) se sintió indignado al comprobar que cada vez era más difícil conseguir el códigofuente de los programas que utilizaba para adaptarlos a sus necesidades, tal como había hecho hasta entonces.

El mismo Stallman cuenta como anécdota lo mucho que se enfadó al descubrir que la compañía que les había vendido una nueva impresora para el laboratorio donde trabajaba no le quería facilitar el código fuente de los drivers. ¡Él sólo quería modificarlos para que le avisara automáticamente cuando se atascaba el papel! La compañía se negó a proporcionárselos.

A partir de ese momento, Stallman decidió ser consecuente con sus ideales e iniciar un gran proyecto para intentar abrir otra vez el código fuente de los programas. Consciente de que no podría conseguir que las compañías cedieran en este punto, se propuso crear su propio sistema operativo y aplicaciones iniciando un proyecto llamado **GNU**.

De especial interés para entender los motivos que llevaron a Stallman a iniciar **GNU** es su primer manifiesto, el documento donde explicó a toda la comunidad en qué consistiría el proyecto, cómo lo orientaría y por qué tenía que hacerlo. En él empezó a describir el concepto de **software libre** y para qué creía necesario que programadores y desarrolladores de alrededor del mundo contribuyeran con él. Aunque en muchas ocasiones se confunde el concepto de software libre con el de software gratuito (en inglés, free tiene los dos significados), en posteriores documentos se ha dejado muy claro que el software libre no debe por qué ser gratuito. Debemos entender como software libre programas de los cuales podemos conseguir su código fuente, estudiarlo, modificarlo

y redistribuirlo sin que nos obliguen a pagar por ello. Lo que debemos tener claro es que sí que podemos pedir el dinero que queramos por los programas y su código fuente, el soporte que podemos ofrecer a los usuarios, los libros que vendamos o el material que proporcionemos, tal y como muchas compañías que distribuyen **GNU/Linux** hacen. Sin embargo, en ningún momento, podemos obligar a que los usuarios no distribuyan el software que les hemos vendido. Éste debe poder ser distribuido de forma libre. Es una forma diferente de entender el software a la que estamos acostumbrados. En muchos de los textos de la **FSF (Free Software Foundation)** se habla más de filosofía que de ingeniería. Debemos entender todo este movimiento más como una forma de pensar o hacer las cosas que como una compañía más de software.

La filosofía que en la FSF se tiene del software lo define con las siguientes **cuatro libertades**:

- La libertad 0 se refiere a la libertad de poder usar el programa para cualquier propósito.
- La libertad 1 es la que permite estudiar cómo funciona el programa y adaptarlo a las propias necesidades. El acceso al código fuente es una condición necesaria para garantizar esta libertad.
- La libertad 2 es la que permite distribuir libremente copias del software, ayudando al vecino.
- La última libertad, o libertad 3, es la que permite mejorar el programa y hacer públicas las propias mejoras, en beneficio de toda la comunidad. El acceso al código fuente, asimismo, es un requisito imprescindible para asegurar esta libertad.

Para dar todas estas libertades al software que se desarrollaba en el proyecto y a los usuarios finales del mismo se escribió la licencia, con la cual se ha protegido todo este tipo de programas, la **GPL (General Public License)**. Esta licencia pone por escrito las ideas anteriormente comentadas.

El proyecto empezó a producir software a partir de 1984, comenzando con el desarrollo de todas la herramientas necesarias para poder implementar un sistema operativo completo. Aunque realizar un proyecto de estas características es un proceso largo y complejo, desde el principio muchos programadores y desarrolladores de software se vieron cautivados por la idea de Stallman y empezaron a colaborar con él de forma gratuita. La comunidad no paró de crecer, y poco a poco empezaron a disponer de las herramientas necesarias (editores, compiladores, etc.) para implementar el núcleo del sistema operativo, que era la tarea que requería las herramientas que se estaban desarrollando. Desde el primer momento se quiso crear un sistema operativo parecido a UNIX y siguiendo las normas POSIX (*Portable Operating System Interface*). Si bien UNIX también tenía sus problemas y carencias, era, y sigue siendo, suficientemente bueno como para adaptarse a la mayoría de las necesidades. La tarea de diseñar y escribir el núcleo del sistema operativo fue la que se dejó para el final del proceso. Aún actualmente está por finalizar definitivamente y el núcleo del GNU, llamado **Hurd**, permanece en fase de desarrollo.

