



MANUAL ESCENCIAL GNU/LINUX - UBUNTU



Marco Antonio Caballero Moreno

ÍNDICE

1 1. INTRODUCCION	3
1.1 Que es el Software Libre?.....	4
1.2 Inicios del Software Libre.....	5
1.3 GNU/LINUX.....	5
1.4 Línea de tiempo del desarrollo de distribuciones Linux.....	6
1.5 Ubuntu	7
2 2. VIRTUAL BOX.....	8
2.1 Instalación de Virtual Box	8
2.2 Instalación de Ubuntu en Virtual Box	11
3 3. GENERALIDADES DE GNU/LINUX.....	17
3.1 ¿Qué es GNU?.....	17
3.2 ¿Qué es el software libre?.....	17
3.3 ¿Qué es el GNU/LINUX?.....	17
3.4 ¿QUÉ ES UNA DISTRIBUCIÓN DE LINUX?.....	19
3.5 UBUNTU	20
4 4. SISTEMA DE FICHEROS	21
4.1 La jerarquía del sistema de ficheros.....	21
4.2 ESTRUCTURA DE DIRECTORIOS.....	22
5 5 OPERACIONES BÁSICAS EN LA TERMINAL	23
5.1 TERMINAL O CONSOLA	23
5.2 VIZUALIZACIÓN, CREACIÓN Y CAMBIO DE DIRECTORIO (pwd, ls, cd, mkdir)24	
5.3 VISUALIZACIÓN DE FICHEROS (cat, more, less, head, tail).....	28
5.4 EDICIÓN DE FICHEROS (touch, vi, ee, mcedit).....	30
5.5 COPIA Y BORRADO DE FICHEROS (cp, mv, rm).....	32
5.6 COPIA Y BORRADO DE DIRECTORIOS (cp, mv, rm).....	34
5.7 INSTALACION DE PAQUETES	35
5.8 DESINSTALACION DE PAQUETES.....	36
6 ¿POR QUÉ EXISTEN GRUPOS, USUARIOS Y PERMISOS?	36
7 ¿QUÉ ES EL SUPERUSUARIO?	37
8. PERMISOS.....	37
9 ¿QUIÉNES SOMOS? (WHOAMI, GROUPS).....	38

10. GESTIÓN DE GRUPOS (GROUPADD, GROUPDEL, GROUPMOD)	39
11. GESTIÓN DE USUARIOS (ADDUSER, USERDEL, USERMOD).....	40
12 CAMBIO DE GRUPO Y DE DUEÑO (CHOWN, CHGRP)	41
13 CAMBIO DE PRIVILEGIOS (CHMOD)	42
14. GESTIÓN DE SERVICIOS EN UBUNTU	44

1. INTRODUCCION

GNU/Linux es a día de hoy el sistema operativo libre por excelencia. La versatilidad y soporte a nivel mundial está dado por las mismas personas que lo usan, tanto programadores y empresas grandes como por ejemplo IBM, que contribuye activamente con su desarrollo “GNU/Linux es de todos”. Debemos agradecer todo esto al movimiento del software libre GNU que Richard Stallman Encabezo a partir de los años 80’s. Agradecer también a el genio Linus Trovals por contribuir con el desarrollo del “kernel” (núcleo) de LINUX y que además lo dio gratis a toda la comunidad mundial sin pedir nada a cambio más solo reconocer su trabajo, eso es digno de elogio. Sobre estas historias se sienta las bases de nuestro hoy tecnológico.

Este SO es y sigue siendo un gran empuje a muchas de la empresa de tecnología que hoy conocemos, como Google, Facebook, Amazon, Uber entre las otras muchas. ¿Por qué lo usan? La respuesta es simple, porque es libre y es de todos, ya que no se tiene que pagar licencias de uso, cosa que ocurre con otros sistemas operativos. En nuestro ámbito Local, Startups peruanas hacen uso de este sistema operativo para correr sus servicios WEB casi de forma religiosa. Casos como Joinnus, Chazqui, Urbaner, Culqi y muchas más lo usan.

GNU/Linux no es solo un sistema operativo, sino está compuesto de muchos otros derivados llamados Distros o Distribuciones, La más popular y preferida por todos es por excelencia Ubuntu Server , con una gran cuota en el mercado de los servidores web de casi el 70%.

La Realidad a día de hoy es que la tendencia de seguir utilizando Linux va en aumento y se hace necesario aprender más acerca de este movimiento para poder sacarle todo el jugo, y aprovecharlo en nuestros emprendimientos de tecnología.

1.1 Que es el Software Libre?

- Es aquel que puede utilizar y distribuir libremente. También puede ser modificado y vuelto a distribuir.
- Para que un programa sea considerado de software libre, éste debe cumplir cuatro libertades: libertad de uso, libertad de distribución, libertad de modificación y libertad de distribución de versiones modificadas. Estos suelen estar licenciados bajo la GPL



Para que un programa sea considerado software libre, debe cumplir 4 Libertades:

- Libertad de Uso.
- Libertad de distribución.
- Libertad de modificación.
- Libertad de distribución de versiones modificadas

LIBRE NO SIGNIFICA GRATIS!

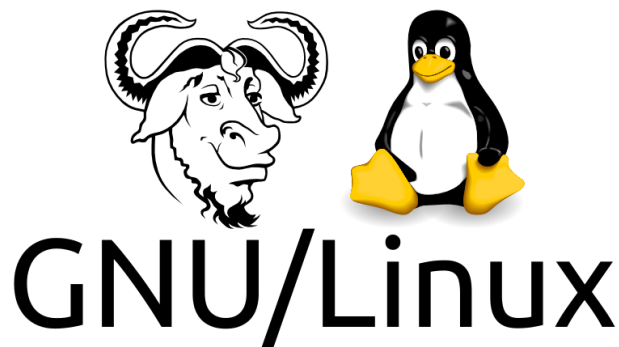


1.2 Inicios del Software Libre

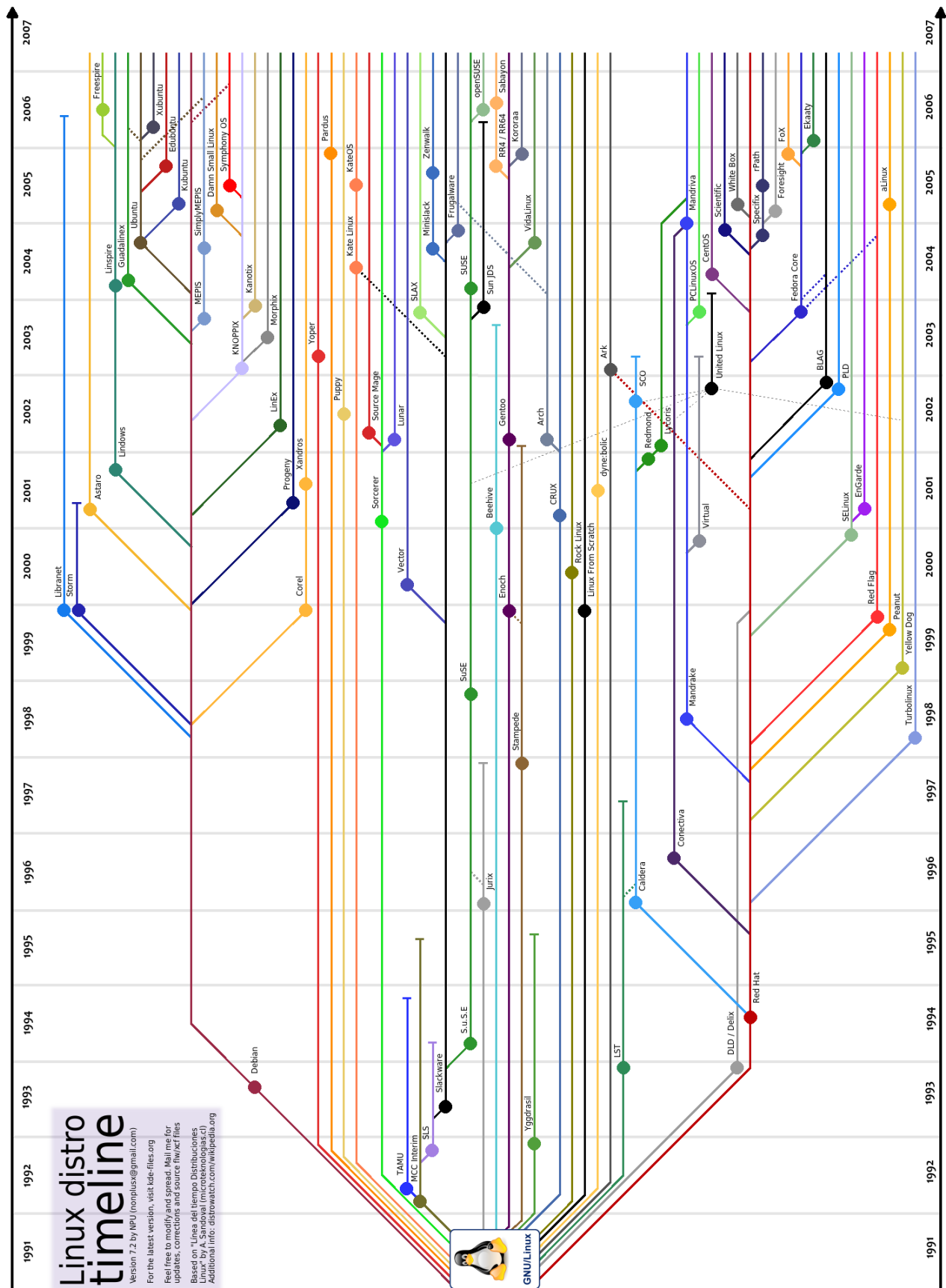
- Richard Stallman(1983- Dunda GNU Ver 7)
- Linus Trovals(1991-Crea Linux)
- 1992 – GNU/Linux (Ver 7)
- GNU: Proyecto Iniciado por Stallman, objetivo es crear un sistema totalmente libre, Poya la licincia GPL.
- Kernel Linux Es liberado en 1991 con licencia GPL

1.3 GNU/LINUX

- Linux (GNU/LINUX). es el resultado de la contribución de un gran número de compañías y grupos de personas. De hecho, el *sistema GNU/Linux* es un componente central, el cual se transforma en muchos productos diferentes: las llamadas **distribuciones**.
- Las distribuciones cambian la apariencia y funcionamiento de Linux completamente. Las hay desde grandes sistemas completos totalmente equipados (respaldadas por compañías) hasta las más ligeras que entran en un llavero USB o funcionan en computadores viejos (usualmente desarrolladas por voluntarios).
- Cuando se obtiene una distribución de GNU/Linux, también **se obtiene la libertad para estudiarla, copiarla, cambiarla, y redistribuirla. Eso es lo que la hace realmente libre.**



1.4 Línea de tiempo del desarrollo de distribuciones Linux



1.5 Ubuntu

Ubuntu: es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores. Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad de uso, los lanzamientos regulares (cada 6 meses) y la facilidad en la instalación. Ubuntu es patrocinado por Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

El nombre de la distribución proviene del concepto zulú y xhosa de ubuntu, que significa “humanidad hacia otros” o “yo soy porque nosotros somos”. El sudafricano Mark Shuttleworth, mecenas del proyecto, se encontraba muy familiarizado con la corriente. Tras ver similitudes entre los ideales de los proyectos GNU, Debian y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de Ubuntu. El eslogan de Ubuntu – “Linux para seres humanos” (en inglés “Linux for Human Beings”) – resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.

