

Módulo 01

El Sistema Operativo

GNU/Linux



Introducción al Sistema
Operativo GNU/Linux
DCIC - UNS



Copyright

- Copyright © **2005-2017** A. G. Stankevicius
- Se asegura la libertad para copiar, distribuir y modificar este documento de acuerdo a los términos de la **GNU Free Documentation License**, Versión 1.2 o cualquiera posterior publicada por la Free Software Foundation, sin secciones invariantes ni textos de cubierta delantera o trasera.
- Una copia de esta licencia está siempre disponible en la página <http://www.gnu.org/copyleft/fdl.html>.
- La versión transparente de este documento puede ser obtenida de la siguiente dirección:

<http://cs.uns.edu.ar/~ags/teaching>



Contenidos

- ¿Qué es **GNU/Linux**?
- El Proyecto **GNU** y el Software Libre.
- Evolución histórica de **GNU/Linux**.
- Proyección de fragmentos del documental “Revolution **OS**”.
- Discusión de los conceptos abordados.
- Mitos acerca del Software Libre.
- **GNU/Linux**: ¿¿Por qué no ahora??



¿Qué es GNU/Linux?

- **GNU/Linux** es un **sistema operativo...**
- ...pero, ¿qué es un sistema operativo?
 - Un sistema operativo es el encargado de gestionar la **interacción entre el usuario y la computadora** y de **administrar los recursos disponibles de manera eficiente.**
- No es posible usar una computadora sin su respectivo sistema operativo.
 - Ejemplo muy conocido: MS-Windows.
 - Ejemplo no tan conocido: MacOS.



¿Qué es el proyecto GNU?

- El **proyecto GNU** fue iniciado por Richard M. Stallman con el propósito de crear un sistema operativo completo y libre: el sistema **GNU**.
- **GNU** es un acrónimo recursivo que significa “**GNU's Not Unix**” (**GNU** no es Unix).
 - Además es el nombre en inglés de este bicho → (un tipo de antílope)



Hardware vs. software

- El **hardware** (HW) de una computadora es todo aquello que se puede tocar.
- El **software** (SW) de una computadora es todo el resto.
- Por ejemplo:
 - El sistema operativo (**SO**, también **OS**).
 - Las aplicaciones.
 - Los juegos.



Software libre

- La Free Software Foundation (**FSF**), fue fundada por Richard M. Stallman para promover el uso del **software libre**.
- Pero, una vez más, ¿qué significa que el software sea libre?
- ¡“Free Software” significa “software libre” y no “software gratis”!
- La ambigüedad viene del inglés, donde tanto “libre” como “gratis” se escriben igual: “free”.



Las cuatro libertades

- **Libertad 00:** la libertad para ejecutar el programa con cualquier fin.
- **Libertad 01:** La libertad de estudiar cómo trabaja el programa y de adecuarlo para que haga lo que usuario desee.
- **Libertad 10:** la libertad de redistribuir el programa de manera de colaborar con el resto de la sociedad.
- **Libertad 11:** la libertad de hacer pública y distribuir a terceros la versión mejorada.



¿Qué es el código fuente?

- El **código fuente** de un programa es la receta que explica cómo es que hace lo que hace.
 - ➔ El código fuente de los programas se escribe en algún lenguaje de programación.
 - ➔ Los programas pueden ser distribuidos con o sin su código fuente.
- Nótese que para ejercer las libertades **01** y **11** hace falta **tener acceso al código fuente**.



Evolución histórica

- El software nació libre.
- En la década del '60, al comprar **HW**, se accedía al catálogo del **SW** del fabricante.
- Todo el software se distribuía junto con su código fuente.
- Al final de los '70, **IBM** anunció que iba a vender parte de su **SW** por separado.
- A partir de entonces, el **SW** propietario (no libre) pasó a ser moneda corriente.



Evolución histórica (cont)

- **1983:** Richard M. Stallman (**RMS**) anuncia la creación del proyecto **GNU**.
- Un año después funda la Free Software Foundation (**FSF**).
- Primeros componentes del sistema **GNU**: un compilador de **C** (**gcc**), un editor de texto (**emacs**) y un depurador (**gdb**), todos escritos por **RMS**.



