

Programación y Administración de Sistemas

1. Introducción a la administración de sistemas

Pedro Antonio Gutiérrez

Asignatura "Programación y Administración de Sistemas"

2º Curso Grado en Ingeniería Informática

Escuela Politécnica Superior

(Universidad de Córdoba)

pagutierrez@uco.es

7 de febrero de 2017



Objetivos del aprendizaje I

- Distinguir entre las distintas generaciones de sistemas operativos y cómo va surgiendo la necesidad de administrar dichos sistemas operativos o servicios.
- Enumerar los sistemas operativos más usuales.
- Definir el sistema informático de una organización.
- Definir el Departamento de Informática, sus funciones y su organización.
- Definir el rol del administrador de sistemas dentro de una organización.
- Enumerar las tareas típicas del administrador de sistemas, incluyendo tareas a nivel *hardware*, mantenimiento *software* y documentación, soporte a usuarios, servicios, seguridad y copias de seguridad.
- Definir estrategias y consejos genéricos en administración de sistemas.

Objetivos del aprendizaje II

- Explicar qué es GNU/Linux, cómo surge, su filosofía y las principales distribuciones de las que disponemos, destacando las más adecuadas para administración de sistemas.
- Definir qué es *software* libre y dar ejemplos de *software* que no sea libre.
- Enumerar las ventajas del *software* libre para la administración de sistemas.
- Identificar el nivel de calado del *software* libre en administración de sistemas.
- Enumerar desventajas del *software* libre.
- Enumerar ventajas y desventajas del uso de interfaces gráficas de usuario en administración de sistemas.
- Saber qué es un superusuario y cómo se pueden realizar tareas como superusuario mediante la herramienta `sudo`.
- Usar herramientas de GNU/Linux para comunicarse con otros usuarios del sistema.

- 1.1. Un breve recorrido por los sistemas operativos.
 - 1.1.1. Generaciones de los sistemas operativos.
 - 1.1.2. Sistemas operativos más usuales.
- 1.2. El sistema informático y la organización.
 - 1.2.1. El sistema informático.
 - 1.2.2. El Departamento de Informática.
 - 1.2.2.1. Funciones del Departamento de Informática.
 - 1.2.2.2. Organigrama del Departamento de Informática.
- 1.3. La figura del administrador de sistemas.
 - 1.3.1. Rol.
 - 1.3.2. Tareas detalladas.
 - 1.3.2.1. Nivel *hardware*.
 - 1.3.2.2. Mantenimiento *software* y documentación.
 - 1.3.2.3. Soporte a usuarios.
 - 1.3.2.4. Servicios.
 - 1.3.2.5. Seguridad.
 - 1.3.2.6. Copias de seguridad.
 - 1.3.3. Estrategias.

1.3.4. *Software* libre.

- 1.3.4.1. ¿Qué es GNU/Linux?.
- 1.3.4.2. Distribuciones de GNU/Linux: Debian, Ubuntu, Red Hat, CentOS, Fedora.
- 1.3.4.3. ¿Qué es *software* libre?.
- 1.3.4.4. ¿Qué no es *software* libre?.
- 1.3.4.5. Ventajas del *software* libre en administración de sistemas.
- 1.3.4.6. Uso del *software* libre en administración de sistemas.
- 1.3.4.7. Desventajas del *software* libre.
- 1.3.4.8. Interfaces gráficas de usuario para administración de sistemas.

1.4. Superusuario dentro del sistema.

1.4.1. De usuario a superusuario.

- 1.4.1.1. El superusuario o administrador.
- 1.4.1.2. La herramienta sudo.

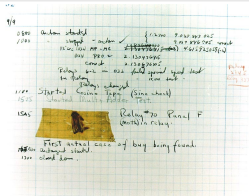
1.4.2. Comunicación con el resto de usuarios.

- Cuestionarios objetivos.
- Tareas de administración.

Generaciones de los sistemas operativos

Primera generación (1945-1955)

- Ordenadores muy voluminosos, tarjetas perforadas.
- No necesitaban Sistema Operativo (SO), el operario introducía la tarjeta con el código correspondiente.



Fuente: https://es.wikipedia.org/wiki/Error_de_software



Fuente: https://en.wikipedia.org/wiki/Patch_%28computing%29



Generaciones de los sistemas operativos

Segunda generación (1955-1968)

- Aparecen los transistores, los ordenadores disminuyen de tamaño y se puede empezar a pensar en SOs.
- Lenguaje de Control de Tareas (JCL).
- Lenguajes de **alto nivel** y de **bajo nivel** (*assembler*).
- Se comienza a hablar de **superusuarios** y de usuarios.
- Dispositivos de **entrada/salida** (cintas magnéticas...).



Generaciones de los sistemas operativos

Tercera generación (1968-1981)

- Aparecen los circuitos integrados (*Large Scale Integration*, **LSI**), que incluyen multitud de componentes electrónicos.
- Comienzan a aparecer equipos **de propósito general**.
- Nuevos conceptos: escalabilidad, **multiprogramación**, discos duros.
- Multiprogramación → *buffering*, gestión de procesos en el SO...
- Infrautilización de los ciclos de CPU → sistemas de tiempo compartido.



