

Nombre:

Fecha: / 11 /2010

Grupo: 1 ☐ 2 ☐ 3 ☐ 4 ☐

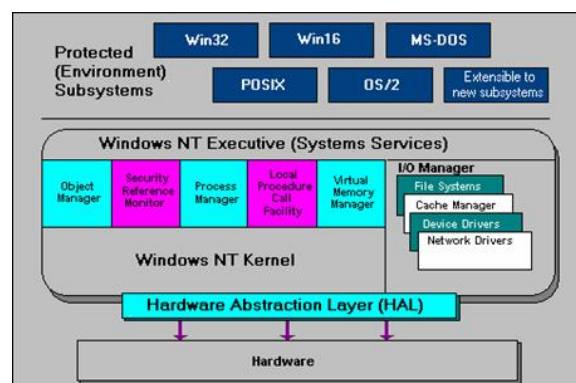
PRÁCTICA 13

INSTALACIÓN DE PROGRAMAS EN LINUX. DISTINTAS POSIBILIDADES

En esta práctica veremos diversas formas que permiten la instalación de aplicaciones o programas en Linux. Como ya dijimos en la práctica sobre instalación de Linux, cada distribución (en nuestro caso Ubuntu) trae una serie de programas y aplicaciones predeterminados (como por ejemplo, "OpenOffice" o "gedit", de los que ya hemos hecho uso), y el propio proveedor de la distribución da facilidades para instalar otro gran número de programas (quizá recuerdes también "startupmanager", que instalábamos a partir del "Gestor de paquetes Synaptic" de forma casi transparente al usuario).

Esos no son los únicos programas que podemos instalar en nuestra distribución. Hay programas que desarrollan otros proveedores de software para Linux (como por ejemplo SAGE, el reproductor de Flash o Acrobat Reader) que pueden no formar parte de nuestra distribución (Ubuntu) y cuyo proceso de instalación será necesariamente distinto.

También puede haber programas que no estén disponibles para Linux, pero que podamos instalar dentro de nuestra distribución a través de programas intermediarios, como por ejemplo "Wine". Wine es una reimplementación de las librerías (o APIS) Win16 y Win32, propias de sistemas operativos Windows NT (e incluso anteriores):



Como puedes observar en la imagen superior, Win32 y Win16 son dos librerías que se encuentran en los sistemas Windows entre las aplicaciones de usuario (aplicaciones de oficina, navegadores, clientes de correo...) y el núcleo del sistema operativo. Estas librerías permiten a los programadores de aplicaciones "olvidarse" de los detalles concretos de implementación del sistema operativo y de realizar llamadas al sistema, haciendo que sus aplicaciones interactúen únicamente con Win32 y Win16. Algunas (la mayoría) aplicaciones de usuario en Windows sólo usan llamadas a estas librerías. Wine lo que hace es reimplementar las librerías Win32 y Win16 para el kernel de Linux, permitiendo que las aplicaciones desarrolladas para Windows (MS-DOS, Windows 3.11, Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP, Windows Vista y Windows 7) se puedan ejecutar también sobre el kernel de Linux.

1. Un sistema operativo es en esencia un núcleo y una serie de programas que permiten a un usuario interactuar con el sistema. En nuestro caso, es lo que se denomina GNU/Linux: el kernel o núcleo Linux con las aplicaciones base del sistema del proyecto GNU (en rigor Linux es solamente el núcleo). La distribución

GNU/Linux Ubuntu viene formada por un sistema operativo GNU/Linux y un conjunto de programas integrados sobre el mismo:

http://es.wikipedia.org/wiki/Distribución_GNU/Linux

http://es.wikipedia.org/wiki/Anexo:Distribuciones_GNU/Linux

2. Además de las aplicaciones que conforman nuestra distribución, podemos instalar otras que aumenten la funcionalidad de nuestro sistema. Para facilitar la instalación y administración de nuevas aplicaciones existen los paquetes. Un paquete viene a ser una pieza de software que cumple una determinada funcionalidad. Así pues, un paquete puede ser un programa, una biblioteca, un conjunto de plugins, un conjunto de tipografías... Pueden ser incluso, por ejemplo, iconos. Existen diversas formas de instalar dichos paquetes en nuestro sistema. En esta práctica vamos a repasar varias alternativas, haciendo especial énfasis en las propias de nuestra distribución. En la documentación de Ubuntu puedes encontrar información sobre cómo instalar aplicaciones y sobre los paquetes:

