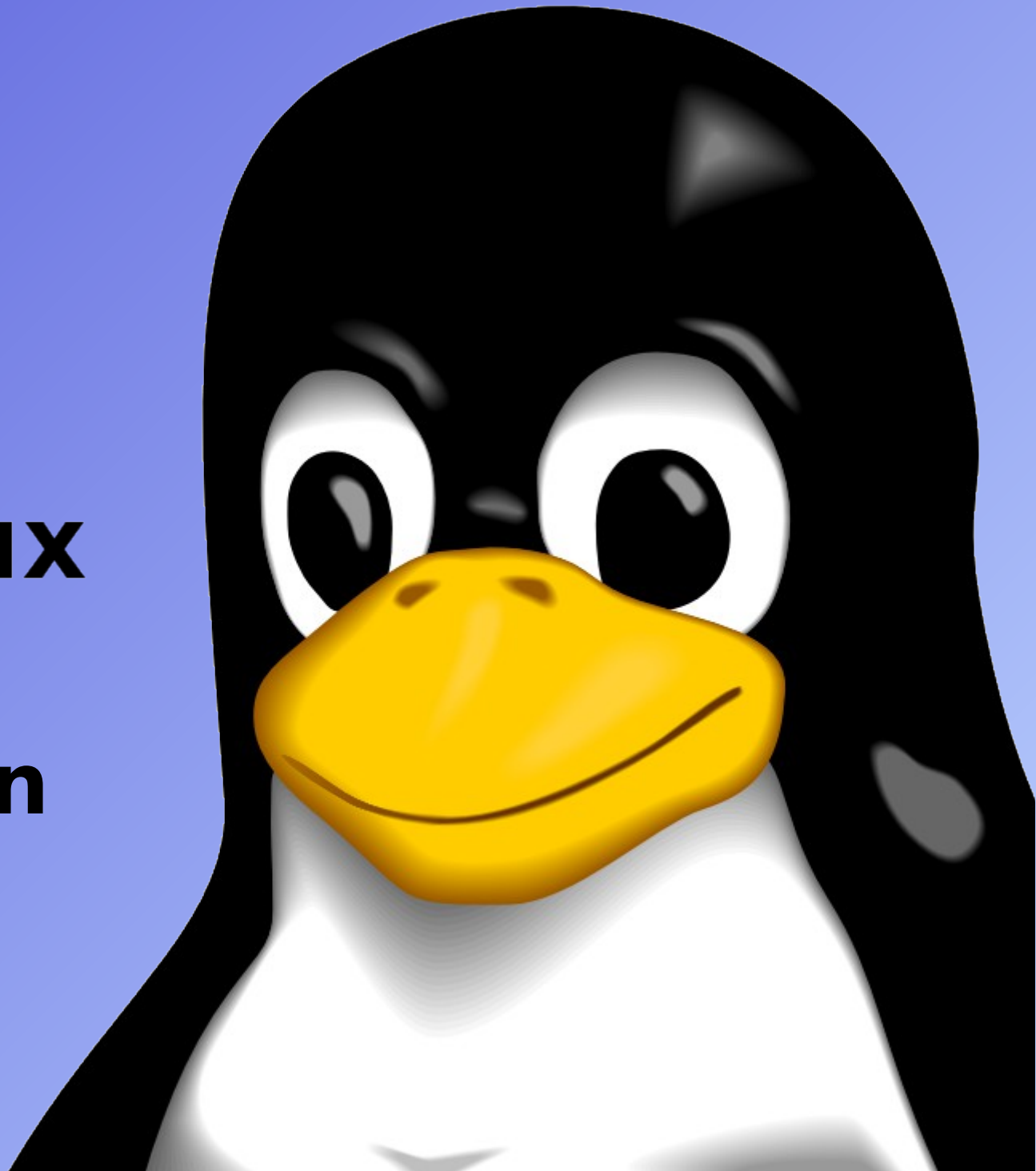


GNU/Linux

**Una
introducción**



Introducción a GNU/Linux

**Basado en el curso de GNU/Linux para seres humanos dictado por el
BBLUG en 2007**

Autores:

Matias D'Ambrosio
Pablo Odorico
Lisandro D. N. Pérez Meyer
Martín Ribelotta
Rodolfo Brasnarof

Modificaciones: 2014 Lisandro D. N. Pérez Meyer

Presentaciones: <http://perezmeyer.com.ar/files/introlinux/>

Licencia *GNU Free Documentation License*:
<http://www.gnu.org/copyleft/fdl.html>



¿Qué es un Sistema Operativo?

