

Utilización y Administración avanzada de sistemas GNU/Linux y aplicaciones Software Libre para estudiantes universitarios

Software libre: Historia y ventajas

Manuel Estrada Sáinz

Ramón Rey Vicente

Utilización y Administración avanzada de sistemas GNU/Linux y aplicaciones Software Libre para estudiantes universitarios

Software libre: Historia y ventajas

por Manuel Estrada Sáinz y Ramón Rey Vicente

Copyright (c) 2.007 Ramón Rey Vicente <rrey@ramonrey.net>, Manuel Estrada Sáinz <>.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Historial de revisiones

Revisión 1.1 18-04-2007 Revisado por: Ramón Rey VicenteManuel Estrada Sáinz

Tabla de contenidos

1. Software libre	1
1.1. ¿Qué es?	1
1.1.1. Software propietario	1
1.1.2. Definición de Software Libre	1
1.1.3. ¿Porque tanta oposición?	3
1.1.4. Control y seguridad	4
1.2. ¿Cómo funciona?	5
1.2.1. Historia de un proyecto	5
1.2.2. Modelos de desarrollo	5
1.3. Ventajas	6
1.4. Mitos	7
2. GNU/Linux	9
2.1. Un poco de historia	9
2.1.1. UNIX prospera.	9
2.1.2. Los tiempos oscuros.	9
2.1.3. Richard Matthew Stallman.	9
2.1.4. El proyecto GNU.	10
2.2. Filosofía UNIX.	10
2.2.1. Todo es un archivo.	10
2.2.2. La navaja suiza.	10
2.2.3. Manual en línea.	11
2.3. ¿Por qué llamarlo "GNU/Linux" en lugar de "linux"?	11
2.4. ¿Qué son las "distribuciones" de GNU/Linux?	11
2.5. Documentación	12
Bibliografía.....	13
A. GNU Free Documentation License.....	14
A.1. PREAMBLE	14
A.2. APPLICABILITY AND DEFINITIONS	14
A.3. VERBATIM COPYING.....	15
A.4. COPYING IN QUANTITY	16
A.5. MODIFICATIONS.....	16
A.6. COMBINING DOCUMENTS.....	18
A.7. COLLECTIONS OF DOCUMENTS	18
A.8. AGGREGATION WITH INDEPENDENT WORKS.....	19
A.9. TRANSLATION	19
A.10. TERMINATION.....	19
A.11. FUTURE REVISIONS OF THIS LICENSE.....	20
A.12. ADDENDUM: How to use this License for your documents.....	20

Capítulo 1. Software libre

1.1. ¿Qué es?

1.1.1. Software propietario

Antes de entrar a definir el software libre, veamos algunas características del software propietario mediante un ejemplo.

Imaginad que vais a comprar un coche y las condiciones de compra son:

1. Ud. sólo puede circular con su coche por la provincia en la que reside. Si quisiera circular por otra provincia diferente necesitaría pagar más dinero en concepto de licencia.
2. No podrá ceder ni alquilar su coche.
3. No podrá modificarlo de ninguna manera, no podrá ponerle otro radio-cassette, colgarle unos dados del retrovisor, cambiarle los neumáticos, ... Para hacerlo tendrá que solicitarlo al vendedor y obviamente le cobrarán por ello, y al sólo poder hacer las modificaciones el vendedor ¿se imagina cómo serán las tarifas?
4. No podrá abrirlo/desmontarlo para estudiar su funcionamiento.

¿Compraría un coche en estas condiciones? Seguro que no. Entonces, ¿cuál es la razón de comprar software propietario bajo unas condiciones similares?

Cuando compra un software propietario, si se molesta en leer la licencia que lo acompaña, verá que:

1. Sólo podrá instalar el software en un determinado número de equipos, requiriendo el pago adicional, en concepto de licencias, si quisiera instalarlo en más equipos.
2. Ud. no puede ceder ni alquilar el software que acaba de comprar.
3. No puede modificarlo de ninguna manera. El único que puede hacerlo es el desarrollador y en las condiciones que considere oportunas (y siempre y cuando le salga rentable).
4. No podrá realizar ingeniería inversa para estudiar su comportamiento.

1.1.2. Definición de Software Libre

El "Software Libre" es un asunto de libertad, no de precio. Para entender el concepto, debes pensar en "libre" como en "libertad de expresión", no como en "barra libre" [N. del T.: en inglés una misma palabra (free) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión].

"Software Libre" se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, deberías tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquier persona en cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.