Generaciones de los sistemas operativos

Cuarta generación (1981-2001)

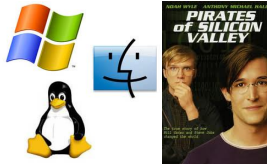
- Aparecen los *Very Large Scale Integration*, **VLSI**, y los **microprocesadores**.
- **Conectividad** de dispositivos (*Plug and play*).
- Aplicaciones **cliente/servidor**.
- **Máquinas virtuales, porciones virtuales...**

Quinta generación

- Tendencia a que los ordenadores los puedan manejar personas **no expertas** en la informática.



Sistemas operativos más usuales



- **Mac OS**: es el nombre del primer sistema operativo de Apple para los ordenadores Macintosh.
- **Windows**: Es un sistema operativo gráfico para ordenadores personales cuyo propietario es la empresa Microsoft.
- **UNIX**: Es un sistema operativo, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T.



El sistema informático

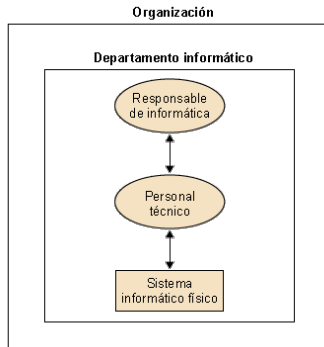
Ópticas posibles del **Sistema Informático** de una organización

- **Para la organización:** es un departamento como cualquier otro, con unos recursos disponibles para el resto de departamentos.
- **Para los informáticos:** es un conjunto de servidores, redes y ordenadores personales para “hacer cosas”.
- **Para los usuarios:** es una herramienta más que proporciona la organización para mejorar su tarea.
- **Para la dirección:** lo usan como una gran base de datos para hacer consultas que pueden ayudarles en la toma de decisiones.



El Departamento de Informática

- El Departamento de Informática (o Departamento de Tecnologías de la Información, término algo más general), se encarga de mantener y gestionar el Sistema Informático.



El Departamento de Informática

1 Sistema Informático Físico:

- *Hardware:*

- **Servidores:** múltiples servidores especializados → + control, - riesgo de fallos.
- **Ordenadores Personales (PCs):** prácticamente uno por trabajador.
- **Cableado y electrónica de red:** tanto los cables, como la electrónica de control de la red (concentradores, encaminadores, cortafuegos y conmutadores).

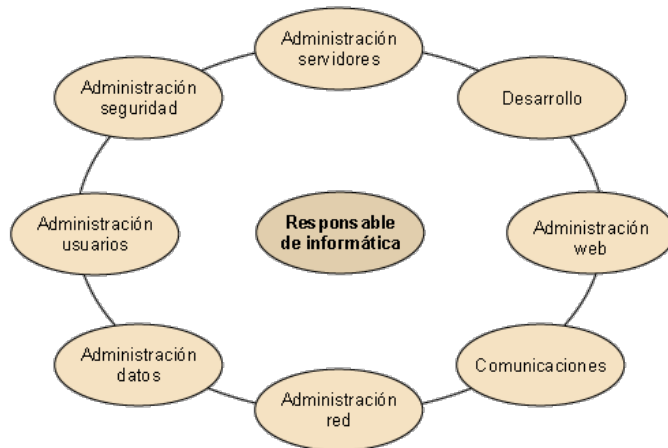
- *Software:*

- **Sistemas Operativos:** se debe hacer distinción entre el SO de los servidores y el de los PCs.
- **Software empresarial de base.**
- **Aplicaciones específicas.**

2 Personal: Responsable de Informática y Personal Técnico.



Funciones del Departamento de Informática



Funciones del Departamento de Informática

- Todas estas funciones se suelen cubrir con una o, como mucho, dos personas.

Administración de servidores

- Instalar, mantener y reparar los servidores que prestan los servicios del sistema informático de la organización.

Administración de usuarios

- Atención de las necesidades de los usuarios.
- Mantenimiento de sus equipos de sobremesa (microinformática).
- Parte más “visible” de sus funciones.



Funciones del Departamento de Informática

Administración de la red

- Responsabilidad sobre la parte física de la red.
- Asegurar que se encuentra en buen funcionamiento y que llega, de forma óptima, a todos los puntos de la organización.

Administración de los datos

- Mantener la integridad de la información de la organización.
- La información debería de estar en los servidores (aunque a veces se encuentra distribuida por todo el sistema).
- ¿Minería de datos?.



Funciones del Departamento de Informática

Administración de la web

- Mantenimiento del servidor web y, muchas veces, del contenido de la web.
- Homogeneidad global.

Administración de la seguridad

- La seguridad informática es compleja:
 - Desde la seguridad de la información existente,
 - hasta la protección física del equipamiento contra robos, incendios...
- Tarea presente en todas las funciones.



