SSV. IES Haría UT4. Actividad 2

Características fundamentales de Linux



Este texto se distribuye bajo licencia:

Creative Commons

Reconocimiento-Compartirlgual 3.0

Usted es libre de:

- copiar, distribuir y comunicar públicamente la obra
- hacer obras derivadas

Bajo las condiciones siguientes:



Reconocimiento. Debe reconocer y citar al autor original.

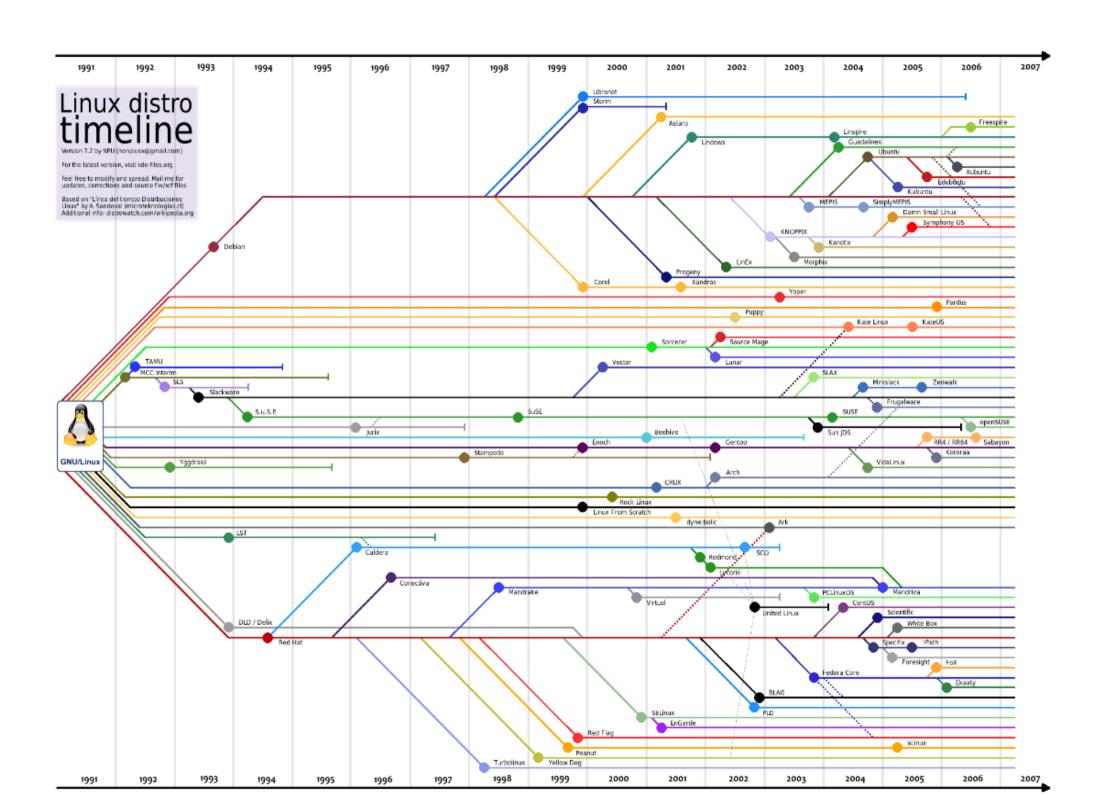


Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

- 1- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- 2- Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

Los derechos derivados de usos legítimos u otras limitaciones no se ven afectados por lo anterior.

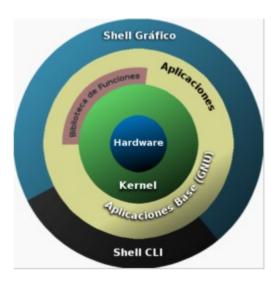
[http://creativecommons.org/licenses/by-sa/3.0/es]



1. Distribución de Linux

Es un conjunto de aplicaciones software empaquetadas y configuradas basadas en el núcleo de Linux para satisfacer las necesidades de un grupo específico de usuarios.