Ubuntu Está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto. Estadísticas web sugieren que la cuota de mercado de Ubuntu dentro de las distribuciones Linux es, aproximadamente, del 49 %,34 y con una tendencia a aumentar como servidor web.

Las versiones actuales de Ubuntu LTS (Soporte técnico extendido) incluyen a las versiones 16.04 LTS y 18.04 LTS. Cada una de estas versiones cuenta con una versión de escritorio “desktop” que contiene interfaz gráfica para el usuario del día a día , y la otra es la versión para servidores “Server” que no incluye interfaz gráfica. Y optimizada para trabajo en red



2. VIRTUAL BOX



Virtual Box es un software diseñado para poder ejecutar sistemas operativos huéspedes, haciendo uso de la virtualización de hardware (Máquina Virtual) dentro de otros sistema operativos (en este caso desde Windows podemos ejecutar Virtual Box y en el correr nuestro sistema operativo Linux)

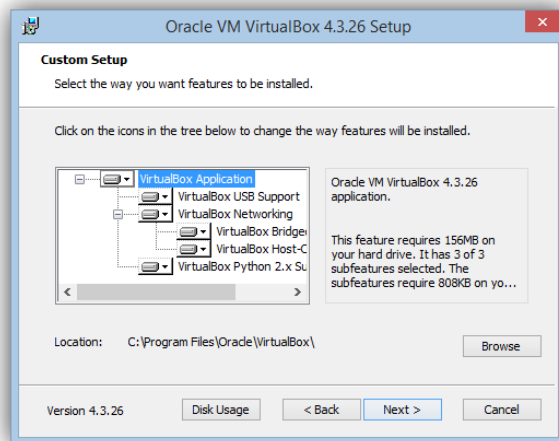
Virtual Box es Open Source, por lo que no hay que pagar alguna licencia

2.1 Instalación de Virtual Box

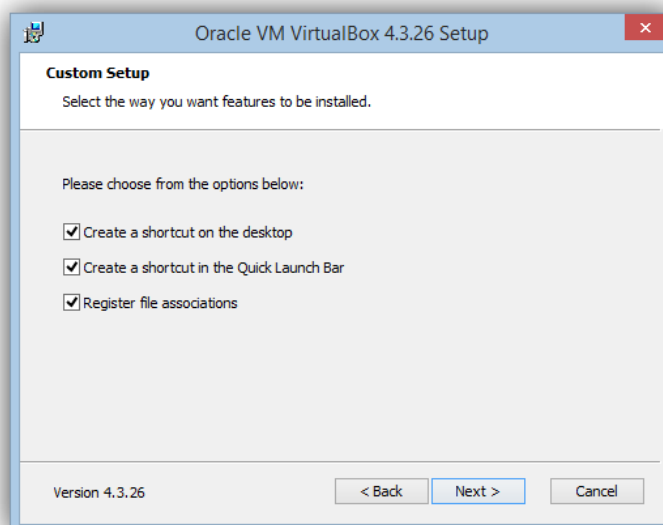
En primer lugar instalaremos VirtualBox en nuestro sistema que, en este caso, corresponde a Windows. Ejecutaremos el lanzador del programa y se nos abrirá la primera pantalla tal y como aparece a continuación.



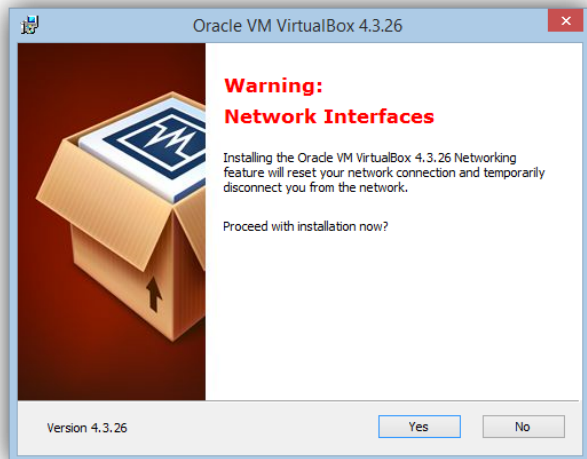
Nos da la bienvenida y nos informa de que si continuamos se instalara virtualbox en nuestro sistema, pulsamos en “next” y en la siguiente pantalla deberemos elegir los componentes y la ruta donde lo instaremos.



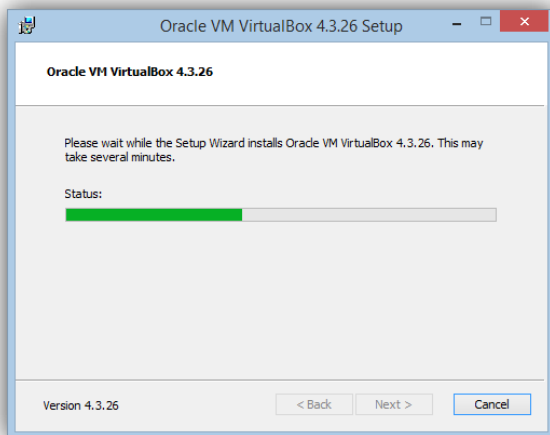
También los accesos directos que queremos configurar.



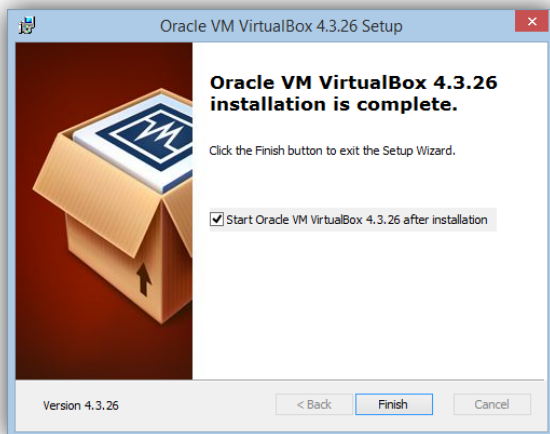
En esta pantalla se nos advertirá de que se van a copiar los archivos, se reiniciarán las tarjetas de red y se instalará el programa pulsamos en install y esperaremos el final de la instalación.



Una ventana como esta nos dirá como va la instalación así que tendremos que esperar a que finalice.

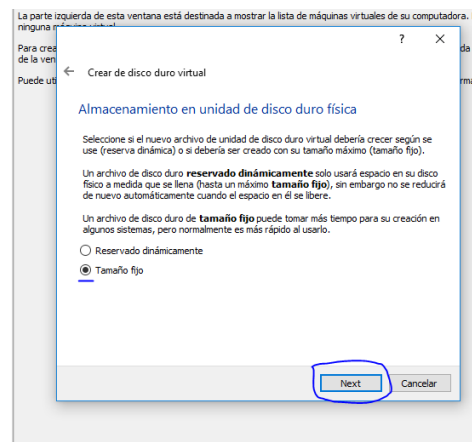
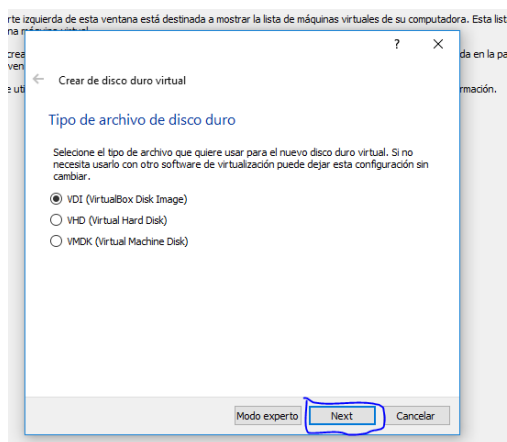
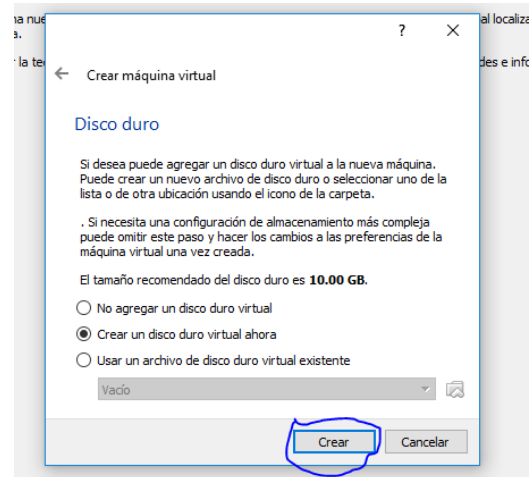
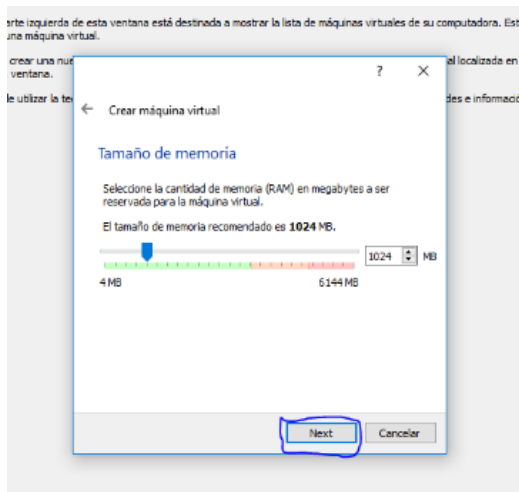
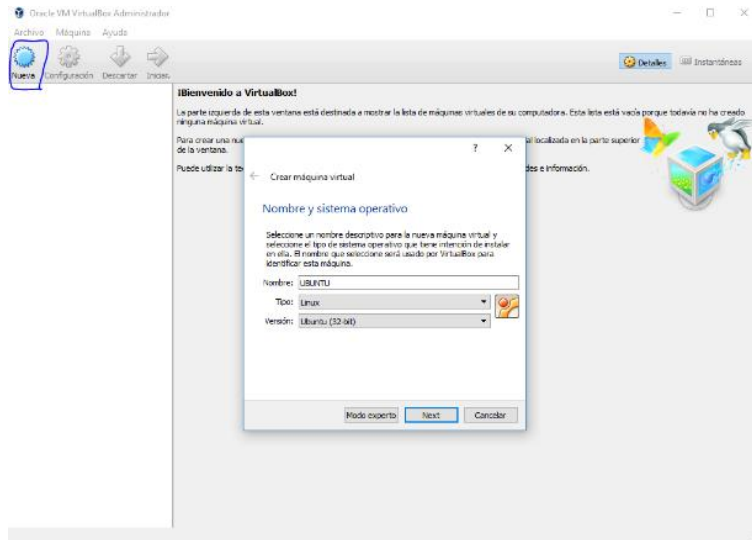