Funciones del Departamento de Informática

Desarrollo

- Una organización suele necesitar **software específico**.
- A veces, en lugar de comprarlo, se **desarrolla**.
- Esta asignatura no cubre este aspecto, en todo caso, cubriría su implantación.

Responsable de Informática

- Es el enlace entre:
 - las necesidades de la empresa,
 - y el trabajo que se lleva a cabo en el departamento.
- Decide: qué software comprar (o si se hace a medida y cómo), servidores necesarios, ordenadores y red de comunicaciones...



Funciones del Departamento de Informática

UCO

- Área de Sistemas: servidores, correo, comunicaciones...
- Área de Gestión: usuarios, licencias de *software*, web, sigma...

Interdependencia

- Cada función → tareas definidas.
- Sin embargo, **no son independientes**, sino que tienen que trabajar **coordinadas**.
- Las funciones tienen intersecciones en algunos puntos.
- Algunas tareas tienen que estar mezcladas para conseguir una gestión, un servicio y una atención al usuario en las mejores condiciones posibles.



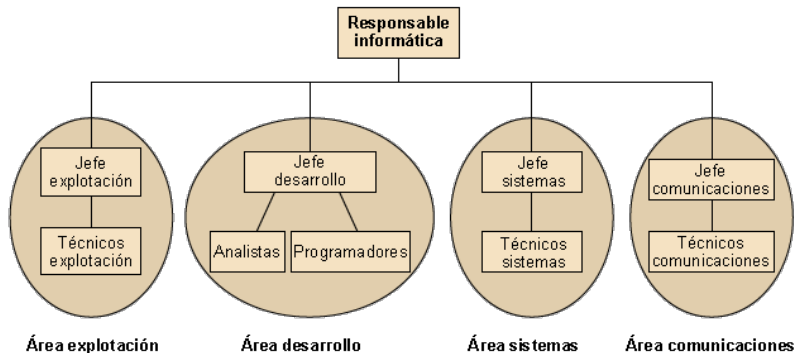
Funciones del Departamento de Informática

Ejemplo

- ¿Dar de alta a un usuario en los servidores es una tarea para el administrador de usuarios o para el administrador de servidores?
 - Si es la misma persona quien hace las dos funciones, no hay problema.
 - Si no lo es, se tiene que decidir quién se encarga:
 - En caso de que lo haga la persona que administra los servidores, se ha dividido la tarea de dar de alta a los usuarios (y todo lo que comporta) en diversas personas. Hay que valorar si vale la pena.
 - Si sólo lo hace la persona que administra a los usuarios, entonces “manipulará” los servidores para hacerlo. Hay que valorar si ello es prudente.



Organigrama del Departamento de Informática



Administrador de Sistemas

Un *Administrador de Sistemas* es la persona que tiene la responsabilidad de *implementar, configurar, mantener, monitorizar, documentar* y *asegurar* el correcto funcionamiento de un sistema informático, o algún aspecto de éste.

El *Administrador del Sistemas* tiene por objeto garantizar el tiempo de actividad (*uptime*), rendimiento, uso de recursos y la seguridad de los servidores que administra de forma *proactiva*.

Fuente: http://es.wikipedia.org/wiki/Administrador_de_sistemas



El administrador de sistemas

Administrador de sistemas

- Persona con el **poder** y la **responsabilidad** de establecer:
 - acciones,
 - procedimientos,
 - y normas,para lograr que el sistema informático sea:
 - eficiente,
 - seguro,
 - fiable,
 - y amigable.
- Cualidades:
Autoridad + responsabilidad + servicio + cooperación



El administrador de sistemas

- Dedicación del **Administrador de sistemas**:
 - **Idealmente** → Una persona encargada **solo** de la administración.
 - **Generalmente** → Comparte la labor de administración con otro tipo de trabajo.
- Labores del **Administrador de sistemas**:
 - **Idealmente** → sería uno de los miembros del **Departamento de Informática**.
 - **Generalmente** → puede hacer todo o casi todo el trabajo del **Departamento de Informática**:
 - Atender al teléfono, al fax, administrar los ordenadores...
 - Ordenar pedidos, atender a los usuarios, desarrollar *software*, reparar *hardware*, reírse de los chistes del jefe...



El administrador de sistemas

- ¿Qué se espera del **administrador**?
 - Amplios conocimientos de todo el sistema: *hardware*, *software*, datos, usuarios...
 - Capacidad reconocida para tomar decisiones.
 - Ambición y espíritu de superación.
 - Eficacia y moral irreprochables.
 - Responsabilidad: se trabaja con datos muy importantes, hay un jefe por encima...



Tareas detalladas: nivel más *hardware*

- Planificar y administrar el **entorno físico**:
 - Diseñar la habitación, especificar el sistema de refrigeración, las conexiones de energía, el control del entorno (alarma contraincendios, seguridad física...).
- Planificar los **cortes de suministro** para realizar actualizaciones o para administrar los dispositivos.
- Localizar, reparar y reemplazar componentes defectuosos (a nivel *hardware*).
- Configurar y mantener la **conectividad** entre los *hosts* (redes):
 - Monitorización.
 - Resolución de problemas.
 - Calidad de servicio.
- Instalar y mantener dispositivos del sistema, *hardware* y *drivers*. Especificar dispositivos soportados.