También deberías tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas modificaciones existen. Si publicas tus cambios, no tienes por qué avisar a nadie, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador ni a ninguna otra entidad específica.

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar). Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero debes tener la libertad de distribuir estos formatos si encontraras o desarrollaras la manera de crearlos.

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, debes tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el software libre.

Para que estas libertades sean reales, deben ser irrevocables mientras no hagas nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no le hayas dado motivos, el software no es libre.

Así pues, quizás hayas pagado para obtener copias de software GNU, o tal vez las hayas obtenido sin ningún coste. Pero independientemente de cómo hayas conseguido tus copias, siempre tienes la libertad de copiar y modificar el software, e incluso de vender copias.

"Software libre" no significa "no comercial". Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante.

A veces las normas de control de exportación del gobierno y las sanciones mercantiles pueden restringir tu libertad de distribuir copias de los programas a nivel internacional. Los desarrolladores de software no tienen el poder de eliminar o sobrepasar estas restricciones, pero lo que pueden y deben hacer es rehusar el imponerlas como condiciones de uso del programa. De esta manera, las restricciones no afectarán a actividades y gente fuera de las jurisdicciones de estos gobiernos.

Por último, fíjate en que los criterios establecidos en esta definición de software libre requieren pensarse cuidadosamente para interpretarlos. Para decidir si una licencia de software concreta es una licencia de software libre, lo juzgamos basándonos en estos criterios para determinar si tanto su espíritu como su letra en particular los cumplen. Si una licencia incluye restricciones contrarias a nuestra ética, la rechazamos, aun cuando no hubiéramos previsto el problema en estos criterios. A veces un requisito de una licencia plantea una situación que necesita de una reflexión minuciosa, e incluso conversaciones con un abogado, antes de que se pueda decidir si la exigencia es aceptable. Cuando llegamos a una conclusión, a veces actualizamos estos criterios para que sea más fácil ver por qué ciertas licencias se pueden calificar o no como de software libre.

Por lo general, para decidir si un determinado software es libre, puedes hacerte las siguientes preguntas:

1. ¿Te dan las fuentes del programa?
2. ¿Puedes modificar esas fuentes?
3. ¿Puedes distribuir lo que modifiques?
4. ¿Puedes vender esas modificaciones al precio que quieras?
5. ¿Debes añadir las fuentes, obligatoriamente, al distribuir?

Según la FSF si la respuesta a las cuatro primeras es afirmativa el programa es software libre, si además la quinta es también positiva, entonces, será libre y con "copyleft".

1.1.3. ¿Porque tanta oposición?

A muchas empresas de software propietario no les interesa el software libre por motivos claros y sencillos:

- Hay muchas empresas que se dedican a vender software de mala calidad. La disponibilidad del código fuente daría a conocer, sin ninguna duda, la falta de profesionalidad de dichas empresas.
- Habría una mayor competencia y las empresas que más éxito tendrían serían aquellas que proporcionen el mejor servicio, y no aquellas que se aprovechan de su posición de privilegio.

- La utilización de técnicas "lock in" (de dependencia) es prácticamente imposible en software libre, lo que quiere decir, que si no te dan un buen servicio, puedes coger el software e irte con otra empresa que trabaje mejor, o incluso mejorarlo y adaptarlo por ti mismo.

Aprovechando una posición predominante en el mercado y mediante el uso de formatos de almacenamiento de ficheros y protocolos de comunicación propietarios (información) se puede impedir que otras entidades puedan dar los mismos servicios y mantener así al usuario esclavo.

1.1.4. Control y seguridad

Cuando adquirimos un software propietario rara vez se nos suministra el código fuente. Sin embargo, la mayoría de la gente no le da importancia a este hecho ya que, como ellos dicen, "¿Y qué me importa a mí que me den el código fuente si no tengo los conocimientos necesarios para leerlo?".

Sin embargo, la única forma de poder fiarnos de la seguridad de un programa informático es tener a nuestra disposición el código fuente, ya que de esta manera podemos ver cómo ha sido programado y si lo ha sido de forma correcta. Además, también podremos comprobar que en dicho programa no hay:

- Puertas traseras: No es necesario que el desarrollador las incluya. Hay veces en las que un intruso introduce una puerta trasera sin el conocimiento del desarrollador; otras veces la competencia puede pagar a un programador descontento para que introduzca esa puerta trasera.
- Funcionalidades no documentadas: Un programa puede realizar ciertas tareas de las que no somos conscientes. Por ejemplo, un programa para cifrado de correo electrónico tendrá acceso a nuestras claves y podría haber sido programado para mandar esas claves a una determinada persona. O bien podría incluir información sobre nosotros y nuestras claves en las firmas digitales que utilicemos mediante la inclusión de canales subliminales.