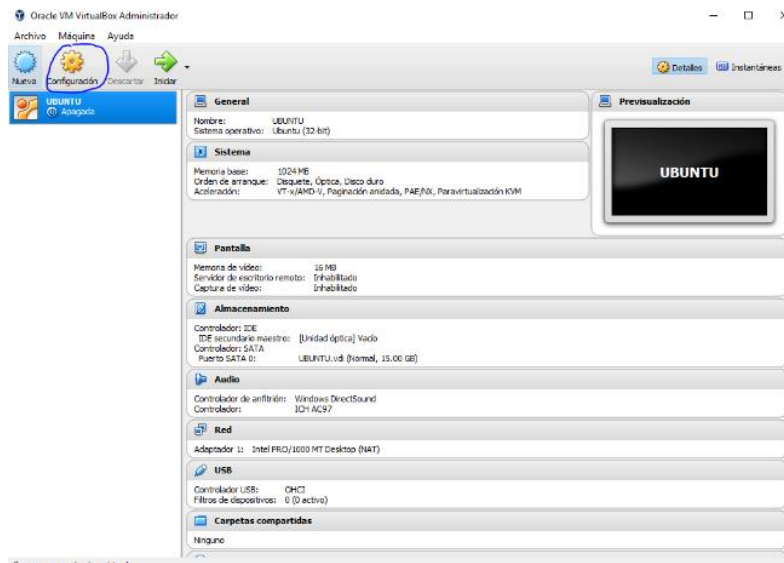
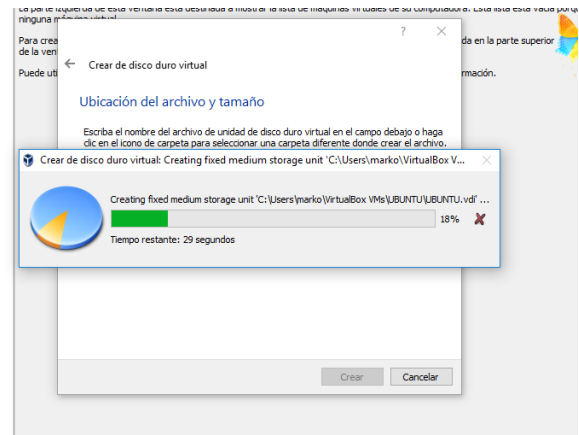
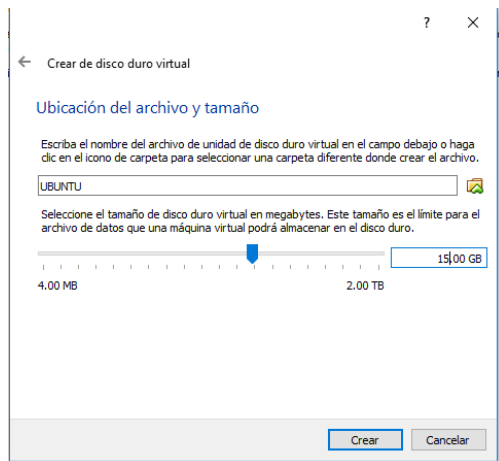
Esta es la ventana que nos avisará que el programa se ha instalado correctamente en nuestro sistema y que ya está listo para ser usado.



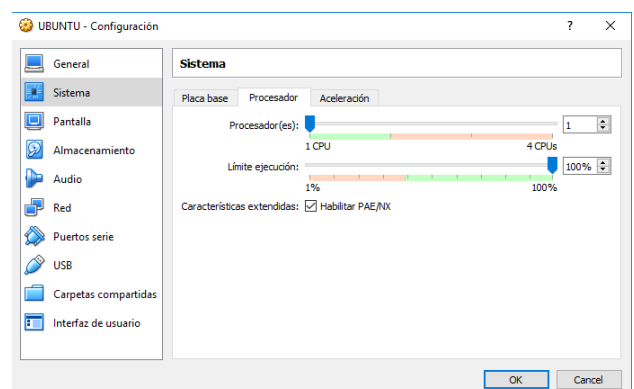
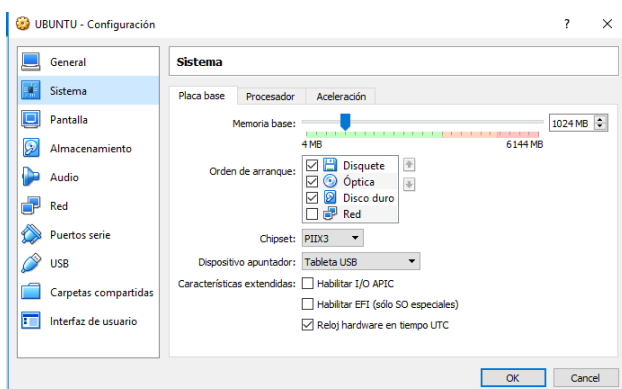
2.2 Instalación de Ubuntu en Virtual Box

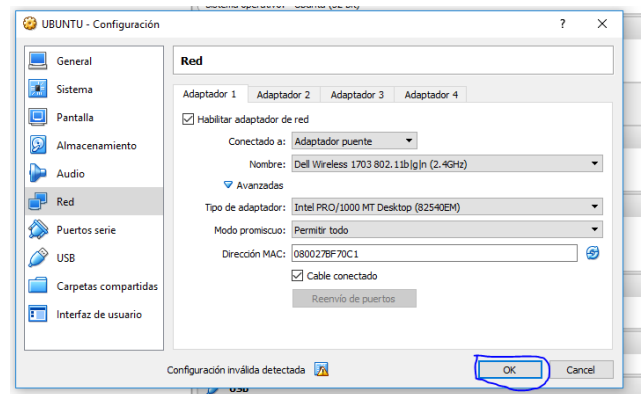
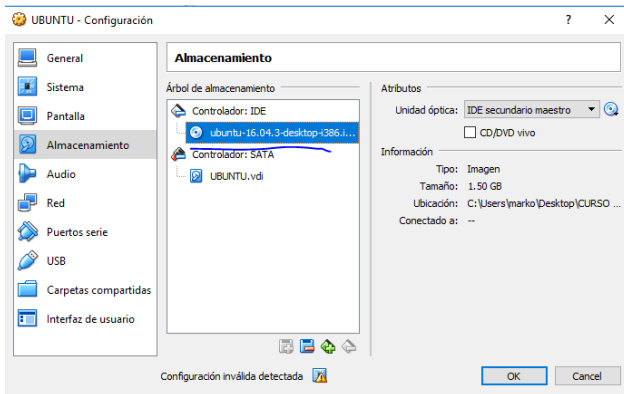
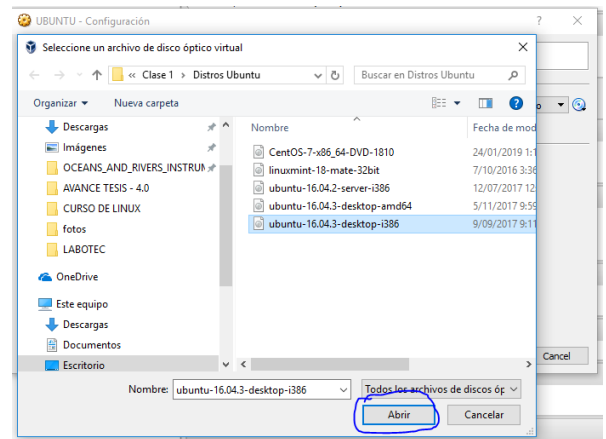
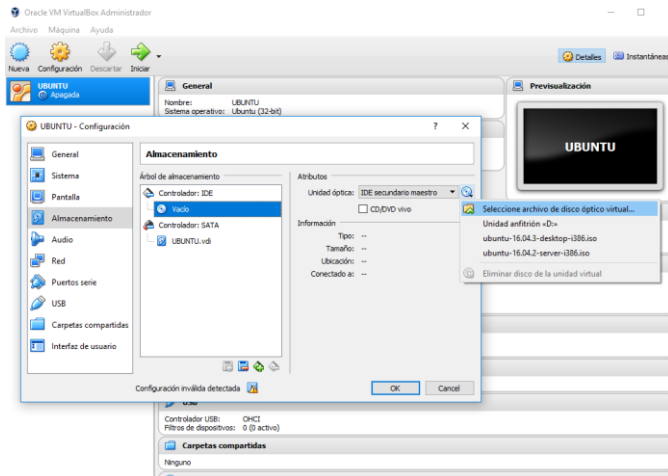
Clic en Icono NUEVA