Para explicar lo que es Linux, primero hay que explicar qué es un Sistema Operativo, y el asunto es que un sistema operativo se supone que no debe ser visto, porque realmente nadie usa un sistema operativo, sino los programas en sus PCs.

Y la única misión de un sistema operativo es ayudar a esos programas a ejecutarse.

Así que un sistema operativo no hace nada por si mismo, solo espera a que otros programas le pidan ciertos recursos, ficheros del disco, o conexión con el mundo exterior.

- *Linus Torvalds*



El Software Libre

Libertad 0: el software se puede usar. Es la libertad que nos otorga casi cualquier software.

Libertad 1: el software se puede modificar. Es decir, se puede personalizar, mejorar, adaptar para las necesidades particulares de un determinado usuario.

Libertad 2: el software se puede distribuir. Es decir, se puede copiar, vender, prestar o compartir a las personas que el usuario desee, sin tener que pedir permiso al autor del software.

Libertad 3: el software se puede distribuir modificado. Se trata de una suma de la 1 y la 2. Permite que las mejoras que un usuario le haya hecho a un determinado software puedan compartirse con otros usuarios.



El Software Libre

En general las ideas del Software Libre buscan:

Promover la generación de mejor software (a través de la suma de pequeños aportes de desarrolladores y usuarios)

Colaborar para que toda la sociedad se vea beneficiada con los avances del software

Logramos mejorar la sociedad al tener disponibles más y mejores herramientas.



El Software Libre

"Si tú tienes una manzana y yo tengo una manzana, e intercambiamos las manzanas, entonces tanto tú como yo seguiremos teniendo una manzana. Pero si tú tienes una idea y yo tengo una idea, e intercambiamos ideas, entonces ambos tendremos dos ideas."

- George Bernard Shaw



El Software Libre



Software libre **NO** implica
software gratis.



El Software Libre

Los usuarios, la calidad del software y la seguridad



Los usuarios son incentivados a participar:

Reportando *bugs*

Documentando y traduciendo

Dando a conocer el software

Generando...

Software de mayor calidad

Reducción en tiempos de corrección

Mayor movimiento en el desarrollo

Incentivo a los desarrolladores



El Software Libre

Las libertades individuales



Uno de los principales objetivos del software libre es proteger los derechos y las libertades individuales de los usuarios:

Software con seguridad y privacidad transparente (código fuente disponible, protocolos abiertos, etc.)

Sin ningún tipo de **D**igital **R**ights **M**anagement



El proyecto GNU



“GNU's Not Unix”

En 1984, **Richard M. Stallman** fundó el Proyecto GNU con el objetivo de conseguir un sistema operativo **libre y abierto**. A partir de ese momento, un gran número de colaboradores se fueron sumando al proyecto, desarrollando software libre para reemplazar cada una de las herramientas del sistema UNIX.



Free Software Foundation



La FSF es una organización sin fines de lucro fundada por RMS en 1985 dedicada a promover los derechos del usuario de computadoras a usar, estudiar, copiar, modificar y distribuir programas.

Resguarda Copyrights

Desarrolla licencias

Asiste legalmente a desarrolladores

Denuncia violaciones a la GPL

Mantiene el proyecto GNU

Proteje los derechos del usuario



El movimiento Open Source



Fundada en 1998 por Bruce Perens y Eric S. Raymond para darle más aceptación comercial al software libre.