La única forma de detectar todo esto es mediante la disponibilidad del código fuente, ya que encontrar un bug en un programa de ordenador no es tan sencillo como se piensa, a menos que se disponga del código fuente. La disponibilidad del código fuente nos da más seguridad en el sentido de transparencia: como el código fuente está disponible se puede auditar y comprobar así que está libre de puertas traseras y/o funcionalidades no documentadas, ya que en caso de tenerlas se descubrirían, y su hallazgo sería una auténtica vergüenza para la empresa que lo vende, pudiéndola llevar a la ruina.

1.1.4.1. Estándares abiertos

Para conservar la libertad de la información, es imprescindible el uso de estándares abiertos. Esto es, estándares establecidos por entidades internacionales e independientes de las empresas particulares (ISO, IEEE, ...).

De otra manera dependeríamos de los intereses de la empresa que comercializa el software. Siendo este el único que puede leer nuestra información y no siendo posible que otras entidades escriban software alternativo, estaríamos vendidos.

1.2. ¿Cómo funciona?

El énfasis del sistema esta en la colaboración. Se trata de una comunidad de usuarios/desarrolladores unida por un fin común. En el fondo todos son egoístas, trabajan en el proyecto porque lo usan y les interesa que funcione lo mejor posible y solucione sus propios problemas. De esta manera, todos se benefician.

1.2.1. Historia de un proyecto

1. alguna persona o entidad comienza un proyecto para satisfacer una necesidad propia. Por lo general no tienen que empezar de cero ni hacer todo el trabajo, hay ya otros proyectos que hacen cosas similares y bibliotecas de funciones que hacen parte del trabajo. No tienen mas que coger lo que les interesa de entre todo el software disponible a nivel mundial y mirar como se han resuelto otros problemas similares.
2. Cuando el proyecto se hace publico, otra gente puede utilizarlo, encontrar deficiencias y corregirlas.
3. Cuando se han hecho modificaciones, se tratan de integrar con el proyecto original, y son los autores originales los que deciden si los cambios se aceptan o no.
4. Si se aceptan los cambios, el proyecto mejora y la historia se repite.
5. En el caso de que los cambios sean rechazados, el autor de estos puede o bien mantener los cambios independientemente del proyecto original, o iniciar un nuevo proyecto (bifurcación/fork).

1.2.2. Modelos de desarrollo

1.2.2.1. Catedral

Es el modelo tradicional de desarrollo de software. De este tipo son las técnicas de desarrollo que se estudian en las asignaturas de Ingeniería del Software y se usan en las empresas.

Sus principales características son:

- Paso a paso, avances pequeños
- Siguiendo un diseño de un arquitecto magistral
- Gran secreto

- Grandes recursos
- Solo se deja entrar a los feligreses una vez terminada

1.2.2.2. Bazar

Es el modelo mas habitual en software libre. A menudo es considerado inviable por los expertos en ingeniería del software, pero el hecho es que funciona.

Sus principales características son:

- Gran número de desarrolladores
- Diferente lugar geográfico
- Voluntarios
- Diferente Idioma (Ingles LinguaFranca)
- No hay un diseño escrito sino un problema por resolver

Bajo este modelo se ha producido software de gran calidad, como:

- El núcleo Linux
- Apache
- Samba
- The Gimp

1.3. Ventajas

- Internacionalización: dado el carácter global de los proyectos, siempre hay quien haga las traducciones.
- Reutilización de código e ideas: siendo el código libre, cualquiera puede coger partes de otros proyectos o ver como han resuelto otros los distintos problemas.
- Reutilización de componentes: dado el carácter cooperativo, los proyectos tratan de producir componentes reutilizables como puede ser aspell, un corrector ortográfico de gran calidad. El equipo de desarrollo de aspell hace el motor de corrección ortográfica, y todos los demás proyectos pueden beneficiarse de ello. Y los usuarios solo tienen que preocuparse de instalar diccionarios una vez para todos los programas.
- Rapidez de desarrollo: son decenas, cientos y a veces miles las personas que colaboran en determinadas fases del desarrollo.
- Robustez: las extensivas pruebas de funcionamiento entre los usuarios realimentan a los desarrolladores en ciclos increíblemente cortos.