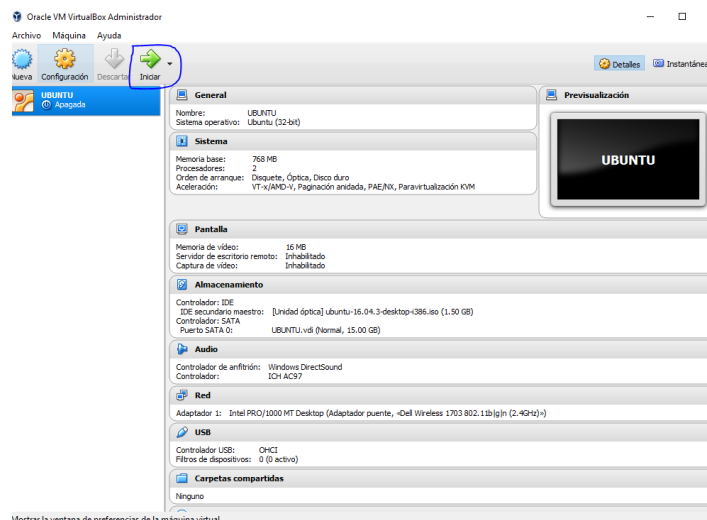


Una Vez Creada la máquina virtual, procederemos a configurarla

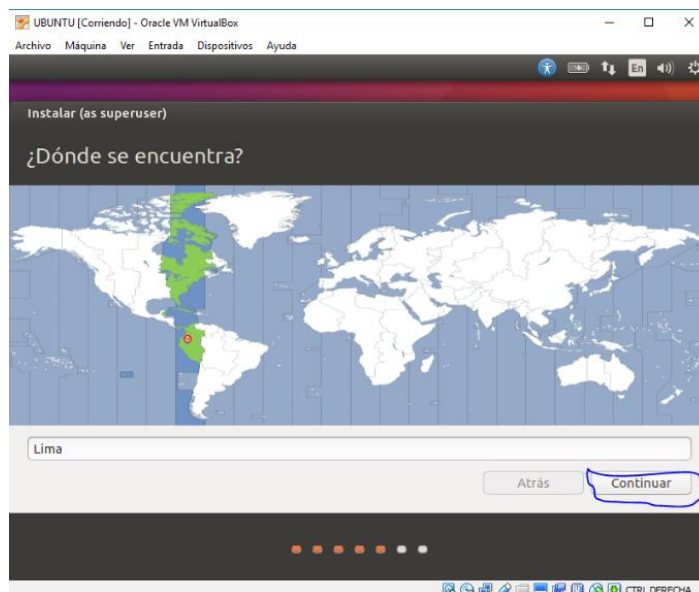
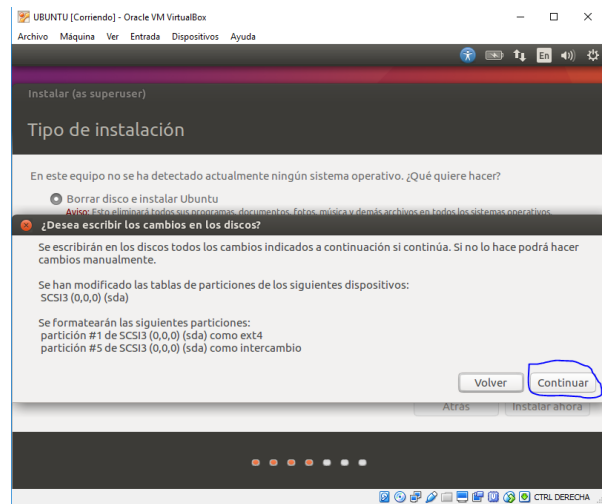
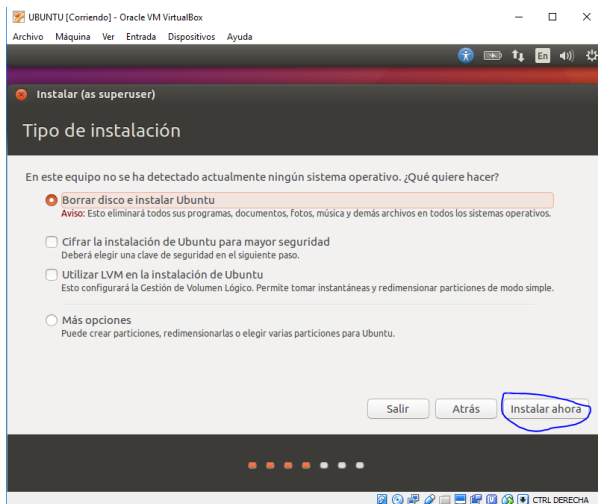
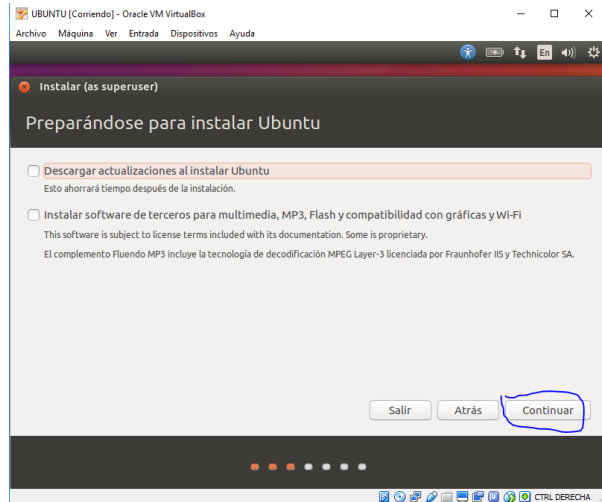
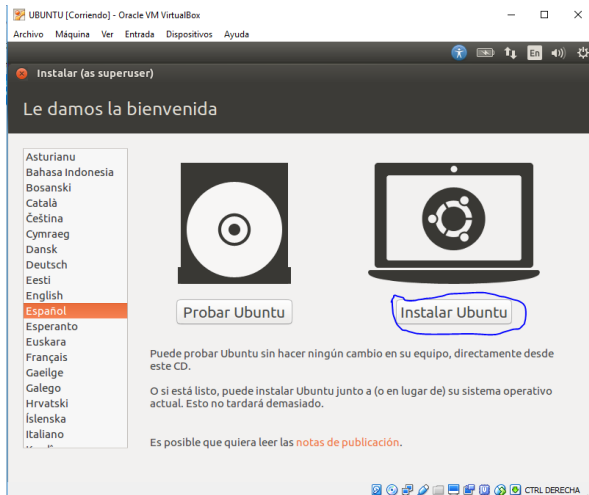


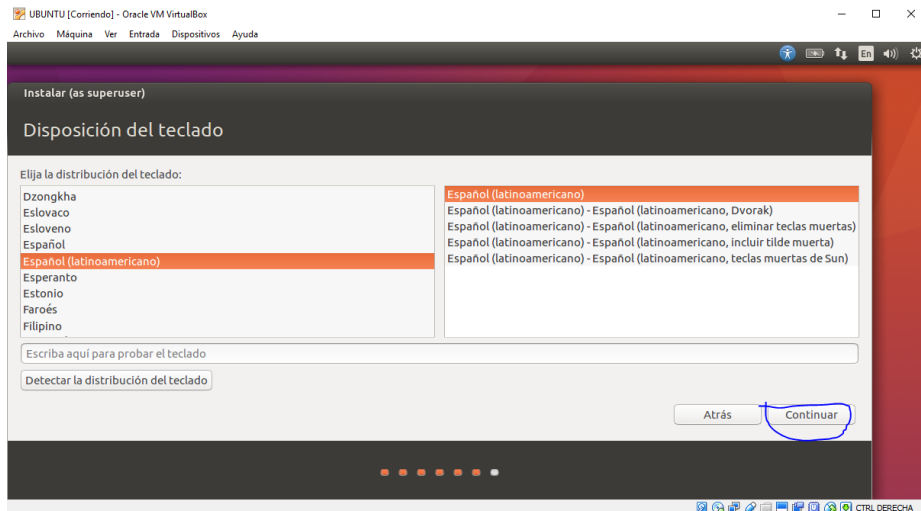


Luego de haber configurado la máquina virtual, La encenderemos para empezar con el proceso de instalación del sistema operativo Ubuntu Desktop.

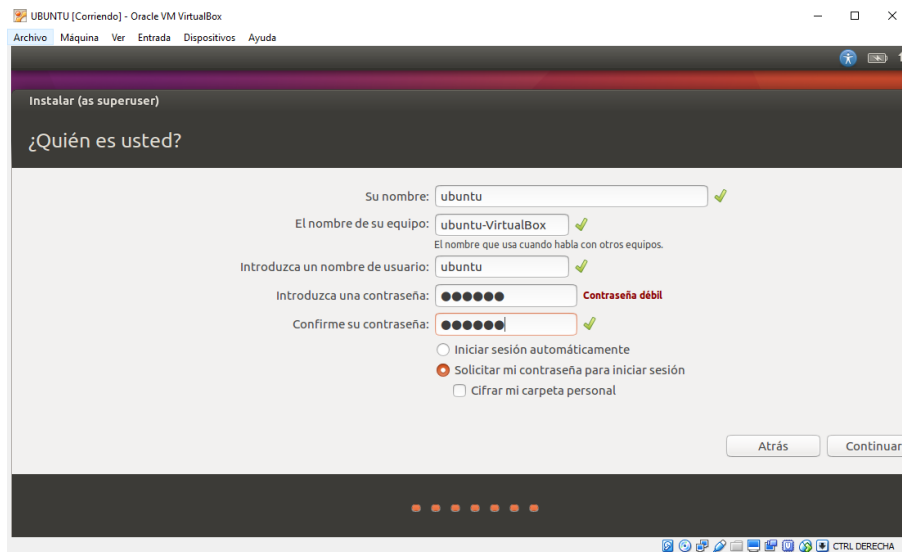


Manual esencial GNU/LINUX - Ubuntu

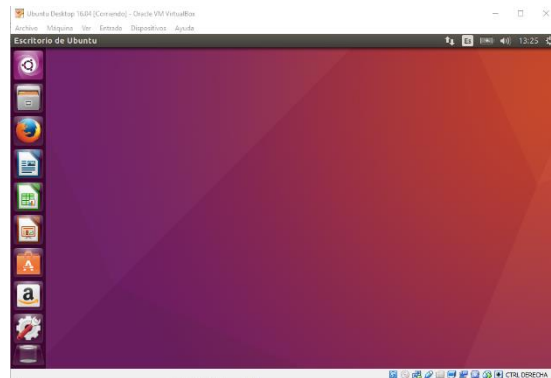
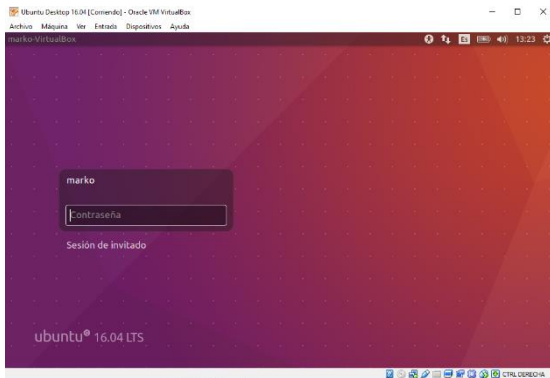
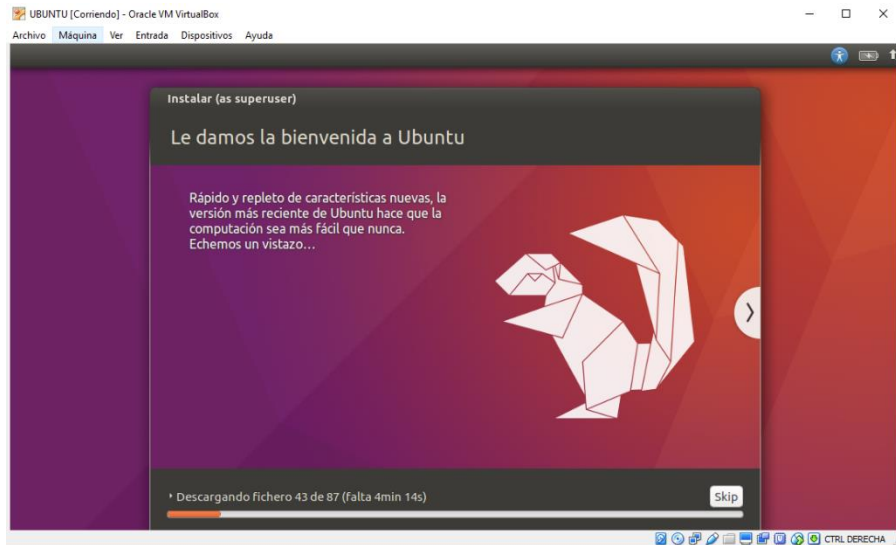




Para el inicio del sistema Ubuntu necesita crear un usuario por defecto, para eso tendremos que colocar el nombre del usuario y la contraseña, estos datos servirán posteriormente para ingresar al sistema operativo en modo supe usuario.