Creada debido a la ambigüedad de la palabra *Libre* en inglés (*Free*)

Creadores de la Open Source
Definition

Se basa en los beneficios tecnológicos del Software Libre



El movimiento Open Source

Open Source Definition

Libre distribución

Código fuente

Trabajos derivados

Integridad del código fuente original

Sin discriminación de personas o grupos

Sin discriminación de áreas de iniciativa

Distribución de la licencia

La licencia no debe ser específica de un producto

La licencia no debe restringir otro software

La licencia debe ser tecnológicamente neutral



¿Qué es una licencia?

Es un documento legal que especifica qué libertades se otorgan y cuales se niegan a los usuarios.

Usar software sin cumplir con su licencia es **ILEGAL**



Licencias propietarias

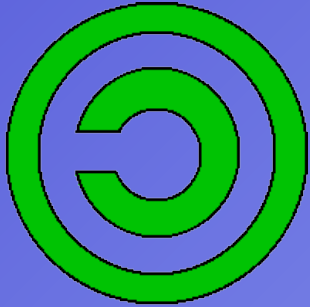


MS EULA
Shareware
Freeware
Etc.

Venta de ***permisos para usar*** el software
Código fuente no disponible u obligación de
aceptar NDAs cuando se ve el código
Algunos prohíben la *ingeniería inversa*



Licencias libres: GPL v2



Gnu
Public
License

Finalizada en
Junio de 1991

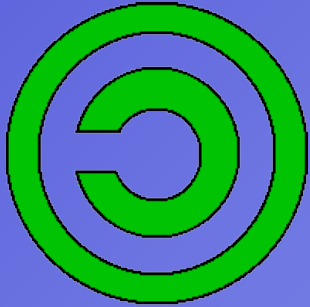
Asegura las 4 libertades

La mayor cantidad de Software Libre es GPL

El software *“Nace libre y muere libre”*



Licencias libres: GPL v3



Gnu
Public
License

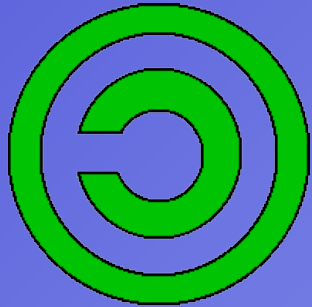
Finalizada en
Junio de 2007

Compatibilidad con más licencias libres

Nuevos términos contra:

- Patentes
- DRM
- Otras restricciones a las 4 libertades





Lesser
Gnu
Public
License

Finalizada en
Junio de 1999

Fué creada como un compromiso entre una licencia “fuerte” (GPL) y licencias mas permisivas (BSD, MIT)

Permite que el software sea *linkeado* con software no (L)GPL, sea éste tanto libre como propietario
Impone las 4 libertades sobre el código bajo ésta licencia pero no sobre el código al que fué linkeado



Licencia BSD



Berkeley
Software
Distribution

Última revisión
en 1999

Originalmente creada para BSD Unix

Es una licencia permisiva

Permite que nuevas restricciones se le apliquen al software (por ejemplo, cerrarlo)

Similar a la licencia MIT, NetBSD, FreeBSD y OpenBSD



UNIX



Bell Laboratories



Desarrollado en los 60' y 70' principalmente por Ken Thompson y Dennis Ritchie

Portable, Multitarea, Multiusuario, desarrollado en C, *"Todo es un archivo"*, modelo cliente/servidor, modular... **Filosofía UNIX**

Muchos sistemas operativos, entre ellos GNU/Linux, siguen con esta filosofía



Linux Kernel



- Licencia GPL
- Escrito principalmente en C
- Compatible con UNIX y POSIX
- Portable
- Monolítico
- Multitarea
- Soporte de memoria virtual
- Linking dinámico de librerías
- Soporte de Threading
- Etcetera...

Las primeras versiones fueron desarrolladas por Linus Torvalds en 1991 como un remplazo de MINIX. Hoy en día es mantenido por cientos de desarrolladores. Linus es principalmente el “*Dictador Benevolente*” del proyecto.



Linux Kernel

La necesidad del proyecto GNU

Si bien RMS junto a la FSF estaban desarrollando el sistema GNU, se encontraron en necesidad de un núcleo, ya que Hurd, su kernel propio, estaba (y sigue estando) siendo desarrollado muy lentamente debido a su diseño de micro-kernel.