http://doc.ubuntu-es.org/Instalar_aplicaciones

http://doc.ubuntu-es.org/Formatos_de_paquetes

3. Instalación de aplicaciones.

3.1 A partir del código fuente.

Aunque la mayoría de las aplicaciones disponibles para sistemas GNU/Linux están disponibles en formatos de paquetes preparados para su instalación, hay algunas que tenemos que compilar a partir del código fuente. La compilación depende de nuestro sistema, de la arquitectura de nuestro ordenador...por eso el código fuente de las aplicaciones suele venir acompañado de un programa de configuración que prepara el proceso adaptándolo a cada sistema. A partir de él, realizamos la compilación e instalación (por supuesto, para ello necesitamos tener previamente en nuestro sistema el compilador necesario). Tienes un ejemplo de este proceso aquí:

http://www.guia-ubuntu.org/index.php?title=A%C3%B1adir_aplicaciones (Sección 3).

La instalación a partir del código fuente no suele ser la forma más sencilla de instalar un programa. Entre otras cosas, requiere que tengamos en nuestra máquina los compiladores necesarios para el lenguaje de programación en que los códigos fuentes se encuentren (c++, perl, java). Dependiendo del tamaño de la aplicación, su proceso de compilación puede llevar incluso varios días. Por ejemplo, observa lo que se dice en el fichero "README.txt" del proceso de instalación de SAGE a partir de su código fuente (<http://sunsite.rediris.es/mirror/sagemath/src/README.txt>):

3. Extract the Sage source tarball and cd into a directory with no spaces in it. ...

To start the build, type

make

...

4. Wait about 1 hour to 14 days, depending on your computer (it took about 2 weeks to build Sage on the Google G1 Android cell phone).

Sin embargo, sí presenta dos importantes ventajas:

a) Disponemos en nuestro ordenador de los ficheros fuente del programa instalado. Esto nos permitirá poder explorarlos y tener acceso a las librerías y funciones del programa instalado (en el caso de SAGE, podremos explorar la definición de cada una de sus operaciones matemáticas...).

b) El proceso de compilación de los ficheros fuente se optimiza para el hardware y SO de que dispongamos, mejorando en algunos casos el rendimiento de la aplicación instalada.

3.2 Archivos binarios.

Son los equivalentes en Linux de los "ejecutables" de Windows. Son ficheros que no hay que descomprimir, desempaquetar... simplemente ejecutar (para ello habrá que darles los permisos necesarios, recuerda lo que veíamos en la práctica anterior sobre permisos "rwx"). Son aplicaciones ya compiladas para una arquitectura específica, que simplemente descomprimos en nuestro ordenador y están directamente preparadas para su uso.

Por ejemplo, la versión de GoogleEarth para Linux está disponible como un archivo binario. Trata de acceder a la dirección <http://earth.google.es/> y descarga el fichero "GoogleEarthLinux.bin". Para llevar a cabo su instalación, abre un intérprete de mandatos ("Aplicaciones -> Accesorios -> Terminal") y dirígete a la carpeta en la que se encuentra el mismo.

Para poder instalarlo debemos asignarle permisos de ejecución. Comprueba por medio del mandato

```
$ls -l
```

los permisos que tiene el fichero. Si no tiene permisos de ejecución en el sistema ("x"), no vamos a poder ejecutarlo (y por tanto instalarlo). Le asignamos permisos de ejecución, por medio del comando:

```
$chmod u+x GoogleEarthLinux.bin
```

Comprueba los permisos de que dispone ahora el archivo por medio de:

```
$ls -l
```

Si tiene permisos de ejecución, vamos a ejecutarlo ahora por medio de:

```
$/GoogleEarthLinux.bin
```

¿Qué sucede? ¿Se ha ejecutado el fichero? ¿Qué sucede al ejecutarse el mismo? ¿Has conseguido realizar la instalación del mismo?

En el caso de SAGE, este tipo de instalación también está disponible (<http://www.sagemath.org/doc/installation/binary.html>), y posiblemente sea la más sencilla de completar, ya que no exige dependencias ni paquetes adicionales.