- Extensibilidad: cualquiera puede desarrollar nuevas funcionalidades. La calidad de su desarrollo y su aceptación por parte de los usuarios valida la incorporación del nuevo código a la distribución oficial.
- Soporte técnico:
 - GNU/Linux cuenta con el mayor soporte técnico del mundo. La comunidad de usuarios, que va desde meros aficionados y estudiantes a curtidísimos profesionales y consultores del mundo UNIX, tiene una predisposición a la colaboración, especialmente a través de los diferentes medios que ofrece Internet, que permite obtener tiempos de respuesta a cuestiones de tipo servicio técnico muy superiores a los servicios convencionales.
 - Soporte técnico a través de canales comerciales en crecimiento explosivo: autónomos, pymes y grandes empresas del entorno GNU/Linux y últimamente compañías como HP e IBM disponen de programas de servicio técnico 24h, 365 días al año.
 - La disposición del código fuente permite a la empresa atacar los hipotéticos problemas con sus propios recursos, bien sea solucionando 'bugs' o bien añadiendo o extendiendo funcionalidades de las aplicaciones. Esto no es posible en entornos comerciales sin una penalización temporal o económica, o aún ambos, normalmente inabordable.

1.4. Mitos

- Dado que cualquiera puede modificarlo y redistribuirlo, al final hay un montón de versiones distintas y es todo un caos.

Hacer muchas variantes del mismo software, no es rentable, supone mucha duplicación de esfuerzos y confusión. Eso es precisamente el tipo de cosa que se trata de evitar desde el software libre. Nadie en su sano juicio bifurca un proyecto sin una buena razón. Y en cualquier caso siempre se puede coger el software directamente del equipo de desarrollo oficial.

- Si no tienes conocimientos suficientes, ni tiempo muchas veces, ¿de qué te sirve el código fuente?

Hay gente que sí tiene el conocimiento y el tiempo, y cuando se descubre algún fallo salta a la luz pública enseguida. De esta forma, todos los usuarios que lo utilizan son conscientes de esos fallos de seguridad y pueden obrar en consecuencia. En materia de seguridad, la falta de transparencia sólo perjudica a los usuarios, porque puede resultar mucho más difícil descubrir las vulnerabilidades del software que estén utilizando.

- Al estar el código fuente disponible los intrusos tienen ventajas, ya que pueden descubrir antes los fallos y aprovecharse de ellos.

Cierto, pero hay algo más sobre este tema, que desde los sectores contrarios al software libre no se comenta. No sólo los intrusos los descubren, sino que también hay gente dedicada a la "caza" de bugs que descubre los fallos, los pone en conocimiento de los usuarios y los arregla, con lo cual los usuarios

pueden obrar en consecuencia (por otro lado, al ser conscientes del fallo de seguridad y gracias a la disponibilidad del código fuente, es posible que cualquier persona con los conocimientos adecuados pueda arreglar el fallo y poner la versión ya reparada a disposición de los usuarios). Con software propietario, habría que esperar a que la empresa sacara el parche o Service Pack correspondiente y no hay que olvidar que esto lo hará cuando le salga económicamente rentable. Generalmente se espera a tener corregidos una determinada cantidad de errores y sólo entonces se libera el correspondiente Service Pack, en lugar de solucionar los fallos cuando se detectan. De este tipo de políticas también salimos perjudicados los usuarios.

Por otro lado, en el mundo propietario, cuando alguien descubre un agujero de seguridad, puede explotarlo durante mucho tiempo sin el conocimiento del afectado o el productor del software. En software libre, se descubren mas agujeros que salen pronto a la luz publica y son tapados rápidamente. En el peor de los casos no es mas que un intercambio de ventajas.

Capítulo 2. GNU/Linux

2.1. Un poco de historia

A finales de la década de los 60, Ken Thompson y Dennis Ritchie, de los Bell Laboratories (AT&T), desarrollaron un sistema operativo que denominaron jocosamente UNIX. Esto fue una broma, porque el proyecto en el que había estado trabajando antes Thompson se llamaba MULTICS.