Luego de que Linus Torvalds liberó Linux con licencia GPL, el proyecto GNU decidió adoptarlo, creando así al sistema GNU/Linux.



POSIX



Portable
Operating
System
Interface
X

IEEE 1003

Pretende generalizar las interfaces de los sistemas operativos para que las aplicaciones se ejecuten en distintas plataformas (por medio de una API genérica)

Sistemas que lo cumplen:

- Linux
- BSD
- WindowsNT (soporte parcial)



Fuentes de información

Filosofía

Read

The

F*****

Manual

Antes de hacer una pregunta,
buscar en internet y leer la
documentación!

*“I can explain it to you, but I can't
understand it for you.”*

google.com

wikipedia.org

Ayuda de KDE y Kubuntu

man [comando]

/usr/share/doc/

/usr/doc

Grupos de usuarios



Fuentes de información

Listas de correo e IRC

De software o grupos de usuarios

Muchas veces hay en varios idiomas

Usar la Nettiquete, preguntar
inteligentemente y tener paciencia!

<http://download.bblug.usla.org.ar/netiquette.png>

<http://www.sindominio.net/ayuda/netiquette.pdf>



Las Distribuciones

El porqué de su existencia

Antes de que existieran las distribuciones, un usuario de Linux debía ser una especie de “experto” en Unix que supiera no sólo que librerías y ejecutables se necesitan, sino también detalles específicos de configuración de cada uno de ellas.

Empiezan a aparecer con los primeros usuarios de Linux no programadores, que se sentían más interesados en un sistema operativo usable y no en aplicaciones específicas

Slackware es la más antigua de ellas aún activa



Las Distribuciones

Diferencias entre ellas

Objetivo de uso

Filosofía

Nivel de conocimiento del usuario

Modelo de seguridad

Directorios de instalación

Grupos y usuarios predefinidos

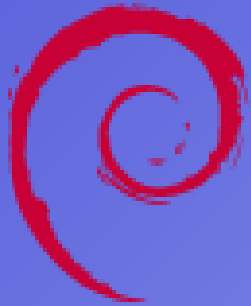
Sistema de arranque

Sistema de instalación

Sistema de paquetes



Distribuciones



debian

Agosto de 1993

Originalmente se basó en Linux, creando Debian GNU/Linux. Actualmente soporta otros kernels como Hurd y NetBSD

Conocida por su adherencia a las filosofías de Unix y Software Libre

Abundante cantidad de opciones: mas de 30.000 paquetes por arquitectura, soportando once arquitecturas, entre ellas i386, X86-64 y ARM

Sirvió de sistema base para un gran número de otras distribuciones, como ser Knoppix, Ubuntu y derivados, Mephis,...

Sistema de paquetes APT (**A**dvanced **P**ackaging **T**ool)



Distribuciones



Ubuntu

Octubre del 2004

Ubuntu: “Linux for human beings”

Basada en Debian GNU/Linux, pero con mayor enfoque en la *usabilidad*, utilizando así el sistema deb/APT

Centrada en el escritorio Gnome y las librerías GTK (**G**imp **T**ool **K**it)

Envía CDs a domicilio *igratuitamente!*



Distribuciones



Kubuntu

Abril del 2005

Subproyecto de Ubuntu

Kubuntu: “hacia la humanidad” en Bemba

Comparte todas las características de Ubuntu, a excepción de que se basa en KDE (**K Desktop Environment**) y las librería QT de Trolltech

También envía CDs a domicilio *gratuitamente*

Distribución elegida para el curso



Distribuciones



Red Hat

Noviembre de 1994

Creada por Red Hat, Inc.