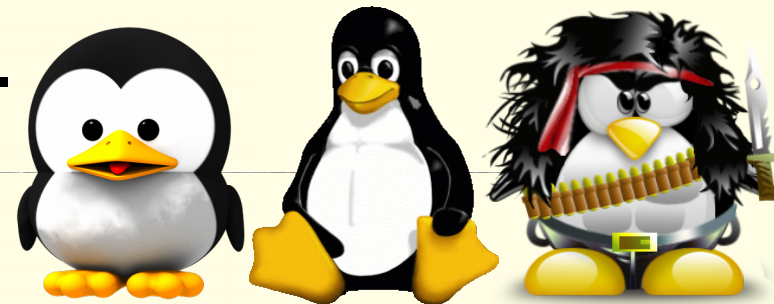
Evolución histórica (cont)

- Para garantizar las cuatro libertades, **RMS** inventó el concepto del copy-left (la inversa del copy-right).
- **1990**: El sistema **GNU** estaba casi completo, sólo faltaba terminar el kernel (núcleo).
- **1991**: Linus Torvalds tomando como inspiración al sistema operativo Minix escribió un kernel para el cual eligió que sea distribuido como software libre.



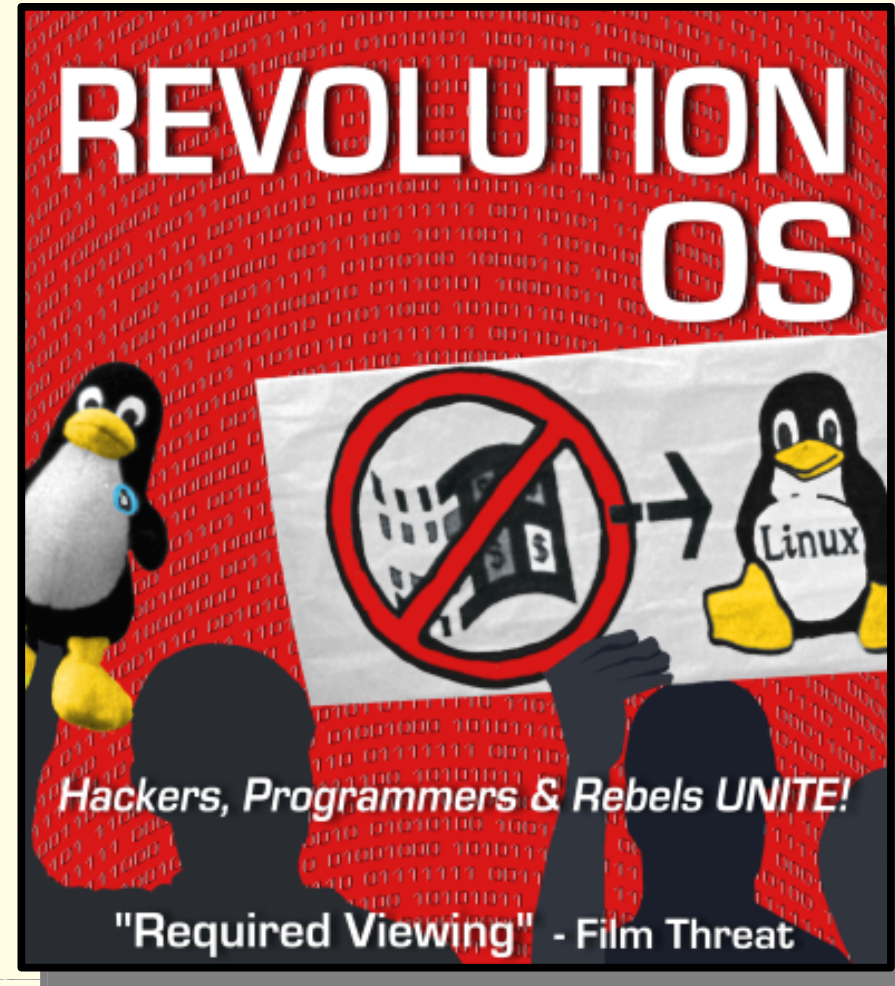
Evolución histórica (cont)

- La unión del sistema operativo **GNU** (al que le sólo faltaba el kernel), con el kernel que diseñó Linus Torvalds (en la actualidad conocido como Linux), dio lugar al Sistema Operativo **GNU/Linux**.
- **1996:** Un pingüino mordió a Linus, por lo que eligió al pingüino Tux como mascota.



Revolution OS

- A continuación vamos a repasar pasajes del documental **Revolution OS** (2001), dirigido por J. T. S. Moore.