Tareas detalladas: mantenimiento *software* y documentación

- Mantenimiento *software*:
 - Instalación y configuración de sistemas operativos.
 - Detección de problemas en el *software* y reparación.
 - Configurar y mantener aplicaciones de negocio:
 - Aplicaciones propias (p.ej. Sigma en la UCO).
 - *e-mail*.
 - Agendas, calendarios...
- Documentación:
 - Documentar todo el sistema.
 - Mantener documentos sobre configuraciones locales y políticas locales.



Tareas detalladas: soporte a usuarios



- **Formar** a los usuarios en el manejo del *software* y en seguridad.
- Ayudar a los usuarios y proporcionar **soporte**.
- Establecer un sistema de rastreo de problemas para contestar las cuestiones de los usuarios (sistema Hermes de la UCO para notificación de incidencias).
- Asegurar que los usuarios tiene acceso a toda la **documentación**.



Tareas detalladas: servicios

- Instalar y mantener las **cuentas de usuario**, desarrollar **políticas de uso** aceptables y de nombrado de usuarios, instalar/configurar/administrar **servicio de nombres**, manejar las licencias de *software*...
- Determinar los requisitos *software*, los **parches** a instalar, los **servicios** a proporcionar y cuáles deshabilitar.
- Configurar los **servicios** de red (con sus políticas y sus requisitos de **seguridad**):
 - Impresión, ficheros compartidos, servicio de nombres...
- Instalar, configurar y administrar servidores *web*.



Tareas detalladas: seguridad

- Determinar **cuotas de disco**, políticas de **manejo del espacio** y monitorizar los ficheros de log.
- Configurar y manejar la **seguridad del sistema**:
 - Seguridad para aplicaciones de negocio.
 - Lectura de listas de correo de seguridad y de notificaciones CERT, SNORT (reglas *firewall* liberadas, pago por alertas inmediatas).
 - Instalar y configurar *firewall* para limitar el acceso de intrusos.
 - Recabar evidencias en caso de intrusión y limpiar el rastro.



Tareas detalladas: copias de seguridad

- Configurar y mantener *backups* del sistema:
 - Determinar la estrategia y las políticas de copias de seguridad.
 - Configurar el *software* de copia.
 - Realizar/automatizar copias.
 - Mantener *logs*.
 - Comprobar la integridad de las copias.
 - Determinar planes de supervivencia a catástrofes.
 - Realizar restauraciones.



El administrador de sistemas

- Estrategia del **Administrador de sistemas** al realizar una tarea:
 - 1 **Planearlo** antes de hacer los cambios, haciendo un estudio detallado de los pasos que hay que realizar.
 - 2 Hacer los **cambios reversibles**, haciendo copia de seguridad del sistema o de los ficheros de configuración a modificar.
 - 3 Realizar los **cambios de forma incremental**, probándolos si fuese posible (más fácil localizar los fallos).
 - 4 **Probarlo, probarlo, probarlo**, ..., antes de hacerlo público.
 - 5 Conocer **realmente cómo** trabajan las cosas.
- Cuando se realice cualquier modificación:
 - Precaución **antes de...**
 - Probarlo **después de...**



El administrador de sistemas

- Es una buena idea disponer de un **cuaderno de bitácora**:
 - En el se registran todos los cambios realizados sobre la configuración del sistema.
 - Sirve para uno mismo y para los demás.
- La mayoría de las veces tendremos que editar múltiples ficheros de configuración, para lo que necesitaremos un **editor de texto**.
 - `vi` (o su versión mejorada `vim`) es un editor estándar, que podremos encontrar en cualquier sistema GNU/Linux.
 - `pico` es más simple de utilizar.
 - Muchas veces solo podremos acceder al servidor por conexión `ssh`, en modo consola, por lo que no podremos utilizar editores gráficos como `gedit` o `leafpad`.



¿Qué es GNU/Linux?

• Linux:

- En agosto de 1991, el estudiante finlandés **Linus Torvals**, presenta en Internet la **versión 0.01** del kernel de un nuevo SO, inspirado en MINIX (aunque sin código de MINIX).
- Esta primera versión tenía poco más de 10.000 líneas de código.
- En 1992, Linux **se libera bajo licencia GPL**. A través de Internet, muchos programadores se unieron al proyecto.
- En 1994 Linux alcanzó la **versión 1.0**.
- En 2003, llegamos a la **versión 2.6**, con casi 6 millones de líneas de código.
- En 2011, **versión 3.0**.
- En 2015, **versión 4.0** (última 4.9.8¹)

¹<https://www.kernel.org/>



¿Qué es GNU/Linux?