Era un sistema elegante y, sobre todo, sencillo. Originalmente corría en una máquina (DEC PDP-7) que era menos potente que cualquier teléfono móvil actual. El sistema primeramente se escribió en lenguaje ensamblador, específico para el PDP-7. Si bien esto le brindaba una gran eficiencia al código, tenía un inconveniente muy serio: el sistema no podía transportarse a otro ordenador distinto; si querían correr UNIX en un ordenador nuevo, tendrían que reescribirlo íntegramente. Claramente, este enfoque no era muy práctico. Tuvieron una idea: crear un lenguaje ensamblador 'abstracto', de manera que, reescribiendo UNIX en dicho lenguaje, el esfuerzo de transportarlo a una nueva arquitectura debería de ser mínimo. Así nació el lenguaje de programación C. Y, con él UNIX fue reescrito, convirtiéndose así en el primer sistema operativo transportable de la historia.

2.1.1. UNIX prospera.

Por aquel entonces, AT&T, que poseía los Bell Laboratories, no podía explotar comercialmente UNIX, debido a su calidad de monopolio. La compañía permitió su distribución a empresas y universidades por un precio simbólico. Esto fue un factor crucial en la expansión y desarrollo ulterior de UNIX. En Berkeley, por ejemplo, crearon su propia variante del sistema, que llamaron BSD (de Berkeley Software Distribution). Allí se desarrollaron herramientas tales como el editor 'vi', o la capa de red. Mientras tanto, AT&T seguía trabajando en su versión, llamada *System V*. El lenguaje AWK, por ejemplo, fue contribuido en esta época. A su vez, la empresa *SUN Microsystems* desarrolló una variante llamada SunOS, que luego pasaría a llamarse Solaris. Toda esta plétora de variantes fue posible porque el código fuente de UNIX estaba disponible, y las mejoras realizadas podían compartirse, para el beneficio de los usuarios. Pero toda este sistema de cooperación se vería pronto amenazado.

2.1.2. Los tiempos oscuros.

En el año 1984, debido a la ley antimonopolio existente en EE.UU., AT&T fue obligada a dividirse. A partir de este momento, la restricción que le impedía explotar comercialmente UNIX desapareció, y la primera medida que se dejó notar fue la restricción en la distribución del código fuente del sistema. Esto suponía una amenaza para el esquema de cooperación que se había establecido. Pero había alguien que estaba dispuesto a intentar cambiar esto.

2.1.3. Richard Matthew Stallman.

En el año 1984, Stallman decidió iniciar el proyecto GNU (GNU's Not UNIX), un proyecto cuya finalidad era proporcionar un sistema operativo similar a UNIX, pero con una licencia que impidiese una 'vuelta a la oscuridad' como la que sufrió el propio UNIX. Dicha licencia se llamó GPL (GNU Public License) y le confiere al *software* la propiedad de ser libre y *permanecer* libre.

2.1.4. El proyecto GNU.

Stallman comenzó a construir una herramienta fundamental para el sistema: el *compilador* para el lenguaje C (gcc, de GNU C Compiler). Esta pieza de *software* se ha convertido probablemente en el nexo de unión más importante de todo el software libre. Con el tiempo, fue el compilador utilizado por Linus Torvalds para desarrollar el famoso *kernel* Linux. Un porcentaje muy alto de todo el código asociado al *software* libre, está escrito en C. éste lenguaje es, pues, una *lingua franca*, como podría haber sido el esperanto, en el marco de las lenguas humanas. Y, gracias a Stallman, el compilador gcc (y el resto del sistema) es, literalmente, patrimonio de la humanidad.

2.2. Filosofía UNIX.

¿Por qué tuvo tanto éxito el enfoque de UNIX? Aparentemente, su simplicidad fue un factor decisivo. En su diseño, sus creadores antepusieron la facilidad de comprensión a la eficiencia, de manera que era fácil *entender* el código y, por ende, adaptarlo a las necesidades de otros. UNIX no es una reliquia del pasado; de hecho, la mayor parte de los sistemas operativos actuales son una evolución de UNIX (incluidos MS-DOS y Windows NT/2000/XP). Por eso conviene conocer los principios en los que se fundamenta, puesto que esos mismos principios estarán presentes (de una u otra manera) en los sistemas que hoy podamos manejar.

2.2.1. Todo es un archivo.

Esta idea, propia de la orientación a objetos (si bien la precede), consiste en que la unidad básica para la interacción con el sistema es una entidad llamada archivo que, como los archivos en papel, puede abrirse, leerse, avanzar hojas hacia delante y hacia atrás, escribir en él, y cerrarse. Este modelo tan sencillo puede parecer ingenuo, pero ha probado ser extremadamente valioso. Permite a un programa acceder transparentemente a un documento de texto o a un puerto de comunicaciones.