Es la *distro* mas antigua orientada al usuario final

Sistema de paquetes RPM (**RPM Packet Management**)

Antecesora de Mandrake (ahora Mandriva) y de Fedora Core

También existe Red Hat Enterprise Linux, versión comercial pensada para la empresa

Originalmente solo traía el escritorio Gnome



Distribuciones



Fedora Core

Noviembre del 2003

Nace a partir del fin de producción de Red Hat Linux, sponsoreada por Red Hat y desarrollada por la comunidad Fedora, existente desde antes del soporte de Red Hat

Sin soporte técnico oficial de Red Hat

Orientada por defecto al escritorio Gnome



Distribuciones



Gentoo

Marzo del 2003

Tiene una comunidad de usuarios muy activa
Su uso puede ser “difícil” para usuarios nuevos
Sistema de paquetes *Portage* en el cual los programas se configuran y compilan automáticamente al ser instalados. Esto transforma a gentoo en una *meta-distribución*
Portage hace a la distribución muy configurable, de alto rendimiento, y a su vez portable
Portage tiene software *muy* actualizado
Usa una versión modificada del init SystemV



Distribuciones

¿Éstas son **todas** las distribuciones?

NO

Existen muchas mas distribuciones, como:

Arch

Suse

Red Flag (China)

Especializadas (gubernaciones, dispositivos, etc.)

Incluso existen *distribuciones Argentinas*:

Ututo (Live-CD, ianterior a Knoppix!)

Ututo-e (ahora mayormente conocida por Ututo):

basada en Gentoo Linux, es la única distribución

100% Libre en términos del proyecto GNU.

Tarea, mirar DistroWatch en: www.distrowatch.com



Paquetes y repositorios

El concepto de repositorios

Los programas de GNU/Linux no se bajan “*de cualquier lado*” y se instalan “*en cualquier lado*”

El manejo de software es **centralizado**, esto es posible gracias a los sistemas de paquetes y el hecho de que el software sea libre



Paquetes y repositorios

Organización de Ubuntu y derivados

	Software libre	Software no libre
Soportado	Main	Restricted
No soportado	Universe	Multiverse



Sistemas de paquetes

Advanced Packaging Tool

Sistema de manejo de paquetes utilizado por Debian y derivados

Originalmente desarrollado para funcionar como *front-end* del sistema *dpkg*, que a su vez utiliza archivos *.deb*

Capaz de descargar, instalar, actualizar, remover y compilar (opcionalmente) paquetes de repositorios APT

Realiza chequeo y descarga automática de dependencias al instalar un programa.

No es un programa per se, APT es una librería con funciones utilizadas por programas específicos como *apt-get*, *apt-cache* y *aptitude*

Posee chequeo automático de dependencias



Sistemas de paquetes

*Advanced **P**ackaging **T**ool*

Los repositorios se configuran en `/etc/apt/sources.list`
Existen *GUIs* que facilitan su uso, como Synaptic

“Éste APT tiene los poderes de Súper Vaca”

El mensaje anterior sirve de pista para ejecutar en una consola:

```
$ apt-get moo
```

Tarea para el hogar :-)



Sistemas de paquetes

RPM Package Manager

Sistema creado por Red Hat

Agrega, actualiza, quita, verifica y solicita paquetes

Funciona con una base de datos, donde guarda la información de los paquetes instalados

Verificación criptográfica

DeltaRPMs (actualizaciones a partir de *diffs*)

Conoce sus dependencias, pero no las resuelve automáticamente

Requiere de Yum o urmpi para resolver dependencias de forma automática



Sistemas de paquetes

Portage

Usado por Gentoo

Basado en la idea de *BSD Ports*

Instala directamente desde el código fuente del programa.

Capaz de descargar, configurar, instalar, actualizar,
remover paquetes del *Portage Tree*

Muchos paquetes están también disponibles ya
compilados (versión binaria)

Los “paquetes” llamados *ebuilds* no incluyen las fuentes,
sino información de dependencias, configuración, etc.

Funciona también en FreeBSD, Mac OS X y Solaris

Puede instalar varias versiones del mismo software

La configuración en la compilación es global por medio de
USE Flags y *flags* del compilador



¿Mejor, igual, peor?

GNU/Linux comparado con otros SSOO

Debate :-D



Fin

¿Preguntas?



Fin

Gracias