- **GNU:**
 - El proyecto GNU (*GNU's Not Unix*) fue iniciado en 1983 por **Richard Stallman** bajo los auspicios de la *Free Software Foundation*².
 - Objetivo: crear un **sistema operativo completo** basado en software libre, incluyendo herramientas de desarrollo de software y aplicaciones
 - En el momento de la liberación, GNU no tenía listo su kernel:
 - Linux fue adaptado para trabajar con las aplicaciones de GNU:
Sistema GNU/Linux = Kernel Linux +
Aplicaciones GNU: compilador (gcc), librería C (glibc) y depurador (gdb), *shell* bash, GNU Emacs, GNOME, Gimp,...
 - GNU tiene ahora su propio kernel: **GNU Hurd**

²Noticia: <https://groups.google.com/forum/m/#!msg/net.unix-wizards/8twfRPM79u0/1xlg1zrWrU0Jy>



¿Qué es GNU/Linux?

- ¿Qué es GNU/Linux?



- Es un Sistema Operativo libre.
 - Surgió como reimplementación y reelaboración de UNIX.
 - Se acoge al estándar POSIX.
 - No es la única alternativa libre a UNIX (FreeBSD, OpenBSD, NetBSD...), pero si la más popular.
- Creado y mantenido por voluntarios.
- Código abierto → Se puede conocer, modificar y extender.
- Gratuito.



¿Qué es GNU/Linux?

- ¿Qué es GNU/Linux?



- GNU es un alternativa libre al Sistema Operativo UNIX, diseñado por **Richard Stallman** (1983).
- El núcleo fue diseñado por el finlandés **Linus Torvalds** (1991).
- Se unieron y formaron **GNU/Linux**.
- Controversia: http://es.wikipedia.org/wiki/Controversia_por_la_denominaci%C3%B3n_GNU/Linux



Distribuciones de GNU/Linux

- **Distribución GNU/Linux:** Colección de *software* que forma un S.O. basado en el *kernel* Linux. normalmente incluye:
 - El kernel Linux,
 - las aplicaciones GNU,
 - *software* de terceros, libre o propietario: X Windows, servidores, utilidades,...
- Las distribuciones difieren en el empaquetado de los programas (rpm, deb, tgz), el programa de instalación y las herramientas específicas:
 - Lista de distribuciones en Wikipedia: http://en.wikipedia.org/wiki/List_of_Linux_distributions
 - Información interesante en <http://www.distrowatch.com>
- Algunas de las más populares son Debian, Red Hat (Fedora), SuSE, Gentoo, Ubuntu, Guadalinex...



Distribuciones de GNU/Linux: Debian



<http://www.debian.org/>

- Distribución totalmente libre, sin fines comerciales.
- Tres ramas en la distribución:
 - Stable: destinada a entornos de producción (desde abril 2015, versión 8.0 *jessie*).
 - Testing: software más nuevo, en fase de prueba (actualmente *stretch*).
 - Unstable: en fase de desarrollo (siempre *sid*).
- Algunas características:
 - Gran número de aplicaciones disponibles.
 - Potente formato de empaquetado: paquetes deb y herramienta apt.
 - Instalación y cambio de versiones a través de red.



Distribuciones de GNU/Linux: Ubuntu



<http://www.ubuntu.com/>

- Enfocada a ordenadores de escritorio (*Desktop/laptop Computers*), aunque existe la versión para servidores.
- Basada en Debian, Ubuntu concentra su objetivo en la **usabilidad, lanzamientos regulares y facilidad en la instalación.**
- Patrocinado por *Canonical Ltd.*, una empresa privada fundada y financiada por el empresario sudafricano *Mark Shuttleworth*.
 - Última versión: Ubuntu 16.10 (*Yakkety Yak*, 13/10/2016).
 - Próxima versión: Ubuntu 17.04 (*Zesty Zapus*).
 - Última versión con soporte a largo plazo: Ubuntu 16.04 LTS.
- Proyectos: **kubuntu, edubuntu, xubuntu, lubuntu...**
- https://es.wikipedia.org/wiki/Ubuntu_%28filosof%C3%ADa%29



Distribuciones de GNU/Linux: Red Hat



<http://www.redhat.com/en>

- Gran firma comercial del mundo GNU/Linux.
- Fundada por Marc Ewing y Bob Young, en 1994.
- Introduce el formato de empaquetado `rpm` (*RedHat Package Manager*).
- Desde 2002, orientado **en exclusiva al mercado corporativo**, solamente a suscriptores de pago (RHEL):
 - Cede la última distribución **personal** (RH 9) a la comunidad \Rightarrow aparecen los proyectos Fedora/CentOS.
- Última versión: **Red Hat Enterprise Linux 7 (Maipo)** desde junio de 2014.
- Distribuciones libres que clonan RHEL: **CentOS**, Scientific Linux, ClearOS, etc.