Si dispones de Ubuntu 10.04, descarga el archivo comprimido correspondiente a esta distribución a tu ordenador. Una vez en el mismo, todo lo que debes hacer es abrirlo con el "Gestor de aplicaciones" propio de Ubuntu 10.04. Cuando lo abras, extráelo (por ejemplo) a tu directorio personal ("/home/tu_usuario") y una vez descomprimido, dentro de la carpeta resultante busca el fichero "sage". Dicho fichero debería abrir una terminal en la que puedes encontrar "sage".

4. Los paquetes son la agrupación instalable de los diferentes archivos necesarios para que un programa funcione. Los paquetes no son ejecutables, sino que son

gestionados por terceras aplicaciones. Además, los paquetes de Linux no traen consigo las librerías compartidas (dependencias) y, llegado el caso de necesitarse, serán instaladas o se nos informará de la necesidad de hacerlo, según el gestor de paquetes que estemos utilizando. Es decir, el programa "A" depende del programa "B", y éste a su vez de "C" (cada uno a su vez puede depender de una o más librerías). Así que si queremos instalar "A", tendremos que instalar todos los demás. Esto puede convertirse en una tarea ardua y complicada. Comandos como "rpm" o "dpkg" permiten instalar paquetes como "A" individualmente, pero no resuelven las dependencias. Para hacer esto se crearon los sistemas de instalación de paquetes, lo que permite hacer más simples las cosas. Las aplicaciones Linux se suministran normalmente en dos formatos o tipos de paquetes, y son dependientes de la distribución:

RPM: se utilizan en distribuciones basadas en Red Hat, Fedora, Mandriva... pueden tener tanto binarios como código fuente.

DEB: distribuciones basadas en Debian (Ubuntu, Linex...). Pueden contener tanto binarios como código fuente.

La mayor parte de las aplicaciones disponen de versiones DEB y RPM. Se pueden convertir paquetes y pasarlos de rpm a deb y viceversa, si se diera el caso que no existieran paquetes en alguno de los formatos, con el programa ALIEN (<http://kitenet.net/~joey/code/alien/>), pero éste no siempre podrá garantizar el correcto funcionamiento de un paquete convertido a otro. En general, se garantizará mejor la instalación de un paquete si éste fue creado directamente para su tipo correspondiente.

4.1 DEB: Es el formato de paquetes nativo de Debian, y por tanto de Ubuntu. Para instalar estos paquetes sólo tienes que hacer doble click sobre el fichero en el navegador Nautilus (el navegador por defecto usado en Ubuntu) y automáticamente se lanzará la aplicación gdebi, que se ocupará de instalar el paquete y buscar las dependencias de otros paquetes que pudiera necesitar para su correcta instalación.

Si lo preferimos, también se pueden instalar mediante la línea de comandos, mediante el comando dpkg, que no instala automáticamente las dependencias sino que se limita a indicarnos durante el proceso de instalación.

Algunas opciones de dpkg:

dpkg -I: Comprueba los paquetes instalados en la máquina y ofrece un listado completo.

dpkg -L NombrePaquete: Informa sobre el contenido (los ficheros) que forman un paquete.

dpkg -i NombrePaqueteCompleto: Para instalar paquetes que tenemos localmente y no necesitamos descargar.

dpkg -r NombrePaqueteCompleto: Desinstalación.

dpkg -s fichero: Informa de los paquetes que contienen ese fichero (a qué paquetes pertenece).

Aquí tienes un pequeño HOWTO para usar dpkg: <http://www.alcancellibre.org/staticpages/index.php/como-dpkg/print>

Comprueba los paquetes instalados en tu máquina. Si no eres capaz de verlos en una sola pantalla, utiliza una de las formas que aprendimos en la práctica anterior

(uso de "less" con una tubería, volcado a un fichero externo "paquetes_ubuntu_804" con ">>").

En el intérprete de mandatos, visualiza el contenido de top. ¿Qué hace el programa? ¿Qué tipo de información nos muestra?

Obtén el archivo .deb adecuado en <http://packages.debian.org/lenny/htop> (observa que tu plataforma es una i386) y procede a instalar htop en tu ordenador. Recuerda que lo puedes hacer desde el navegador Nautilus, sólo con ejecutar el fichero ".deb". También lo puedes hacer desde el intérprete de mandatos por medio del uso de "gdebi" (para lo cual tendrás que ejecutar la acción como súper-usuario, es decir, deberás añadir "sudo" antes del comando apropiado). Si el paquete no tuviera dependencias que resolver, también lo podrías instalar por medio del mandato:

```
sudo dpkg -i htop....deb
```

Comprueba si lo anterior funciona en este caso.