Esperar a que el proceso de instalación termine, el tiempo estimado dependerá del tipo de procesador que posea su computadora.



Al terminar le pedirá que reinicie la máquina virtual.

***NOTA:** Si ocurre algún error inesperado al momento de la instalación contacte de inmediato con el instructor de inmediato.

3. GENERALIDADES DE GNU/LINUX

3.1 ¿Qué es GNU?

Es el Movimiento que Richard Stallman en los años 80 inicio, frente a las políticas capitalistas que las grandes empresas de tecnología como IBM o AT&T, en las cuales ofrecían software a un elevado precio y que además el código fuente era cerrado y no existía libertad de modificación en estos. Stallman decidió ser consecuente con sus ideales e iniciar un gran proyecto para intentar abrir otra vez el código fuente de los programas. Consciente de que no podría conseguir que las compañías cedieran en este punto, se propuso crear su propio sistema operativo y aplicaciones iniciando un proyecto llamado GNU. Desarrollando la filosofía del software Libre o “free software y difundiénolo a nivel mundial.

3.2 ¿Qué es el software libre?

- Es aquel que puede utilizar y distribuir libremente. También puede ser modificado y vuelto a distribuir.
- Para que un programa sea considerado de software libre, éste debe cumplir cuatro libertades: libertad de uso, libertad de distribución, libertad de modificación y libertad de distribución de versiones modificadas. Estos suelen estar licenciados bajo la GPL GPL.

Para que un programa sea considerado software libre, debe cumplir 4 Libertades:

- Libertad de Uso.
- Libertad de distribución.
- Libertad de modificación.
- Libertad de distribución de versiones modificadas

Inicios del Software libre

- Richard Stallman(1983- Dunda GNU Ver 7)
- Linus Trovals(1991-Crea Linux)
- 1992 – GNU/Linux (Ver 7)



3.3 ¿Qué es el GNU/LINUX?

Para entender qué es Linux debemos saber antes qué es un sistema operativo(a veces se utilizan simplemente las siglas S.O.). Podemos dar una definición sencilla de este concepto:

“Un sistema operativo es un programa que permite al usuario interactuar con el ordenador y sus componentes (monitor, disco duro, impresora, etc) y que facilita la realización de tareas básicas como copiar o mover ficheros de un sitio a otro, editar archivos de texto, establecer una conexión a internet o hacer copias de seguridad”.

El sistema operativo es el primer programa que se ejecuta al encender el ordenador.

A un nivel superior tenemos los programas que permiten al usuario realizar tareas específicas. Estos programas se denominan aplicaciones de usuario, o simplemente aplicaciones. Podemos encontrar muchos ejemplos en el trabajo diario con el ordenador: programas de gestión contable como ContaPlus, procesadores de texto como OpenOffice.org Writer o MS Word, programas de retoque fotográfico como The Gimp o Photoshop

GNU/Linux (comúnmente Linux a secas) es uno más de los muchos S.O. que existen en la actualidad que tiene una serie de características que lo hacen especial:

- **Libre:** Se puede descargar de internet, se puede copiar y distribuir sin que por ello se incurra en ningún tipo de delito. La licencia que establece los términos de uso, copia y distribución se denomina

Licencia GNU (www.gnu.org)

- **Hecho por voluntarios:** Linux no se creó para obtener beneficios con él sino para satisfacer una serie de necesidades a la hora de trabajar con el ordenador. Hoy día sigue funcionando así. Cuando alguien necesita un determinado programa, simplemente lo crea y lo pone al servicio de la comunidad para que lo use y para que cada cual lo mejore y lo adapte a sus propias necesidades.

- **Multiusuario:** Varios usuarios pueden conectarse y usar el mismo ordenador a la vez.

- **Multitarea:** Pueden funcionar varios programas al mismo tiempo en la misma máquina.

- **Multiplataforma:** Hay versiones de Linux para gran cantidad de plataformas: todos los PCs basados en procesadores Intel o AMD, ordenadores Digital/Compaq con procesadores Alpha, ordenadores Apple, ultraportátiles como el Asus e incluso dispositivos móviles como el Sharp Zaurus.

- **Estable:** Linux es un sistema operativo muy maduro, probado durante mucho tiempo. Hay muchos servidores que llevan funcionando bajo Linux de forma ininterumpida muchos años sin un solo cuelgue.

- **Eficiente:** Linux aprovecha bien los recursos hardware. Incluso los viejos Pentium pueden funcionar bien con Linux y servir para alguna tarea.

- **Hay miles de programas libres:** Hay una gran cantidad de programas, desde procesadores de texto hasta programas de dibujo pasando por todo tipo de servidores, totalmente libres y gratuitos que se pueden descargar e instalar desde el propio entorno de Linux.

3.4 ¿QUÉ ES UNA DISTRIBUCIÓN DE LINUX?

Una distribución de Linux es el sistema operativo propiamente dicho, lo que se suele denominar el núcleo(kernel en inglés) junto con un programa de instalación y un conjunto de aplicaciones, normalmente de propósito general.

Distribución = Núcleo de Linux + Programa de instalación + Aplicaciones

Las primeras distribuciones eran difíciles de instalar pero actualmente apenas hay que introducir el CD e ir haciendo clic en “siguiente”,

Distribuciones populares :

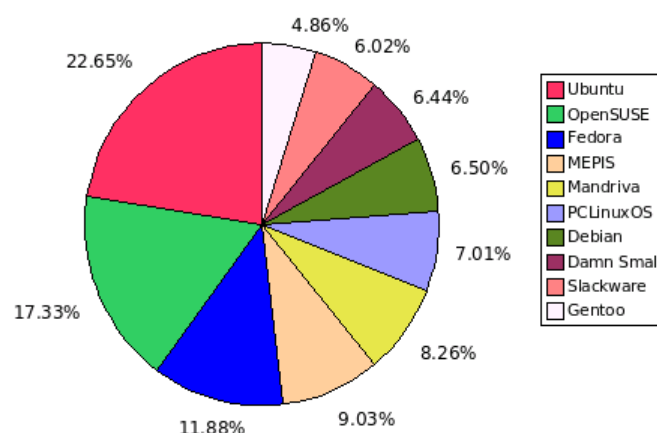
- **Ubuntu Desktop, Ubuntu Server:** Es el SO GNU/LINUX más utilizado en la actualidad. Está enfocada, sobre todo, a ordenadores de escritorio aunque también proporciona soporte para servidores. Está basada en Debian y sus principales características son la facilidad de uso e instalación. Se publica una versión cada 6 meses, una en abril y otra en octubre de cada año. Ubuntu está patrocinado por Canonical, una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth. El eslogan de Ubuntu es toda una declaración de intenciones: “Linux para seres humanos”. Resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.
- **openSUSE:** Se trata de una distribución auspiciada por las empresas Novell y AMD.
- **Linux Mint:** Está basada en Ubuntu y su meta es ofrecer un sistema “listo para funcionar” y que incluya plugins para el navegador, codecs para ver video, programas para reproducir DVD, Java, etc. de tal forma que el usuario se ahorre instalar y configurar estos componentes. Linux Mint es compatible con el software de Ubuntu.
- **Fedora:** Proviene de otra distribución llamada Red Hat que incluía tanto software libre como software propietario. El objetivo del proyecto Fedora es construir un sistema operativo completo, de propósito general, basado exclusivamente en software libre.
- **Debian:** Fue Ian Murdock, en 1993, quién inició el proyecto Debian e inicialmente estaba patrocinado por la Free Software Foundation. Es quizás la distribución que mejor ha sabido mantener a lo largo del tiempo la filosofía del proyecto inicial de GNU/Linux. Por su estabilidad y rendimiento, se utiliza con frecuencia en servidores cuya misión es crítica.

- **Mandriva:** Es la sucesora de la popular Mandrake y, como ella, incluye KDE como entorno gráfico. Hay todo un abanico de versiones de Mandriva: una totalmente gratuita, otra para arrancar desde un lápiz de memoria, otra con múltiples aplicaciones para servidores... La empresa del mismo nombre encargada de esta distribución ofrece también servicio técnico.
- **MEPIS:** La primera versión fue creada por Warren Woodford en el 2002 y está basada en Debian. Tiene KDE como entorno de escritorio y está dirigida especialmente a los usuarios que utilizan el ordenador como estación de trabajo, para tareas ofimáticas y administrativas más que para su uso como servidor. Tiene como punto fuerte un buen programa de instalación con posibilidad de reparticionar NTFS y recuperar el sistema operativo.
- **Slackware:** Creada por Patrick Volkerding, se trata de una de las distribuciones con más solera. Intenta mantener la tradición de ser una distribución fácil de usar y estable. Incluye servidores listos para usar en cuanto se instala el sistema: web, ftp y correo electrónico.

3.5 UBUNTU

Ubuntu es una de las distribuciones Linux que existen en la actualidad, a lo largo de los años ha tenido un impresionante desarrollo y aceptación por parte de la comunidad amante del software libre en todo el mundo

Ubuntu es un mundo distinto si somos usuarios de Windows, lo bueno que podemos resaltar de este sistema operativo es su estabilidad y robustez. En pocas palabras todo lo que tu haces en Microsoft Windows de igual forma lo podrás hacer en Ubuntu utilizando otras herramientas (programas). Este sistema operativo de libre distribución cuenta con muchas versiones las cuales se pueden descargar directamente desde su sitio web gratis.



Fue Lanzado en 2004 Apoyado por la empresa privada Canonical, fundada y financiada por el empresario sudafricano Mark Shuttleworth. A día de hoy es la distribución GNU/LINUX mas utilizada a nivel mundial. Su uso está pensado para trabajar tanto en entornos de Pc escritorio, laptops, Servidores y equipos móviles con arquitecturas ARM como celulares o tablets.