Sigue el siguiente esquema:



Dependiendo del tipo de usuarios al que esté dirigido podemos tener distribuciones de tipo:

- Doméstico
- Empresarial
- Para servidores.

Todas ellas tienen en común que el núcleo o kernel que utilizan es Linux.



Los elementos que suelen componer una distribución son:

- El kernel o núcleo del sistema operativo.
- · Las utilidades básicas.
- Controladores de dispositivos.
- Asistentes para facilitar la instalación, la detección del hardware y el ajuste de los parámetros de los diferentes dispositivos.
- Aplicaciones.
- Sistema de gestión de paquetes.
- Fuentes de texto.
- Documentación.
- Servicios de asistencia.



Coloquialmente se las llama distros

Según el soporte, las distribuciones pueden ser:

- Con soporte comercial, como Fedora (Red Hat), openSUSE (Novell), Ubuntu (Canonical Ltd.), Mandriva y Zentyal.
- Mantenidas por la comunidad de usuarios de la misma como Debian, Arch y Gentoo.
- Hay otras distribuciones que no están relacionadas con alguna empresa o comunidad, como es el caso de Slackware.

Lectura complementaria. Distribuciones Rolling Release y Cyclic Release. http://www.taringa.net/posts/linux/4431203/Rolling-Release-y-Cyclic-Release.html

2. Gestión de paquetes









El software de las distribuciones está dividido en «paquetes».

Cada paquete contiene una aplicación especifica o un servicio.

Ejemplos de paquetes son una librería para manejar el formato de imagen PNG, una colección de tipografías o un navegador web.

El paquete es **generalmente** distribuido en su **versión compilada**, aunque hay distribuciones cuyos paquetes se distribuyen en forma de código fuente y son compilados en el proceso de instalación de los mismos.

La instalación y desinstalación de los paquetes es controlada por un **sistema de gestión de paquetes**.

Cada paquete elaborado para ese sistema de paquetes contiene **meta-información** tal como:

- fecha de creación
- versión

- descripción del paquete
- dependencias e incompatibilidades

El sistema de paquetes analiza esta información para permitir la:

- búsqueda de paquetes
- actualizar las librerías y aplicaciones instaladas
- revisar que todas las dependencias se cumplan y obtenerlas si no se cuenta con ellas de manera automática.

Algunos de los sistemas de paquetes más usados son:

- **RPM**, creado por Red Hat y usado por un gran número de distribuciones de Linux, como por ejemplo Mandriva, Fedora, Centos.
- **Deb**, paquetes Debian, originalmente introducidos por Debian, pero también utilizados por otros como Ubuntu y todos sus derivados.
- .tgz, usado por Slackware, empaqueta el software usando código fuente. Hay además algunas herramientas de más alto nivel para tratar con este formato: slaptget, slackpkg y swaret.
- **Ebuilds**, archivo que contiene información acerca de cómo obtener, compilar e instalar un paquete en el sistema **Portage** de **Gentoo** Linux con el comando **emerge**. Generalmente, estas instalaciones se basan en la compilación de fuentes, aunque algunos paquetes binarios se pueden instalar de esta manera.
- Pacman, para Arch Linux, usa binarios precompilados

3. Evolución de las distribuciones

Antes de que surgieran las primeras distribuciones, un usuario de Linux debía ser experto en programación e informática en general; no solo debía conocer qué bibliotecas y ejecutables necesitaba para iniciar el sistema y que funcionase, sino también los detalles importantes que se requieren en la instalación y configuración de los archivos en el sistema.

Los usuarios vieron en Linux una alternativa a los sistemas operativos DOS, Microsoft Windows en la plataforma PC, Mac OS en Apple Macintosh y las versiones de uso bajo licencia (de pago) de UNIX.

Si bien, históricamente, Linux estuvo mejor posicionado en el **mercado de los servidores**, distribuciones centradas en la facilidad de instalación y uso, tales como Fedora, Mandriva, OpenSuSE, Knoppix y Ubuntu, entre otras, han logrado una mayor aceptación en el mercado doméstico.