Ejecuta ahora desde la línea de mandatos "htop". ¿Qué realiza este programa? ¿Cuál es su diferencia con "top"?

Veamos ahora un segundo ejemplo de instalación de un programa a través de su paquete ".deb". Entra en el sitio web <http://www.google.com/chrome?hl=es&platform=linux>. Descarga el paquete ".deb" correspondiente al navegador "Google Chrome". Guárdalo en tu ordenador e instálalo desde el intérprete de mandatos (sudo gdebi ...). ¿Has encontrado algún problema en la instalación? Si es así, trata de documentar los motivos del mismo. En caso de que hayas podido instalar la aplicación, ejecuta el navegador desde el menú "Aplicaciones -> Internet" y comprueba que el mismo ha sido bien instalado.

Trata de instalar ahora Skype en tu ordenador a partir de su paquete *.deb. Puedes encontrar un enlace al mismo en <http://www.skype.com/intl/en-us/get-skype/on-your-computer/linux/>.

4.2 RPM (RPM package manager): Es un sistema de empaquetamiento que se está convirtiendo en un estándar de hecho en el mundo Linux por las ventajas que supone sobre otros modelos. RPM proporciona al usuario una serie de facilidades que hacen el mantenimiento del sistema más sencillo de gestionar, ya que mantiene una base de datos de los paquetes instalados y de sus archivos, lo que permite realizar consultas y verificaciones del sistema. Asimismo, otra ventaja es que al actualizar software, los archivos de configuración se respetan de manera que no sea necesario volver a realizar los ajustes específicos que ya tuviera definidos, o si no fuera posible, realiza una copia de seguridad de los mismos. En la siguiente guía podrás encontrar información detallada sobre los paquetes rpm: <http://es.tldp.org/COMO-INSFLUG/COMOs/RPM-Como/>.

5. Instalación de paquetes usando programas propios de nuestra distribución (Ubuntu).

5.1 Desde la consola: "apt-get" y "aptitude".

Existen dos formas de instalar programas en modo texto (desde el intérprete de mandatos), por medio de las aplicaciones aptitude y apt-get. Ambos programas son muy similares, salvo en un detalle: aptitude recuerda las dependencias que se han aplicado en la instalación de un paquete. Esto significa que si se instala o actualiza una aplicación con aptitude y luego se quiere desinstalar, aptitude borrará el programa junto con todas sus dependencias (excepto si son usadas por otros

paquetes). Si se instala con apt-get, la desinstalación borrará sólo el paquete especificado, pero no las dependencias.

Uso

Instalar paquetes:

```
$sudo aptitude install <paquetes>  
$sudo apt-get install <paquetes>
```

Desinstalar paquetes:

```
$sudo aptitude remove <paquetes>  
$sudo apt-get remove <paquetes>
```

Actualizar la lista de paquetes disponibles:

```
$sudo aptitude update  
$sudo apt-get update
```

Actualizar el sistema con las actualizaciones de paquetes disponibles:

```
$sudo aptitude upgrade  
$sudo apt-get upgrade
```

Obtener una lista de opciones del comando:

```
$ sudo aptitude help
```

Comprueba si el programa "gimp" está instalado en tu máquina:

```
$which gimp
```

Si está instalado, desinstálalo por medio de alguno de los comandos anteriores. Tras la desinstalación, o en caso de que no estuviera instalado, instala "gimp" usando aptitude, por medio de los mandatos y opciones que acabamos de introducir. ¿Se ha encargado el programa de resolver las dependencias? Ejecuta el programa desde el menú "Aplicaciones -> Gráficos". ¿Para qué sirve el programa?

Aptitude también tiene su interfaz en el intérprete de mandatos. Lo puedes comprobar ejecutando "aptitude" sin argumentos ni opciones. Aquí tienes un tutorial sobre apt-get: <http://doc.ubuntu-es.org/Apt>.

5.2 En modo gráfico: Synaptic.

Apt tiene su interfaz gráfica en el sistema Ubuntu. En <http://doc.ubuntu-es.org/Synaptic> puedes encontrar una guía de cómo utilizar las capacidades de "apt" desde la interfaz gráfica. Puedes encontrar el gestor de paquetes Synaptic en "Sistema -> Administración -> Gestor de paquetes Synaptic". Dentro del gestor de paquetes, busca el paquete "Wine" e instálalo (siempre y cuando no esté instalado ya). Observa cómo "Synaptic" se encarga de definir e instalar las dependencias entre paquetes. ¿Para qué sirve "Wine"?