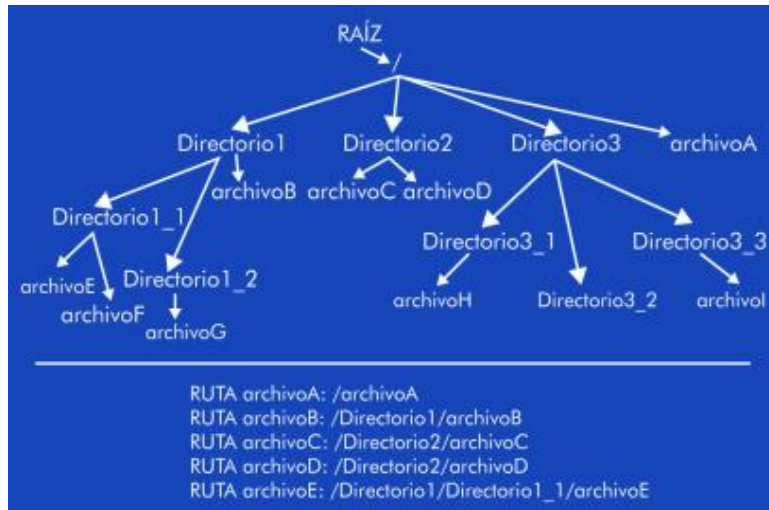
4. SISTEMA DE FICHEROS

El sistema de ficheros es el programa (o módulos del núcleo del operativo) que se encarga de realizar todas las operaciones relacionadas Con el almacenamiento y manipulación de los archivos. Son las funciones que tratan con los dispositivos físicos de almacenamiento del ordenador, como el disco duro

4.1 La jerarquía del sistema de ficheros

Todo sistema operativo necesita guardar multitud de archivos: desde los de la configuración del sistema, los de log, los de los usuarios, etc. En general, cada operativo utiliza su propio sistema de ficheros, caracterizándolo en muchos aspectos como pueden ser el rendimiento, la seguridad, la fiabilidad, etc. GNU/Linux es capaz de leer/escribir archivos con cualquiera de los sistemas de ficheros que actualmente existen, aunque para su propia raíz y directorios principales es necesario un sistema de ficheros que le permita ciertas operaciones. Generalmente, se suele utilizar el tipo ext2, ext3, ext4 o ReiserFS. EXT4 es el utilizado actualmente.

Lo primero que debemos tener claro es que todo el sistema de ficheros parte de una misma **raíz (root)**, a la cual nos referiremos con el carácter **" / "**. Es el origen de todo el sistema de ficheros y sólo existe una. Para organizar los ficheros adecuadamente, el sistema proporciona lo que llamaremos directorios (o carpetas), dentro de las cuales podemos poner archivos y más directorios. De este modo conseguimos una organización jerárquica como la que vemos en la siguiente figura:



4.2 ESTRUCTURA DE DIRECTORIOS.

Imagine el lector por un momento un montón de papeles amontonados en la mesa de una oficina: recibos del teléfono, facturas a clientes, notas tomadas en una reunión, factura de una reparación del coche. Buscar un documento entre todos estos papeles puede ser una pesadilla si están todos mezclados. La solución a este desorden es muy fácil: utilizar carpetas. Etiquetando carpetas y metiendo cada papel en su carpeta correspondiente, todo queda perfectamente ordenado. Puede haber una carpeta donde almacenar los pedidos a los proveedores, otra donde almacenar todas las notas tomadas en reuniones, otra para las facturas... Es más, puede haber subcarpetas dentro de algunas carpetas. Por ejemplo, la carpeta etiquetada como "Facturas" puede contener, a su vez, subcarpetas etiquetadas como "Teléfono", "Electricidad", "Coche... En un ordenador, el almacenamiento de información se lleva a cabo de la misma manera. Trabajando en el entorno gráfico se habla de carpetas y trabajando con comandos en un terminal, se habla de directorios, pero conceptualmente son exactamente lo mismo.

A continuación se muestra una tabla con los directorios más importantes de un sistema Linux:

/ directorio raíz	/bin	Contiene programas ejecutables básicos para el sistema.
	/boot	Contiene los ficheros necesarios para el arranque del sistema.
	/dev	Contiene los ficheros correspondientes a los dispositivos: sonido, impresora, disco duro, lector de cd/dvd, video, etc.
	/etc	Contiene ficheros y directorios de configuración.
	/home	Contiene los directorios de trabajo de los usuarios. Cada usuario tiene su propio directorio en el sistema dentro de /home/.
	/lib	Contiene las librerías compartidas y los módulos del kernel
	/media	Dentro de este directorio se montan los dispositivos como el CD-ROM, memorias USB, discos duros portátiles, etc
	/opt	Directorio reservado para instalar aplicaciones.
	/sbin	Contiene los ficheros binarios ejecutables del sistema operativo.
	/srv	Contiene datos de los servicios proporcionado por el sistema.
	/tmp	Directorio de archivos temporales.
	/usr	Aquí se encuentran la mayoría de los archivos del sistema, aplicaciones, librerías, manuales, juegos... Es un espacio compartido por todos los usuarios.
	/var	Contiene archivos administrativos y datos que cambian con frecuencia: registro de errores, bases de datos, colas de impresión, etc.
	/root	Directorio de trabajo del administrador del sistema (usuario root).
	/proc	Aquí se almacenan datos del kernel e información sobre procesos.

5 OPERACIONES BÁSICAS EN LA TERMINAL

5.1 TERMINAL O CONSOLA

Los sistemas operativos basados en Unix disponen de un intérprete de comandos u órdenes (conocido como terminal, consola o shell) que hace de interfaz entre el usuario y el propio sistema operativo. Es decir, mediante la terminal o consola podemos acceder al sistema operativo sin utilizar la interfaz gráfica y realizar todo tipo de tareas en modo texto.

La terminal o consola es una herramienta poderosa capaz de realizar tareas de forma más rápida que mediante la interfaz gráfica del sistema operativo.


```
marko@marko-VirtualBox: ~
marko@marko-VirtualBox:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Videos
Documentos examples.desktop  Música    Público
```

- Bash (bourne Again Shell), derivado de SH (UNIX)
- Cuando ingresamos por primera vez a la terminal de texto, nos encontramos con una línea de comandos similar a:

```
marko@marko-VirtualBox:~$
marko@marko-VirtualBox:/var/www$
```

marko: (Nombre de usuario)

@: (significa en)

marko-VirtualBox: (Nombre de equipo)

: ~ (indica en que directorio estamos posicionados actualmente)

\$:(indica si estamos como usuario común, si esta **#** indica que estamos como usuario root)

5.2 VIZUALIZACIÓN, CREACIÓN Y CAMBIO DE DIRECTORIO (pwd, ls, cd, mkdir)

- **pwd**

El comando `pwd` muestra cuál es el directorio de trabajo actual, en otras palabras, le dice al usuario dónde se encuentra dentro de la estructura de directorios del sistema. Es muy útil cuando estamos perdidos.

```
luisjose@ubuntu-desktop:~$ pwd
/home/luisjose
```

- **ls**

El comando `ls` muestra el contenido del directorio actual. Por defecto, los archivos ocultos no se muestran. Éste es seguramente el comando que más se utiliza.

```
luisjose@ubuntu-desktop:~$ ls
```

```
Desktop Documents Examples Music Pictures Public Templates Videos
```

Se pueden añadir opciones a `ls`, por ejemplo

```
ls -a
```

muestra todos los archivos, incluyendo los ocultos (cuyo nombre comienza por un punto),

```
ls -l
```

muestra un listado detallado, con la última fecha de modificación de cada archivo, el tamaño, etc.,

```
ls -h
```

muestra el tamaño de los ficheros en bytes, Kb, Mb, etc.

Todas las opciones disponibles, tanto para `ls` como para el resto de comandos se pueden consultar mediante las páginas del manual, con el comando `man` seguido del comando del que se quiere obtener información:

```
luisjose@ubuntu-desktop:~$ man ls
```

Esto dará información detallada sobre el comando `ls`. Para salir del manual basta pulsar la letra “q”

- **cd**

El comando `cd` (change dir) permite cambiar de directorio. Si se utiliza tal cual, sin ningún tipo de argumento, cambia al directorio de trabajo personal. Si se utiliza seguido de una ruta, cambia al directorio que se indica.

```
luisjose@ubuntu-desktop:~$ pwd
```

```
/home/luisjose
```

```
luisjose@ubuntu-desktop:~$ cd /etc
```

```
luisjose@ubuntu-desktop:/etc$ pwd
```

```
/etc
```

En este caso, el usuario estaba en su directorio de trabajo, y ha “saltado” al directorio `/etc`. El lector puede teclear el comando `ls` si tiene curiosidad por ver lo que hay dentro.

Las rutas pueden ser absolutas o relativas. Una ruta es absoluta cuando comienza por el carácter “/” y relativa cuando comienza por cualquier otro carácter. En el ejemplo anterior se ha

usado una ruta absoluta, esto es, /etc. Veamos cómo cambiar a otros directorios utilizando otras rutas absolutas:

```
luisjose@ubuntu-desktop:/$ cd /usr/local/
luisjose@ubuntu-desktop:/usr/local$ ls
bin etc games include lib man sbin share src
luisjose@ubuntu-desktop:/usr/local$ cd /var/spool/
luisjose@ubuntu-desktop:/var/spool$ ls
anacron cron cups cups-pdf mail openoffice
```

Una ruta relativa es algo así como una ruta parcial. La ruta que se aplica es la concatenación de la ruta actual y de la ruta relativa. Veamos un ejemplo:

```
luisjose@ubuntu-desktop:/var/spool$ cd
luisjose@ubuntu-desktop:~$ pwd
/home/luisjose
luisjose@ubuntu-desktop:~$ cd Music
luisjose@ubuntu-desktop:~/Music$ pwd
/home/luisjose/Music
```

Recordemos que el comando `cd` sin argumentos, nos lleva al directorio de trabajo personal.

En este caso,

```
cd Music
```

Sería equivalente a

```
cd /home/luisjose/Music
```

ya que se suma la ruta actual (/home/luisjose) a la ruta relativa indicada (Music) Las rutas, tanto las absolutas como las relativas se pueden utilizar en la mayoría de comandos. No son algo específico que se utilice sólo con `cd`. Podemos, por ejemplo, utilizar rutas como argumentos del comando `ls`.

```
luisjose@ubuntu-desktop:~/Music$ ls /boot/grub/
default          installed-version  minix_stage1_5    xfs_stage1_5
device.map        jfs_stage1_5      reiserfs_stage1_5
e2fs_stage1_5     menu.lst          stage1
fat_stage1_5      menu.lst~         stage2
```

Dos puntos (..) hacen referencia al directorio que hay justo a un nivel superior.

```
luisjose@ubuntu-desktop:~/Music$ ls ..
```

```
Desktop Documents Examples Music Pictures Public Templates Videos
```

`ls ..` muestra el contenido del directorio /home/luisjose que es el directorio que hay justo a un nivel superior de /home/luisjose/Music

```
luisjose@ubuntu-desktop:~/Music$ cd ..
```

```
luisjose@ubuntu-desktop:~$ pwd
```

```
/home/luisjose
```

cd ..sube un nivel en la estructura de directorios

- **mkdir**

Se pueden crear directorios con el comando mkdir. Por ejemplo, para crear una estructura de carpetas

donde un estudiante guardará información sobre sus asignaturas según el siguiente esquema:

/home/luisjose	/Documentos			
	/Escritorio			
	/Imágenes			
	/Música			
	/matematicas	/curso_01	/algebra	/exámenes_antiguos /apuntes
			/analisis	
			/fisica	/libros_de_ejercicios /videos
			/informatica	/compiladores_pascal
	/Video			

tendría que hacer lo siguiente:

```
~$ mkdir matematicas
```

```
~$ cd matematicas/
```

```
~/matematicas$ mkdir curso_01
```

```
~/matematicas$ cd curso_01/
```

```
~/matematicas/curso_01$ mkdir algebra analisis fisica informatica
```

```
~/matematicas/curso_01$ ls
```

```
algebra  analisis  física  informatica
```

```
~/matematicas/curso_01$ cd algebra/
```

```
~/matematicas/curso_01/algebra$ mkdir examenes_antiguos apuntes
```

```
~/matematicas/curso_01/algebra$ cd ..
```

```
~/matematicas/curso_01$ cd fisica
```

```
~/matematicas/curso_01/fisica$ mkdir libros_de_ejercicios
~/matematicas/curso_01/fisica$ mkdir videos
~/matematicas/curso_01/fisica$ cd ..
~/matematicas/curso_01$ cd informatica/
~/matematicas/curso_01/informatica$ mkdir compiladores_pascal
```

Nótese que ya no se muestra en el ejemplo el prompt completo, con el nombre de usuario y el nombre de la máquina. Se seguirá en el libro esta norma a partir de ahora.