La línea de tiempo de evolución de las distribuciones la podemos ver en la imagen de la página 2. Una imagen más detallada la podemos obtener de la siguiente URL: http://upload.wikimedia.org/wikipedia/commons/8/8c/Gldt.svg

4. Licencias de software

Una licencia de software es un contrato entre el autor/titular de los derechos de explotación/distribuidor y usuario consumidor /usuario profesional o empresa del programa informático para utilizar el software cumpliendo una serie de términos y

condiciones establecidas dentro de sus cláusulas.

En la licencia de software se pueden establecer:

- Cesión de determinados derechos del propietario al usuario final sobre una o varias copias del programa informático
- Los límites en la responsabilidad por fallos
- El plazo de cesión de los derechos
- El ámbito geográfico de validez del contrato
- Compromisos del usuario final hacia el propietario:
 - No cesión del programa a terceros
 - No reinstalación del programa en equipos distintos al que se instaló originalmente.

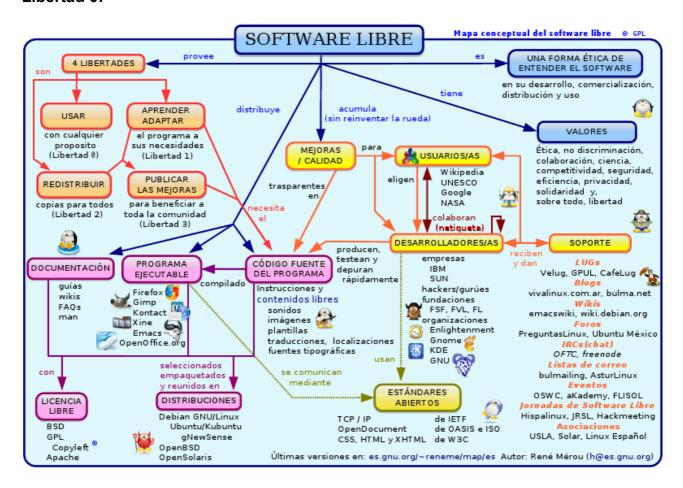
4.1 Software libre y software privativo

El software puede presentarse de dos formas: la herramienta o binario en sí mismo, lo que usted ejecuta y usa; o el código fuente, las instrucciones que el programador escribe y que determinan como funciona el programa.

El software privativo es una caja cerrada la cual no podemos investigar. Al pagar por software privativo no compramos el software, tan solo pagamos por un permiso de uso sumamente limitado en el cual nos definimos como consumidores del mismo.

Existen por definición cuatro libertades básicas que debe poseer un software para ser considerado libre:

Libertad 0:



La libertad de usar el software de la forma que uno quiera. De forma opuesta, muchos programas de computadora privativos no permiten ciertos usos como por ejemplo, crear otro producto que compita con ese software.

Libertad 1:

La libertad de estudiar y mejorar el software. Esto es imposible si no tenemos acceso al código fuente, por eso el Software Libre se caracteriza por distribuirlo además del binario/ejecutable.

Libertad 2:

La libertad de copiarlo y compartirlo. El Software Libre es, como ya dije, un movimiento social. Cree que el compartir es algo bueno y no es un delito. Se han inventado conceptos descabellados como "piratería" para defenestrar el maravilloso acto de ser una persona generosa, una buena persona. El Software Libre defiende esos valores y por eso no es ilegal copiarlo y compartirlo. Es más, su éxito se basa justamente en eso.

Libertad 3:

La libertad de distribuir las mejoras hechas. Gracias a la Libertad 1 podemos aprender como funciona el software, esta libertad nos permite hacer nuestro aporte. Esta libertad es la responsable de generar comunidades de personas que mejoran y comparten sus soluciones unos con otros.

4.2. Licencias de software libre

La mayoría de distribuciones de Linux y el software que incluyen se ofrecen como software libre. Un programa es libre cuando está licenciado bajo una licencia libre.

El número y la diversidad de licencias libres es enorme, muchos proyectos o personas crean su propia licencia que se adapte a sus necesidades o deseos.