6. Los Repositorios

Existen listas de paquetes disponibles para instalar; son los llamados Repositorios, que es de donde "aptitude" y "Synaptic" toman los paquetes que instalan en el ordenador. Los repositorios dependen de cada distribución (¿podrías decir por qué

han de ser dependientes de la distribución?), y pueden crearse repositorios personales.

Aquí hay una guía sobre repositorios de Ubuntu: <http://doc.ubuntu-es.org/Repositorios>.

Es importante que sepas que el repositorio Ubuntu consta de cuatro repositorios diferentes (cuatro componentes del repositorio Ubuntu), dependiendo de qué aplicaciones contienen. Son los llamados Main, Restricted, Universe y Multiverse. Lee la siguiente referencia y explica en tu informe de prácticas qué tipo de software podemos encontrar en cada uno de ellos.

http://doc.ubuntu-es.org/Componentes_de_los_repositorios

¿Dónde se pueden encontrar drivers para una tarjeta ATI?, ¿dónde está g++, gcc...? Cuando "aptitude" o "Synaptic" quieren instalar un paquete u obtener las dependencias necesarias para hacerlo, buscan en los repositorios que están en la lista el lugar donde está ese paquete; para ello tienes que tener configurados los repositorios en tu sistema. Puedes encontrar una lista de repositorios en el fichero "/etc/apt/sources.list". Ábrelo desde el intérprete de mandatos con el programa "gedit" ejecutándolo como "súperusuario":

```
$sudo gedit ruta_del_fichero
```

Comentando o "descomentando" (añadiendo o eliminando "#") las líneas correspondientes puedes activar o desactivar los repositorios. También puedes añadir otros repositorios "externos" a Ubuntu o propios, añadiendo líneas a este fichero. Aquí tienes un enlace que explica este punto:

http://www.guia-ubuntu.org/index.php?title=Activar_repositorios_externos

Synaptic proporciona una herramienta gráfica para manejar los repositorios. Ve a "Configuración>Repositorios" dentro del gestor de paquetes Synaptic y observa qué tienes configurado.

7. Vamos a ver ahora cómo podemos instalar aplicaciones propias de Windows gracias a la herramienta Wine. Esta opción puede ser muy útil, especialmente para usuarios que empiezan con Linux y que quieren utilizar algunas de sus herramientas favoritas de Windows. Una vez te acostumbras a trabajar con Linux, seguro que encuentras herramientas que pueden realizar la función deseada.

Entra en la página web <http://www.spotify.com/es/download/other/>. Lee las instrucciones y las distintas plataformas en que la aplicación está disponible. Vamos tratar de instalar la versión de "Spotify" para Windows sobre "wine", y ver si funciona en Ubuntu. Como ya hemos instalado "wine" desde Synaptic, puedes saltar directamente al paso en el que se pide ejecutar "winecfg" y comprobar si tu tarjeta de audio está bien configurada.

Posteriormente vamos a la página de descarga de Spotify para Windows (<http://www.spotify.com/es/download/windows/>) y seguimos los pasos de la instalación. El proceso de instalación se debería completar de forma correcta. Como puedes observar, para poder ejecutar "Spotify" necesitas crear un usuario y una contraseña. Si quieres hacerlo, puedes darte de alta en la aplicación en <http://www.spotify.com/es/new-user/>.

8. Vamos a instalar en Ubuntu un IDE para C++. Entra de nuevo en el "Gestor de paquetes Synaptic" de Ubuntu, y busca en IDE para C++ entre todas las aplicaciones que se pueden encontrar (Eclipse, Anjuta...). Instala el mismo en tu equipo y comprueba su funcionamiento. ¿Qué entorno has utilizado?

9. Busca algunas de las aplicaciones que utilizas más a menudo (notepad++, messenger, Devcpp, Flash) y trata de instalarlas en Ubuntu de alguna de las formas anteriores que hemos visto (sobre Wine, con un "*.deb", desde un binario...). Comenta el resultado y las dificultades halladas.

10. Enlaza desde tu página de inicio el informe de la práctica, así como el fichero "paquetes_ubuntu_804".