Distribuciones de GNU/Linux: CentOS



CentOS <http://www.centos.org/>

- Objetivo: ofrecer al usuario un software de “clase empresarial” gratuito.
- Robusto, estable y fácil de instalar y utilizar.
 - Mantiene el sistema de paquetes RPM, gestionado a través de yum.
 - Última versión: CentOS 7, desde julio de 2014.
- Recomendado para **servidores**.



Distribuciones de GNU/Linux: Fedora



<https://getfedora.org/>

- Objetivo: construir un SO completo, de propósito general basado exclusivamente en código abierto
 - Mantiene el sistema de paquetes RPM, gestionado a través de yum.
 - Última versión: Fedora 25, 22 de noviembre de 2016.
- Recomendado para **equipos personales**.



¿Qué es software libre? I

- Según la *Free Software Foundation* (<http://www.gnu.org/philosophy/free-sw.es.html>) se entiende por **software libre** aquél que concede cuatro libertades a sus usuarios:
 - **Libertad 0:** La libertad de **usar el programa, con cualquier propósito**. De modo que uno puede instalarlo en cuantas computadoras desee, sin pagar licencias, por tiempo ilimitado y para cualquier tipo de uso, incluyendo comerciales.
 - **Libertad 1:** La libertad de **estudiar cómo funciona el programa, y adaptarlo a sus necesidades**. Así pues uno puede traducirlo a su idioma, corregir algún fallo o añadir alguna funcionalidad. No se depende de los intereses del productor original, sino que puede hacerlo uno mismo o encargárselo a alguien que tenga los conocimientos para hacerlo. Requiere el acceso al código.



¿Qué es software libre? II

- **Libertad 2:** La libertad de **distribuir copias**, con lo que un profesor puede dar copias del software a sus alumnos, y éstos copiárselo a sus amigos. Las copias no tienen porqué ser gratuitas, sino que es lícito cobrar por ellas. Al difundir el programa estaremos ayudando no sólo a otras personas, sino también al autor: su programa llega a más gente, y esto hace que reciba más sugerencias para mejorarlo, y más gente dispuesta a ayudarlo en el desarrollo.
- **Libertad 3:** La libertad de **mejorar el programa y hacer públicas las mejoras a los demás**, de modo que toda la comunidad se beneficie. El acceso al código fuente también es un requisito para ello. Lo más habitual es enviar las mejoras a los autores originales para que las incorporen al programa. Así llegan a todos los usuarios y no hay que volver a hacer las modificaciones en futuras versiones.



¿Qué es software libre? III

- Estas cuatro libertades son las que definen el modelo del software libre, y todas sus demás características son consecuencia de ellas.
- Para que un programa sea considerado software libre, debe facilitar todas y cada una de estas libertades.
- El software que no es libre suele designarse con los términos “software privativo” o “software propietario”. Tanto el software libre como el privativo puede ser, o no ser, **comercial**.



¿Qué no es software libre? I

- **Software de dominio público:** Se denomina software de dominio público a aquél que no tiene *copyright* (derechos de autor), porque el autor ha renunciado a ellos. En caso de que el código fuente del programa fuera también de dominio público, se trataría de un tipo de software libre. Sin embargo, la inmensa mayoría del software libre **sí tiene derechos de autor**, si bien los autores autorizan los usos descritos anteriormente.
- **Software semilibre:** El software semilibre es aquel que proporciona las mismas libertades que el software libre, siempre y cuando sea usado sin ánimo de lucro. Un ejemplo es el sistema de cifrado PGP.



¿Qué no es software libre? II

- **Freeware** o software gratuito: Existen muchos programas que se pueden descargar gratuitamente de Internet o encontrar en el CD-ROM de una revista. Estos programas son gratuitos, pero no libres. Normalmente no dan acceso al “código fuente”, por lo que no pueden modificarse, traducirse, etc. Con frecuencia también se prohíbe vender una copia a alguien dispuesto a comprarla. Por otra parte, el software libre no tiene porqué ser gratuito. El productor o distribuidor puede cobrar la cantidad que estime oportuna por proporcionar una copia. Dos ejemplos bastante extendidos de este tipo de software son Microsoft Internet Explorer y Adobe Reader.



¿Qué no es software libre? III

- *Shareware*: programas que el autor permite utilizar, pero condicionando su uso. Por ejemplo, puede exigir que se pague por él si se emplea más allá de un determinado periodo de evaluación. A veces se distribuye una versión reducida de los programas y hay que pagar para poder disfrutar del programa en toda su capacidad. Normalmente tampoco se ofrece el código fuente.

El compresor de ficheros Winzip, o el programa de chat mIRC son shareware.



¿Qué no es software libre? IV

- **Software con fuentes:** Con algunos programas se entrega el código fuente, para que el usuario pueda comprobar que no hace nada malicioso, pero eso no los hace libres, si su licencia no permite su modificación y redistribución sin condiciones excesivamente restrictivas.

Las primeras versiones de Unix, bajo licencia de AT&T, incluían el código fuente, pero quedaba prohibida su redistribución a terceras partes.