Casi todas las licencias libres coinciden en los siguientes aspectos:

- Se distribuye el código fuente.
- Permiten cualquier uso del programa.

A partir de aquí las hay más o menos restrictivas. Otros aspectos que suelen incluir las licencias libres son:

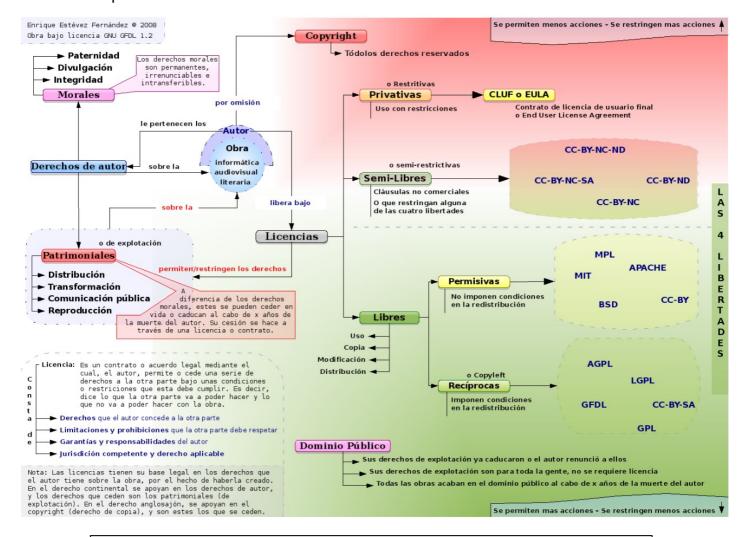
- Atribución se ha de citar y reconocer al autor original
- Compartir igual. Las obras derivadas se han de distribuir con la misma licencia (copyleft).
- Distribución del código. Las obras derivadas han de distribuir el código fuente de los cambios y mejoras realizados.
- Uso comercial. Las obras o programas derivados no pueden distribuirse comercialmente. Este aspecto se suele poder negociar con el autor original del programa.



4.3. Tipos de Licencias libres

Existen diferentes tipos de licencias libres. Algunas cumplen completamente con las cuatro libertades básicas del software libre y otras no. Unas son más restrictivas que otras. Algunos de los tipos de licencias libres más conocidos son:

- GNU GPL
- BSD
- Creative Commons
- MPL. Licencia Pública de Mozilla
- Apache



Lectura complementaria. En el siguiente artículo se ofrece una descripción más detallada de los diferentes tipos de licencias libres http://www.josedomingo.org/web/course/view.php?id=42

Comparativa de Licencias Libres

Con Copileft Robusto

Aseguran:

- Copyleft
- Las cuatro libertades fundamentales del software libre
- Que las modificaciones de éste y el nuevo software derivado sean distribuidos en las mismas condiciones

Licencias de este tipo: GPL

Sin Copileft Robusto

Basadas en proteger "derechos morales" de los autores:

- Sólo mantienen la obligación de mantener los avisos de autoría.
- No contiene obligaciones de copyleft
- Permite la privatización de obras derivadas.
- · Compatibles licencia GPL

Licencia base: BSD (Berkeley Software Distribution).

Licencias derivadas: Apache, XFree86,

Sin Copileft

En muchos casos, derivan de la obligación de publicidad que estaba incorporada en la primera versión de la BSD, pero también pueden surgir de obligaciones:

- Sobre patentes
- Nombramiento
- Indemnizaciones u otros temas.

Los principales ejemplos son la *Netscape Public Licence* (NPL) y la *Mozilla Public Licence* (*MPL*). La licencia MPL se usa para varios programas de mozila.org entre las cuales se encuentran el navegador Mozilla (mozilla.org).