2.2.2. La navaja suiza.

UNIX incorpora un conjunto de herramientas que guardan cierta analogía con una navaja multiusos. Son simples, pero hacen *muy bien* su trabajo. En lugar de construir programas muy complejos, UNIX

proporcionaba muchas pequeñas herramientas, y un esquema para poder *combinarlas* de forma efectiva. Este diseño escala muy bien, permitiendo al sistema crecer, incorporar nuevas herramientas y, a la vez, ser compatible hacia atrás.

2.2.3. Manual en línea.

Cuando Thompson y Ritchie estaban desarrollando UNIX, solicitaron a sus jefes un ordenador más potente (un DEC PDP-11), a cambio de desarrollar un sistema completo de tipografía (no les dijeron nada acerca de UNIX). Con el nuevo ordenador desarrollaron UNIX sobre C y, Joe F. Ossanna desarrolló troff (de *typesetting run-off*). Este sistema fue incluido en el propio UNIX, de manera que el manual del sistema fue escrito con él, estando disponible en línea desde entonces (a través del programa *man*).

2.3. ¿Por qué llamarlo "GNU/Linux" en lugar de "linux"?

Porque de esa manera se reconoce explícitamente que el sistema operativo no solo es el núcleo linux, sino muchas otras piezas que se escribieron con anterioridad y sin cuya existencia nunca habría sido posible construirlo ni tener algo funcional en nuestros equipos. Todas esas piezas juntas forman un sistema GNU, que es como se denomina al proyecto para construir un sistema operativo totalmente libre iniciado a mediados de los ochenta por la Free Software Foundation. De hecho, el sistema GNU podría tener en un futuro no muy lejano otros núcleos, como el Hurd, con los cuales los usuarios podrán elegir entre sistemas GNU/Linux o GNU/Hurd. Lo realmente importante es disponer de un sistema operativo libre, no el núcleo, el escritorio o el subsistema gráfico que lleve (y que irá cambiando con el tiempo).

2.4. ¿Qué son las "distribuciones" de GNU/Linux?

Una distribución es un modo de facilitar la instalación, la configuración y el mantenimiento de un sistema GNU/Linux. Al principio, las distribuciones se limitaban a recopilar software libre, empaquetarlo en disquetes o CD-ROM y redistribuirlo o venderlo.

Ahora las grandes distribuciones -RedHat, SuSE, Caldera, Mandrake, Corel Linux, TurboLinux...- son potentes empresas que compiten entre sí por incluir el último software, a veces también software propietario, con instalaciones gráficas capaces de autodetectar el hardware y que instalan un sistema entero en unos cuantos minutos sin apenas preguntas.

Entre las distribuciones de GNU/Linux, destaca el proyecto Debian/GNU. Debian nace como una iniciativa no comercial de la FSF, aunque luego se independiza de ésta y va más allá del propio sistema GNU/Linux. Es la única de las grandes distribuciones que no tiene intereses comerciales ni empresariales. Son sus propios usuarios, muy activos, quienes mantienen la distribución de modo comunitario, incluidas todas sus estructuras de decisión y funcionamiento. Su objetivo es recopilar,

difundir y promover el uso del software libre. Reúne el mayor catálogo de software libre, todos ellos probados, mantenidos y documentados por algún desarrollador voluntario.

En una distribución hay todo el software necesario para instalar en un ordenador personal; servidor, correo, ofimática, fax, navegación de red, seguridad, etc.

2.5. Documentación

- The Linux Documentation Project (<http://tldp.org/>)
- TLDP-ES/LuCAS (<http://es.tldp.org/>)
- La Espiral (<http://laespiral.org/documentacion.html>)
- Debian: Documentación (<http://www.debian.org/doc/index.es.html>)

Bibliografía

GNU/LINUX: Filosofía de Desarrollo

(<http://es.tldp.org/Presentaciones/200102linuxcol/filosofia/filosofia/>) .

La definición de Software Libre (<http://www.gnu.org/philosophy/free-sw.es.html>) .

La Licencia de IP Filter (<http://www.es.gnu.org/Licencias/ip.html>) .

GNU/Linux y las NTIC (<http://grupos-locales.hispalinux.es/material/linuxbcn/linux.html>) .

() .

Apéndice A. GNU Free Documentation License

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of

Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and

3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

GNU FDL Modification Conditions

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is

included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or

rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Sample Invariant Sections list

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

Sample Invariant Sections list

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public

License, to permit their use in free software.