1.2 ¿Qué es GNU/Linux?

En este contexto, y cuando la FSF todavía no tenía ningún núcleo estable para su sistema operativo, un profesor de la Universidad de Holanda, **Andrew Tanenbaum**, decidió escribir un sistema operativo para que sus estudiantes pudieran estudiarlo. Igual que Stallman, hasta el momento había podido utilizar el código fuente del UNIX de AT&T para que sus alumnos aprendieran a diseñar sistemas operativos. Su idea era escribir un sistema operativo que pudiera ser estudiado y modificado por cualquiera que quisiera. En 1987 se puso manos a la obra y llamó a su proyecto mini UNIX, dando lugar a **MINIX**. Al no utilizar ni una sola línea de código del UNIX de AT&T, no hay ninguna restricción en coger el código, utilizarlo y modificarlo libremente.

Tanenbaum quiso crear un sistema orientado a fines docentes, por lo que lo diseñó utilizando una arquitectura *micro-kernel*, ideal para una fácil comprensión y aportando una tecnología muy novedosa para la época que le permitía versatilidad, multi-plataforma, etc. Éste ha sido uno de los puntos fuertes y débiles a la vez del MINIX: aunque el sistema es una pequeña joya para su estudio y diseño, es muy probable que nunca se pueda utilizar en entornos reales. Se optó por hacerlo entendible, modular y muy pedagógico, pero no rápido. De todas formas, Tanenbaum tampoco pretendía eso; a lo largo de los años MINIX ha ido evolucionando y realmente hoy en día todavía sigue existiendo y siendo estudiado por muchos alumnos de universidades de todo el mundo.

Aquí es cuando entra en juego **Linux**. Mientras la FSF seguía con su gran proyecto proporcionando herramientas para la construcción de un sistema operativo, Tanenbaum orientaba MINIX para fines docentes y muchas empresas seguían haciendo evolucionar sus propias versiones de UNIX. **Linus Torvalds**, estudiante de la Universidad de Helsinki, decide crear en agosto de 1991 su propio núcleo para un nuevo sistema operativo, Linux. Su idea era crear un UNIX para PC para que todos los que quisieran lo pudieran utilizar en su ordenador. La primera aparición en escena que hizo fue en un debate sobre MINIX y sistemas operativos, donde expuso las siguientes ideas:

```
Newsgroups: comp.os.minix

Asunto: What would you like to see most in minix? Fecha: 25 Aug. 91
20:57:08 GMT Organization: University of Helsinki Hello everybody out
there using minix. I'm doing a (free) operating system (just a ho-bby,
won't be big and professional like gnu) for 386(486) AT clones. This
has been brewing since april, and is starting to get ready. I'd like
any feedback on things people like/dislike in minix, as my OS resembles
it somewhat (same physical la-yout of the file-system (due to practical
rea-sons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to
work.
This implies that I'll get something practical within a few months,
and I'd like to know what features most people would want. Any
suggestions are welcome, but I won't promise I'll implement them :-)
```

Si accediéramos al fórum de debate donde apareció este primer mensaje, veríamos cómo rápidamente gente de todo el mundo empezó a interesarse por este nuevo sistema, que al utilizar el compilador e intérprete de comandos de GNU (gcc y bash) como piezas fundamentales, también tenía las características de software libre. Aunque en palabras del mismo Torvalds, si él hubiera sabido la cantidad de trabajo necesario para lograr que su idea funcionase, nunca lo hubiera hecho: esfuerzos de muchos expertos en informática de todo el mundo hicieron posible este proyecto.

De hecho, en los primeros años de su existencia, GNU/Linux se identificaba como el sistema operativo de los *hackers*. Su difícil instalación, manipulación y falta de *drivers* lo hacían una herramienta apta únicamente para gente muy entendida en el tema. Fueron estos primeros usuarios los que diseñaron los *drivers* para los discos, impresoras, tarjetas, etc. y los que empezaron a dar a conocer al mundo este sistema. Poco a poco, el número de usuarios empezó a crecer y actualmente ya existen muchas empresas y grupos de usuarios que crean sus propias distribuciones de GNU/Linux.