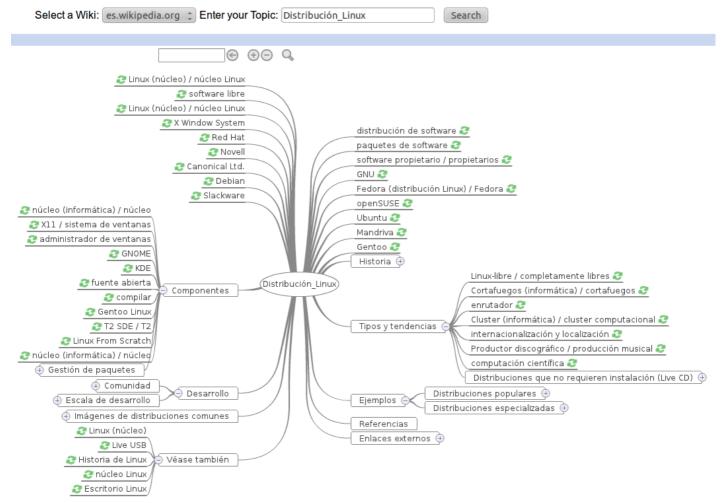
Seudo-libres

Licencias creadas por empresas que intentan beneficiarse del modelo de desarrollo libre. Los ejemplos más importantes son la *Sun Community Source License* (SCSL) y la *Microsoft Shared Source Initiative* (MSSI).

Referencias

Imagen extraída de wikimindmap.org. Resumen de forma visual los conceptos básicos:





Esquema obtenido desde **wikisummarizer.com** a partir de la URL de la wikipedia de distribución Linux:

Una distribución Linux (coloquialmente llamada distro) es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores.

Además del **núcleo Linux**, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU y el sistema de ventanas X Window System.

En el caso de incluir herramientas del proyecto GNU, también se utiliza el término distribución GNU/**Linux**.

Existen distribuciones que están soportadas comercialmente, como Fedora (Red Hat), openSUSE (Novell), Ubuntu (Canonical Ltd.), Mandriva, y distribuciones mantenidas por la comunidad como Debian y Gentoo.

Antes de que surgieran las primeras distribuciones, un usuario de **Linux** debía ser algo experto en Unix; no solo debía conocer qué bibliotecas y ejecutables necesitaba para iniciar el sistema y que funcionase, sino también los detalles importantes que se requieren en la **instalación** y configuración de los archivos en el sistema.

Existía mayor Interés en desarrollar un **sistema operativo** que en desarrollar aplicaciones, interfaces para los usuarios o un paquete de **software** conveniente.

SLS no estuvo bien mantenida; así pues, Patrick Volkerding lanzó una distribución basada en SLS a la que llamó Slackware; lanzada el 16 de julio de 1993.

Los usuarlos vieron en **Linux** una alternativa a los sistemas operativos DOS, Microsoft Windows en la plataforma PC, Mac OS en Apple Macintosh y las versiones de uso bajo licencia (de pago) de UNIX.

La mayoría de estos primeros usuarios se habían familiarizado con el entorno UNIX en sus trabajos o centros de estudios.

Las distribuciones eran originalmente una cuestión de comodidad para el usuario medio, evitándole la instalación (y en muchos casos complación) por separado de paquetes de uso común, pero hoy se han popularizado incluso entre los expertos en éste tipo de sistemas operativos (UNIX/Linux).

Gran parte del software incluido es de fuente abierta o software libre y distribuido por sus desarrolladores tanto en binario compilado como en forma de código fuente, permitiendo a sus usuarios modificar o compilar el código fuente original si lo desean.

RPM, creado por **Red Hat** y usado por un gran número de distribuciones de **Linux**, es el formato de **paquetes** del **Linux** Standard Base.

Un ejemplo podría ser el instalar una versión experimental de alguna de las aplicaciones de la distribución o alguna alternativa (como podría ser utilizar una aplicación de KDE dentro de GNOME o viceversa).

Esto significa que el administrador del equipo tendrá que tomar medidas adicionales para mantener el **software** actualizado.

Summary

Linux (100)o

software (76)

paquetes (73)o

código (56)o

Distribución Li...

código fuente (36)

instalación (33)o

Red Hat (33)

núcleo Linux (30)o

sistema operativo (26)

software libre (26)