5.3 VISUALIZACIÓN DE FICHEROS (**cat**, **more**, **less**, **head**, **tail**)

Los comandos **cat**, **more** y **less** sirven para mostrar el contenido de ficheros de texto. La diferencia radica en cómo se muestra el contenido. A todos estos comandos hay que pasarles como argumento el fichero que se quiere mostrar. Se puede indicar una ruta, en caso de que el fichero que se quiere mostrar no esté en el directorio actual. El comando **cat** muestra por pantalla el contenido de un fichero y, cuando termina, el usuario está otra vez de vuelta en la línea de comandos.

Por ejemplo,

```
~$ cat /var/log/dmesg
```

muestra el contenido del fichero **dmesg** que está dentro del directorio **/var/log**. Si el lector ha probado a hacerlo él mismo, se habrá dado cuenta de que es imposible ver todo el contenido de este fichero, porque ha pasado por pantalla muy rápido. Por eso **cat** se suele utilizar para visualizar el contenido de archivos pequeños.

El comando **more** hace lo mismo que **cat**, a diferencia de que muestra el fichero pantalla a pantalla, es

decir, llena de texto la pantalla y se espera a que el usuario pulse la tecla <espacio> para pasar a la siguiente:

```
~$ more /var/log/dmesg
```

El comando **less** es el más versátil de los tres, ya que permite moverse hacia delante y hacia atrás dentro del fichero, utilizando los cursores o las teclas de “AvPág” y “RePág”:

```
:~$ less /var/log/dmesg
```

En cualquier momento se puede interrumpir la visualización y volver al símbolo del sistema pulsando la letra “q”

Los comandos `head` y `tail` permiten mostrar de forma parcial el contenido de un fichero. Como su nombre indica, `head` muestra las primeras líneas del fichero (la cabecera) y `tail` muestra las últimas líneas (la cola). Veamos algunos ejemplos:

```
~$ head /boot/grub/menu.lst
# menu.lst - See: grub(8), info grub, update-grub(8)
# grub-install(8), grub-floppy(8),
# grub-md5-crypt, /usr/share/doc/grub
# and /usr/share/doc/grub-doc/.
## default num
# Set the default entry to the entry number NUM. Numbering starts from 0, and
# the entry number 0 is the default if the command is not used.
#
# You can specify 'saved' instead of a number. In this case, the default entry
~$ tail /boot/grub/menu.lst
root (hd0,0)
kernel /boot/vmlinuz-2.6.24-19-generic root=UUID=409e68a1-6123-476f-abf7-042854b68f3c
ro single
initrd /boot/initrd.img-2.6.24-19-generic
title Ubuntu 8.04.2, memtest86+
root (hd0,0)
kernel /boot/memtest86+.bin
quiet
### END DEBIAN AUTOMAGIC KERNELS LIST
```

Por defecto, tanto `head` como `tail` muestran 10 líneas, pero eso se puede cambiar con la opción `-n`.

```
~$ tail -n4 /boot/grub/menu.lst
kernel /boot/memtest86+.bin
quiet
### END DEBIAN AUTOMAGIC KERNELS LIST
```

En este caso se han mostrado solamente 4 líneas.

5.4 EDICIÓN DE FICHEROS (touch, vi, ee, mcedit)

El comando touch permite crear un fichero vacío. Con cualquier editor de texto se puede crear un fichero vacío pero con touch es especialmente cómodo y rápido.

```
~$ ls
Desktop  Documents  Examples  Music  Pictures  Public  Templates  Videos
~$ touch prueba.txt
~$ ls
Desktop      Examples  Pictures      Public  Videos
Documents    Music     prueba.txt  Templates
~$ cat prueba.txt
~$
```

Se puede ver en el ejemplo cómo se ha creado el archivo prueba.txt pero al visualizar su contenido con cat, no aparece nada en pantalla, por tanto está vacío

El programa ee es un editor muy rudimentario pero al mismo tiempo efectivo. Podemos editar el archivo anterior y escribir alguna frase:

```
~$ ee prueba.txt
```

Presionando la tecla ESC, el usuario puede salir al menú principal y guardar el fichero.

Podemos comprobar ahora cuál es el contenido del fichero:

```
~$ cat prueba.txt
```

Hola, aquí estoy aprendiendo Linux.

Otro editor muy simple es nano. Se deja al lector curioso probar su funcionamiento y compararlo con

ee. En el hipotético caso de no estar instalado alguno de estos editores, su instalación es muy sencilla,

basta con teclear `sudo apt-get install` seguido del nombre del programa que queremos instalar.

Por ejemplo, si queremos instalar ee:

```
~$ sudo apt-get install ee
```

El programa mcedit es un editor algo más sofisticado que ee o nano (al menos en apariencia) y es una parte de mc (Midnight Commander), un programa muy al estilo del famoso Norton Commander de MS-DOS. Vamos a modificar el archivo prueba.txt creado anteriormente.

Antes de eso, instalaremos mc, ya que no está instalado por defecto:

```
~$ sudo apt-get install mc
```

```
~$ mcedit prueba.txt
```

Con la tecla F2 guardamos los cambios y con dos pulsaciones de ESC (o con la tecla F10) salimos del programa.

Comprobamos ahora que todo se ha grabado bien:

```
~$ cat prueba.txt
```

Hola, aquí estoy aprendiendo Linux.

Me encanta, se pueden hacer muchas cosas.

Hemos dejado para el final al editor de Linux por excelencia, se trata de vi. A primera vista es el más difícil de utilizar, lo cual es cierto, y parece que tiene menos opciones, pero muy al contrario se trata de un programa muy potente. Cualquier “linuxero” que se precie debe saber manejar bien este programa.

Añadiremos una línea más al fichero prueba.txt. Para ello, seguiremos los siguientes pasos:

```
~$ vi prueba.txt
```

- Pulsar la letra “i” para entrar en modo “edición”.
- Escribiremos el texto.
- Pulsar la tecla ESC para salir del modo “edición”.
- Teclear “:” + “w” + INTRO para grabar los cambios.
- Teclear “:” + “q” + INTRO para salir del programa.

Comprobamos una vez más que todo está bien grabado:

```
~$ cat prueba.txt
```

Hola, aquí estoy aprendiendo Linux.

Me encanta, se pueden hacer muchas cosas.

¡Pronto dominaré el editor Vi!

Es más que recomendable realizar el tutorial llamado `vimtutor`.

RESUMEN DEL CAPÍTULO

- Todo usuario necesita un nombre y una contraseña para entrar en el sistema.
- La información se almacena físicamente en directorios y subdirectorios (carpetas y subcarpetas).
- Hay una serie de directorios predefinidos como `/bin`, `/dev`, `/home`, `/etc`, `/var`, etc. para todos los sistemas Linux.

- Hay rutas absolutas, que comienzan por el carácter “/”, y que definen una ruta efectiva completa y rutas relativas, que no comienzan por el carácter “/”, y cuya ruta efectiva sería la concatenación del directorio actual con esa misma ruta relativa.
- Los comandos vistos en este capítulo son los siguientes:

<i>Comando</i>	<i>Acción</i>	<i>Ejemplo</i>
pwd	muestra el directorio actual	pwd
ls	lista ficheros y directorios	ls -l
cd	cambia de directorio	cd mp3/wim_mertens
mkdir	crea uno o varios directorios	mkdir cartas facturas
cat	visualiza un fichero	cat /var/log/dmesg
more	visualiza un fichero pantalla a pantalla	more /var/log/dmesg
less	visualiza un fichero pantalla a pantalla y permite retroceder	less /var/log/dmesg
head	visualiza las primeras filas de un fichero	head -n5 /var/log/dmesg
tail	visualiza las últimas filas de un fichero	tail /var/log/dmesg
touch	crea un fichero vacío	touch listado.txt
ee	editor de textos muy simple	ee listado.txt
mcedit	editor de textos que forma parte de Midnight Commander	mcedit listado.txt
vi	editor de textos muy potente	vi listado.txt
apt-get	instala y desinstala programas	apt-get install mc
man	muestra ayuda sobre un determinado comando	man ls

5.5 COPIA Y BORRADO DE FICHEROS (cp, mv, rm)

• 3.2.1 cp

El comando `cp` sirve para copiar ficheros. Se puede copiar un único fichero o muchos. Se pueden copiar tanto ficheros como directorios. Por supuesto, se pueden utilizar los símbolos comodines. En el proceso de copia intervienen tres factores: lo que se copia, la ruta de origen y la ruta de destino. No

Está de más recordar que las rutas pueden ser tanto absolutas como relativas. La ruta de origen se

Especifica junto con lo que se quiere copiar. Veamos un ejemplo:

```
$ cp /etc/hosts /home/alumno/pruebas/
```

La sentencia anterior copia el fichero hosts, que se encuentra en el directorio /etc al directorio /home/alumno/pruebas/.

Si no se especifica ningún directorio origen, se toma por defecto el directorio actual. Por ejemplo:

```
$ cp *.odt textos/
```

copia todos los archivos con la extensión odt del directorio actual al directorio textos. Cuando se quiere especificar como directorio destino el directorio actual se utiliza el carácter "." Por ejemplo:

```
$ cp /usr/bin/g* .
```

copia todos los ficheros del directorio /usr/bin que comienzan por la letra g al directorio actual.

- **mv**

El comando mv sirve para dos cosas, para mover y para cambiar de nombre. Se puede hacer cualquiera de las dos cosas por separado o las dos cosas al mismo tiempo. Por ejemplo:

```
$ mi_texto.txt carta.txt
```

le cambia el nombre a mi_texto.txt y pasa a llamarse carta.txt. En cambio

```
$ mv carta.txt Documentos/
```

mueve carta.txt al directorio Documentos.