Ventajas del software libre en administración de sistemas

- Libertad de uso, estudio, redistribución y modificación:
 - podemos instalarlo en tantas máquinas como queramos.
 - podemos adaptarlo a nuestras necesidades o las del cliente.
 - podemos revisar el código y corregir errores sin esperar a que lo haga el desarrollador.
 - corrección mas rápida y eficiente de fallos, y rápida resolución de dudas y problemas, gracias al **modelo bazar**³ y a las fuertes comunidades que tiene detrás.
- **Independencia tecnológica**: no nos atamos a ningún proveedor en particular.
- Ausencia de secretismo tecnológico y de patentes (**seguridad jurídica**).
- Fiabilidad y rendimiento.

³http://es.wikipedia.org/wiki/La_catedral_y_el_bazar



Ventajas del software libre en administración de sistemas

- **Formatos estándar:** facilitan la interoperabilidad.
- Métodos simples y unificados de gestión de software (p.ej apt-get): las distribuciones evitan tener que acudir a buscar software de fuentes dudosas.
- Inmensa variedad de soluciones muy maduras: el software libre nace en entornos de servidores.
- Demanda de técnicos FLOSS en expansión, gracias a la creciente adopción por parte de las AAPP y de grandes empresas tecnológicas (Google, IBM, Sun/Oracle, etc.).
- Sistemas potencialmente más seguros: *hackers* y empresas de seguridad de todo el mundo puedan auditar los programas.
- Aspectos económicos: más de mil millones de euros en licencias de Microsoft en España anuales (2006).



Uso de software libre en administración de sistemas

- El mercado suele medirse por unidades vendidas o por beneficios.
- Difícil de evaluar para el caso del FLOSS: sistemas libres son a menudo obtenidos sin coste e instalados sin contratar soporte.
- Muchas veces se instalan en máquinas que no fueron compradas con software libre precargado.
- El método que se usa suele ser mediante acceso a máquinas públicamente accesibles (como servidores *web*).
- Problema: este método no contempla las máquinas no accesibles públicamente.



Uso de software libre en administración de sistemas

- *Software libre en servidores web.*

Source	Date	Unix, Unix-like				Microsoft Windows
		All	Linux	FreeBSD	Unknown	
W3Techs	Feb 2015	67.8%	35.9%	0.95%	30.9%	32.3%
Security Space	Feb 2014	<79.3%	N/A			>20.7%
W3Cook	May 2015	98.3%	96.6%	1.7%	0%	1.7%

Fuente:

http://en.wikipedia.org/wiki/Usage_share_of_operating_systems

- *Software libre en la administración pública.* http://es.wikipedia.org/wiki/Software_libre_en_la_Administraci%C3%B3n_P%C3%BAblica



Uso de software libre en equipos informáticos

Category	Source	Date	Linux	Unix and Unix-like	Windows	In-house	Other
Desktop, laptop (excluding Android and Chrome OS)	Net Applications ^[194]	Oct 2016	2.18% (Ubuntu, etc.)	6.43% (macOS)	91.39% (10, 8.1, 7, Vista)		
Smartphone, tablet	StatCounter Global Stats ^[195]	Nov 2016	68.31% (Android)	23.35% (iOS)	1.25% (Windows 10 Mobile, Windows Phone 8.1 and older)		9.86%
Server (web)	W3Techs ^[196]	Sep 2014	36.72% (Debian, Ubuntu, CentOS, RHEL, Gentoo)	30.18% (AIX, FreeBSD, HP-UX, macOS Server)	33.10% (W2K3, W2K8, W2K12)		
Supercomputer	TOP500 ^[189]	Jun 2016	99.79% (Custom)	0.21%			
Mainframe	Gartner ^[187]	Dec 2008	28% (SLES, RHEL)	72% (z/OS) UNIX System Services			
Gaming console, Handheld game console (7th & 8th generation only)	VGChartz ^[197]	Oct 2016		34.1% (PS4, PS3, Vita, PSP)	16.36% (Xbox One, Xbox 360)	49.54% (Wii U, Wii, 3DS, DS)	0%
Embedded	UBM Electronics ^[198]	Mar 2012	29.44% (Android plus other non-Android Linux)	4.29% (QNX)	11.65% (WCE 7)	13.5% ("Inhouse/custom" is most popular, single choice)	41.1%

Fuente: http://en.wikipedia.org/wiki/Usage_share_of_operating_systems



Desventajas del software libre

- Necesidad de una formación especializada: la gente se suele (o solía) formar con SO privativos.
- Ausencia de interfaces visuales, ya que suelen ser privativas.
- No siempre hay soporte para todo tipo de *hardware* (patentes, *drivers* y especificaciones privativas).
- ¿Mayor mercado laboral en sistemas privativos? (en otros sectores, no en administración de sistemas).