Licencias de software

- ¿Qué vende Microsoft?
 - ¿Software?
 - **¡¡NO!!**
- Las compañías de software privativo no venden software, venden “permisos” para usar su software.
- El documento legal que especifica qué libertades se le otorgan y cuáles se le niegan a los usuarios se denomina **licencia**.



General Public Licence

- La **GPL** es la licencia de software libre más utilizada del mundo, según **Freecode**:
 - La máxima adopción fue alcanzada en el año 2009 con un **72%** de los proyectos usándola.
 - Actualmente se mantiene por arriba del **50%**.
- Esta licencia **le asegura al usuario las cuatro libertades** que comentaba Richard Stallman.
 - Toda trabajo derivado de código licenciado bajo **GPL** sólo puede ser distribuido bajo la misma licencia.



Software código abierto

- El concepto de “**Open Source**” (código abierto) fue propuesto para evitar la citada ambigüedad de la palabra “free”.
 - ➔ ¡Los inversores salían corriendo al escuchar hablar de “Software Gratis”!
- La principal diferencia entre las comunidades de software libre y de código abierto es dónde eligen poner el énfasis.
 - ➔ La comunidad de software libre **prioriza la importancia de la libertad por sobre las cuestiones técnicas.**



Open Source Definition

- Libre redistribución.
- Acceso al código fuente.
- Permitir trabajos derivados.
- Respetar la integridad del código fuente original.
- Sin discriminación hacia personas o grupos.
- Sin discriminación de dominios de aplicación.



Open Source Definition (cont)

- La licencia debe poder ser redistribuida.
- La licencia no debe ser específica a un producto.
- La licencia no debe afectar a otros programas que se distribuyan a la par.
- La licencia debe ser tecnológicamente neutra.



Preguntas para reflexionar...

- El Software Libre, ¿siempre es gratis?
- Entonces, ¿cómo se puede ganar dinero en torno al Software Libre?
- La libertad de uso, ¿me permite compartir un programa libre con otros?
- ¿Qué diferencia al Software Libre del Software Código Abierto?
- ¿Qué era la **GPL**? ¿Es una licencia Libre o es una licencia Open Source?



Preguntas para reflexionar...

- ¿Qué es la **OSI**?
- ¿**OSI** certificó la **GPL**?
- ¿Por qué algunos dicen **GNU/Linux** en vez de simplemente **Linux**? ¿Cuál es el correcto?
- ¿Por que razón una persona querría invertir su tiempo en colaborar con la comunidad de Software Libre?
- Yo no soy programador... ¿podré colaborar de alguna otra forma?



Mitos acerca del SL

- Toda revolución recorre religiosamente las siguientes etapas:
 - Primero te ignoran.
 - Luego te ridiculizan.
 - Más tarde te atacan.
 - Finalmente, tu ganas.
- En la actualidad están en boga un conjunto de mitos infundados en torno al software libre...
 - ¿Estaremos en plena **tercera etapa**?



Mitos acerca del SL

● El software libre es software ilegal:

- El software legal viene en **CDs** y **DVDs**, y el software que descargamos de internet es ilegal.
- Ambas afirmaciones son incorrectas, puesto que el software legal a veces viene en **CDs** y **DVDs**, así como el software que descargamos puede o no ser ilegal.
- Más aun, en el caso del software libre, **el 100% del software que usemos**, ya sea descargado de internet o bien copiado por un amigo, **ha sido, es y seguirá siendo software legal.**



Mitos acerca del SL

- El software comercial le da más derechos legales a los usuarios.
 - ➔ El software comercial cuenta con una licencia y en ocasiones con un **EULA** que claramente explicita los derechos y también las obligaciones de los usuarios.
 - ➔ Este punto es usualmente cierto, pero el mismo argumento también se aplica al software libre, ya que **el software libre también se distribuye bajo una cierta licencia**, por ejemplo, la **GPL**.



Mitos acerca del SL

● Lo bueno sale caro:

- Si por el software pago nada, seguro que la calidad debe ser acorde, es decir, bastante pobre.
- Nuevamente, esto es falso. Puede que exista software libre de calidad cuestionable, de la misma forma que existen programas privativos de mala calidad.
- Más aun, distintos estudios científicos sostienen que **la metodología de desarrollo del software libre produce software de mejor calidad** que el software desarrollado de una manera más tradicional.