Interfaces gráficas de usuario

- Muchas distribuciones traen GUIs o herramientas visuales propias.
 - Son útiles y facilitan las tareas, sobre todo para *sysadmins* noveles.
 - Suelen ser propietarias.
 - O nos hacen dependientes de una distribución en concreto.
 - A veces poseen oscuros detalles en la forma de gestionar los recursos.
- Vamos a estudiar siempre la tecnología y métodos subyacentes, incluyendo ficheros de configuración, comandos, etc...
 - Estos suelen ser comunes a todas las distribuciones, incluso a todos los sistemas Unix.
 - La configuración manual es mejor: más rápida, más flexible, más fiable, más potente y más *scriptable*.



Inciso: ¿cómo promover el software libre?

- La **gratuidad** no es el punto fuerte del software libre.
- Insistir en la gratuidad supone minusvalorar el resto de ventajas (y es injusto para la gente que lo crea y lo mantiene).
- No se debe comenzar hablando de dinero a los que toman las decisiones.
- No hablar del FLOSS en abstracto (“Linux es mejor”): mejor estudiar costes de migración y tratar de cubrir necesidades concretas que no están cubiertas o mejorar lo que hay.
- No hay que ser impaciente: dejar que el software libre crezca con los clientes, introduciendo mejoras de forma progresiva.



El superusuario o administrador

- El **administrador** o **superusuario** es el usuario que tiene siempre todos los privilegios sobre cualquier fichero, instrucción u orden del sistema.
- En **GNU/Linux** ese usuario es **root**, que pertenece al grupo **root**:
 - Directorio HOME: /root (o / en modo monousuario).
 - Si estamos en el sistema utilizando cualquier otro usuario, ¿cómo podemos **convertirnos en administrador**?
 - Salir de la sesión y entrar utilizando **root** como nombre de usuario (deshabilitado por defecto en algunos entornos).
 - Utilizar el comando **su** → nos pedirá la contraseña de **root** y abrirá una **shell** donde tendremos **privilegios de administración**.

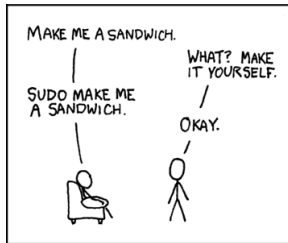


El superusuario o administrador

```
1 pagutierrez@TOSHIBA:~$ whoami
2 pagutierrez
3 pagutierrez@TOSHIBA:~$ su
4 Contraseña:
5 root@TOSHIBA:/home/pagutierrez# whoami
6 root
```



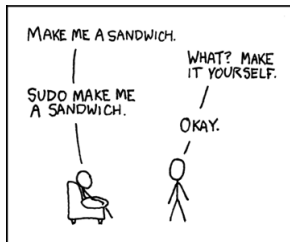
La herramienta sudo



- sudo permite a otros usuarios ejecutar órdenes como si fuesen el administrador.
 - /etc/sudoers \Rightarrow fichero de configuración
 - Fichero de solo lectura, incluso para root.
 - En él estableceremos “quién puede ejecutar qué y cómo” desde sudo.
 - En la mayoría de distribuciones basta con que el usuario pertenezca al grupo sudo.



La herramienta sudo



- visudo \Rightarrow orden para modificar el fichero de configuración /etc/sudoers.

```
1 # Especificación de privilegios de usuario
2 root    ALL=(ALL:ALL) ALL
3 # Los miembros del grupo sudo podrán ejecutar cualquier comando
4 %sudo   ALL=(ALL:ALL) ALL
```

- sudo orden \Rightarrow pide contraseña del usuario.



Comunicación con el resto de usuarios

- El **administrador** debe comunicarse con el resto de usuarios:
 - **write**: enviar un mensaje a un usuario
 - **talk**: conversar con un usuario, incluso aunque esté en otra máquina GNU/Linux.
 - **mesg [y/n]**: habilitar/deshabilitar la llegada de mensajes al terminal.
 - **wall**: mandar un mensaje a todos los usuarios del sistema.
 - Fichero **/etc/motd**: contiene el mensaje del día que se imprime justo después de entrar al sistema (en modo texto).
 - Fichero **\$HOME/.hushlogin** \Rightarrow permite evitar el mensaje del día.
 - Fichero **/etc/issue**: contiene el mensaje que se muestra antes del login, normalmente muestra la versión de Linux (en modo texto).



Referencias



Evi Nemeth, Garth Snyder, Trent R. Hein y Ben Whaley

Unix and Linux system administration handbook.

Capítulo 1. *Where to Start*.

Prentice Hall. Cuarta edición. 2010.



Aeleen Frisch.

Essential system administration.

Capítulo 1. *Introduction to System Administration*.

O'Reilly and Associates. Tercera edición. 2002.



Miguel Vidal y José Castro

El software libre y la administración de sistemas

<http://rodrigoaguilera.net/sites/rodrigoaguilera.net/files/miscelanea/sysadmin.pdf>



Programación y Administración de Sistemas

1. Introducción a la administración de sistemas

Pedro Antonio Gutiérrez

Asignatura "Programación y Administración de Sistemas"
2º Curso Grado en Ingeniería Informática
Escuela Politécnica Superior
(Universidad de Córdoba)
pagutierrez@uco.es

7 de febrero de 2017