Mitos acerca del SL

● El software libre es difícil de usar:

- Yo probé un par de programas libres y la verdad que todos me resultaron difíciles de usar.
- Esta afirmación parece ser parcialmente cierta, pero **no se trata de la facilidad de uso sino más bien de la familiaridad que tengamos con la pieza de software.**
- La experiencia indica que llamativamente los usuarios con menos conocimientos tienen menos objeciones a la hora de adoptar software libre, posiblemente porque a diferencia de los usuarios experimentados están mucho menos condicionados.



Mitos acerca del SL

● El software libre está plagado de errores:

- El software libre se actualiza muy seguido, corrigiendo errores que se van detectando con el paso del tiempo. Evidentemente está mal diseñado o mal programado.
- Esta afirmación absolutamente falsa, hay que tener en cuenta que **el no haber detectado errores no asegura que no los haya**. Por el contrario, recién una vez detectado un cierto error será posible tomar los recaudos necesarios para corregirlo.
- Una medida de la calidad es el tiempo desde que un error es detectado hasta que se lo corrige.



Mitos acerca del SL

● El software libre es inseguro:

- Los potenciales atacantes tiene a disposición el código fuente, por lo que el software libre es mucho más fácil de atacar o de comprometer.
- Esta afirmación no sólo es incorrecta, es además peligrosa. El principio criptográfico de Kerckhoffs nos recuerda que un sistema tiene que ser seguro incluso cuando se conozca el funcionamiento interno de cada uno de sus componentes.
- La seguridad de un sistema dependerá de las políticas que implemente y no de la licencia que adopte.



Mitos acerca del SL

● Nadie usa software libre:

- No conozco a nadie que use software libre, no quiero quedar como extraterrestre con mis amigos.
- Es difícil analizar este planteo por tratarse de una cuestión subjetiva. Sin embargo, **el software libre está en uso prácticamente en todas partes**, más allá de que nosotros nos demos cuenta.
- Por ejemplo, la infraestructura de Google está basada íntegramente en **GNU/Linux**, Firefox y Chrome, los navegadores de los más usados en el mundo, cuatro de cada cinco servidores web corren Apache, etc.



Mitos acerca del SL

● Software libre significa software gratis:

- En esencia que un cierto software sea libre significa que se lo puede conseguir de manera gratuita.
- Es cierto que cualquier software libre tiene que poder ser obtenido de manera gratuita (¿por qué razón? ¿qué libertades están en juego?), pero que un cierto programa sea libre es mucho más que eso.
- En particular, **si un cierto software es libre tenemos la certeza que asegura a sus usuarios las cuatro libertades esenciales.**



Mitos acerca del SL

- Nadie hace dinero con el software libre:
 - ➔ Yo soy programador, y quiero hacer dinero con la computación. Si “regalo” mis programas, ¿cómo hago para pagar el alquiler o ponerle nafta a la moto?
 - ➔ Este es un buen punto, pero **al limitar la generación de ganancias a la venta del software** (o del permiso de uso para hablar con propiedad), **estamos perdiendo de vista las restantes posibilidades** tales como el servicio de soporte, de capacitación, de instalación, de personalización, etc.



Mitos acerca del SL

- El software comercial está mejor soportado:
 - ➔ Como el software libre lo puedo consigo libre de costo, ¿quién se va a encargar de ayudarme cuando tenga algún inconveniente?
 - ➔ Este planteo es falaz, ya que **existen múltiples fuentes de soporte técnico a disposición de los usuarios de software libre**. Siempre está la opción de contratar un servicio de soporte, pero también es posible apelar a las comunidades en torno a los distintos programas.
 - ➔ El software comercial generalmente evidencia un nivel de soporte más pobre que el software libre.



Mitos acerca del SL

- En algunos cursos introductorios al sistema operativo **GNU/Linux** señalan que todo parece indicar que el movimiento de software libre está en la tercera etapa, la etapa de ser atacado.
- Esto es incorrecto, el advenimiento del sistema operativo para celulares **Android**, el cual está basado íntegramente en **GNU/Linux** es la más clara evidencia de que ya estamos en la etapa cuatro: ¡tu ganas!
 - Android **está presente en el 85%** de los celulares.



¿¿Por qué no ahora??

- Todos los años se viene diciendo “**GNU/Linux** está listo para el **PC** de escritorio”...
- ...pero el **PC** de escritorio, ¿sigue viniendo con otro sistema operativo!
 - ➔ Hay mucho dinero en juego.
 - ➔ Faltaban lijar mejor algunas asperezas.
- Hoy en día esto parece haber cambiado... entonces: ¿por qué no ahora **GNU/Linux** también en mi **PC**?



¿Preguntas?