Se pueden hacer las dos cosas a la vez, mover y cambiar el nombre:

```
~$ cd Documentos/
```

```
~/Documentos$ mkdir correspondencia
```

```
~/Documentos$ mv carta.txt correspondencia/carta01.txt
```

En este caso, el fichero carta.txt se ha movido al directorio ~/Documentos/correspondencia y además se le ha cambiado el nombre a carta01.txt

- **rm**

El comando rm se utiliza para borrar ficheros. Es importante destacar que estos ficheros no se envían a una papelera así que NO SE PUEDEN RECUPERAR UNA VEZ BORRADOS.

Ejemplo:

```
$ rm *.txt
```

Esta sentencia borra todos los archivos con la extensión txt del directorio actual.

5.6 COPIA Y BORRADO DE DIRECTORIOS (cp, mv, rm)

De la misma manera que se copian, se borran o se mueven ficheros, se puede hacer lo mismo con los directorios. Hay que tener en cuenta que un directorio puede contener muchos ficheros y, además, otros directorios que, a su vez, pueden contener más ficheros y directorios. Por tanto, si se quiere copiar un fichero completo, con todo lo que tiene dentro, hay que indicarlo con la opción -R. A esto último se suele llamar “copiar de forma recursiva”.

Ejemplo:

```
~$ mkdir multimedia2
~$ cp multimedia/* multimedia2
cp: se omite el directorio «multimedia/imagenes»
cp: se omite el directorio «multimedia/musica»
cp: se omite el directorio «multimedia/presentaciones»
cp: se omite el directorio «multimedia/video»
~$ ls multimedia2
~$
```

Se ha hecho una copia del contenido del directorio `multimedia` al directorio `multimedia2` pero no se ha copiado ningún archivo ¿qué ha pasado? Sencillamente no se ha hecho una copia recursiva (con la opción -R). Se ha intentado copiar únicamente justo dentro del directorio `multimedia` pero no a un nivel inferior. Como a ese nivel no había ningún fichero, no se ha copiado nada.

Vamos a intentarlo ahora de forma recursiva:

```
~$ cp -R multimedia/* multimedia2
~$ ls -R multimedia2
multimedia2:
imagenes  música  presentaciones  video
multimedia2/imagenes:
otras  personales
multimedia2/imagenes/otras:
multimedia2/imagenes/personales:
multimedia2/musica:
estilos_favoritos.txt
```

multimedia2/presentaciones:

multimedia2/video:

Como se puede comprobar, se han copiado tanto la estructura de directorios como los contenidos de cada uno de ellos.

El comando `mv` funciona de forma análoga a `cp`, pero mueve en lugar de copiar. Cuando se trata de renombrar, funciona exactamente igual que con los ficheros.

Ejemplo:

```
~$ mv multimedia2 multimedia_copia
```

Esto le cambia el nombre al directorio `multimedia2` y pasa a llamarse `multimedia_copia`. El lector puede comprobar que el contenido de ese directorio permanece intacto.

Conrmse pueden borrar directorios.

```
~$ rm multimedia_copia/
```

`rm`: no se puede borrar «`multimedia_copia/`»: Es un directorio

Se obtiene un error, ¿qué sucede? Si el lector es perspicaz, sabrá cómo solucionar este problema...

...En efecto, hay que borrar el contenido de forma recursiva:

```
~$ rm -Rf multimedia_copia/
```

Además de la opción `-R`, se ha incluido la opción `-f` que hace que no se nos pida confirmación por cada elemento que se quiere borrar

• Los comandos vistos en este capítulo son los siguientes:

<i>Comando</i>	<i>Acción</i>	<i>Ejemplo</i>
cp	copia archivos o directorios	<code>cp *.txt correspondencia/</code>
mv	mueve o renombra archivos o directorios	<code>mv palabras.txt texto.txt</code>
rm	borra archivos o directorios	<code>rm -R cosas/basurilla</code>
rmdir	borra directorios	<code>rmdir viejo</code>

5.7 INSTALACION DE PAQUETES

Actualizacion de la lista de repositorios de software en el sistema operativo

```
apt-get update
```

Actualización de de software en el sistema operativo

```
apt-get upgrade
```

Instalacion de Paquetes de software

```
apt-get install [nombre del paquete de software]
```

Ejemplo, instalaremos el famoso servidor web http apache:

```
apt-get install apache2
```

5.8 DESINSTALACION DE PAQUETES

Realizar esta 3 operaciones:

```
apt-get remove [nombre de paquete de software]
```

```
apt-get purge [nombre de paquete de software]
```

```
apt-get autoremove [nombre de paquete de software]
```

```
apt-get autoclean [nombre de paquete de software]
```

Ejemplo, desinstalaremos el paquete de software apache

```
apt-get remove apache2
```

```
apt-get purge apache2
```

```
apt-get autoremove apache2
```

```
apt-get autoclean apache2
```

6 ¿POR QUÉ EXISTEN GRUPOS, USUARIOS Y PERMISOS?

Vimos en un capítulo anterior que los ficheros deben estar organizados en directorios (carpetas) con el fin de tenerlos ordenados y poder localizarlos convenientemente. Volvamos a nuestro ejemplo de la oficina. Cada papel está en su sitio, hay carpetas y subcarpetas y todo está organizado. Ahora bien, el contable deberá tener acceso por ejemplo a las carpetas donde se encuentras las facturas y los recibos pero no tienen por qué tener acceso a la información sobre desarrollo de productos o marketing. En un sistema Linux, las carpetas y los archivos funcionan de esta manera. Por ejemplo, los archivos de configuración que se encuentran en el directorio `/etc` sólo pueden ser modificados por el administrador del sistema. Esto previene que cualquier usuario pueda cambiar información crítica y estropear algo.

7 ¿QUÉ ES EL SUPERUSUARIO?

El superusuario, administrador del sistema o simplemente el `root`, es un usuario especial que tiene privilegios para cambiar la configuración, borrar y crear ficheros en cualquier directorio, crear nuevos grupos y usuarios, etc.

IMPORTANTE: ES PELIGROSO TRABAJAR COMO SUPERUSUARIO, SE PUEDE DAÑAR EL SISTEMA DE FORMA IRREVERSIBLE. EL LECTOR DEBE ESTAR SEGURO DE LO QUE HACE CUANDO TRABAJE COMO SUPERUSUARIO.

Una vez hecha esta aclaración, pasemos a hacer algo como `root`:

```
$ touch /etc/prueba.txt
touch: no se puede efectuar `touch' sobre «/etc/prueba.txt»: Permiso denegado
$ sudo touch /etc/prueba.txt
$ ls /etc/pru*
etc/prueba.txt
```

Hemos intentado primero crear el fichero `prueba.txt` en el directorio `/etc` como usuario normal y acto seguido hemos obtenido un error de “Permiso denegado”, lo que quiere decir que un usuario sin privilegios no puede hacer eso. A continuación lo hemos intentado como administrador, para ello hemos usado el comando `sudo`, tras lo que se nos ha preguntado la clave del administrador. Esta vez sí lo hemos conseguido. No tendría mucho sentido que el sistema no preguntase por la clave, ya que en ese caso cualquiera podría ejecutar comandos como administrador con el peligro que ello supone.

8. PERMISOS

La información sobre grupos, usuarios y permisos se puede obtener mediante el comando `ls` junto con la opción `-l`. Vamos a ver los permisos que tiene establecidos el fichero `whatis` que se encuentra en el directorio `/usr/bin`.

```
$ ls -l /usr/bin/whatis
-rwxr-xr-x 1 root root 87792 2008-03-12 14:24 /usr/bin/whatis
```

En la primera columna aparecen los permisos, en la tercera se indica el usuario (en este caso es el administrador del sistema) y en la cuarta columna aparece el nombre del grupo (que en este caso coincide con el de usuario).

Vamos a ver qué significan exactamente los caracteres de la primera columna:

-	r	w	x	r	-	x	r	-	x
Tipo de fichero.	Permisos para el dueño del fichero.			Permisos para el grupo al que pertenece el fichero.			Permisos para el resto de usuarios		

r	Permiso de lectura .
w	Permiso de escritura .
x	Permiso de ejecución .

El tipo de fichero se indica en la siguiente tabla:

<i>Tipo de fichero</i>	
l	Enlace simbólico.
c	Dispositivo especial de caracteres.
b	Dispositivo especial de bloques.
p	FIFO (estructura de datos).
s	Socket (comunicaciones).
-	Ninguno de los anteriores. Puede ser un fichero de texto, un binario, etc.

En el caso que nos ocupa tenemos un carácter “-” como tipo de fichero, porque se trata de un binario (un programa). El dueño del fichero tiene los permisos `rwX`, lo que quiere decir que puede leer, escribir y ejecutar el fichero. Que tiene permiso para escribir significa que puede borrarlo, cambiarle el nombre o editarlo. Tanto el grupo como el resto de usuarios tienen los permisos `r-X`, lo que significa que pueden utilizarlo (pueden leerlo y ejecutarlo) pero no lo pueden modificar.

9 ¿QUIÉNES SOMOS? (whoami, groups)

Antes de empezar a crear usuarios, crear grupos y cambiar permisos, debemos saber quiénes somos y a qué grupo o grupos pertenecemos. Aunque, en principio, entremos en el sistema como un determinado usuario, podemos utilizar `su` para ejecutar comandos como otro usuario distinto, siempre y cuando sepamos la contraseña de ese otro usuario.

```
$ whoami
```

```
luisjose
```

```
$ su alumno
Contraseña:
$ whoami
alumno
```

Para volver a ser el usuario original basta con utilizar exit.

```
$ whoami
alumno
$ exit
exit
$ whoami
luisjose
```

Con el comando groups se puede ver a qué grupo pertenecemos.

```
luisjose@luisjose-xps1330:~$ groups
luisjose adm dialout cdrom floppy audio dip video plugdev scanner lpadmin admin
netdev powerdev sambashare
```

Se pueden especificar uno o más usuarios detrás de groups. Eso nos dirá a qué grupos pertenece cada uno de ellos.

```
luisjose@luisjose-xps1330:~$ groups alumno root
alumno : alumno
root : root
```

10. GESTIÓN DE GRUPOS (groupadd, groupdel, groupmod)

Los comandos groupadd, groupdel y groupmod permiten crear, borrar y modificar grupos respectivamente.

Vamos a crear los grupos oficina_malaga, oficina_jaen y oficina_madrid

```
$ groupadd oficina_malaga
groupadd: incapaz de bloquear el fichero de grupos
$ sudo groupadd oficina_malaga
$ sudo groupadd oficina_jaen
$ sudo groupadd oficina_madrid
```

Vemos que si intentamos crear un grupo como usuario sin privilegios obtenemos un error. Para manejar grupos y usuarios es necesario ejecutar los comandos con privilegios de administrador, por tanto deberemos teclear sudo antes del comando en cuestión.

Hemos escrito mal el nombre del segundo grupo, ¡que no cunda el pánico!, este problema se puede solventar con groupmod.

```
$ sudo groupmod -n oficina_madrid oficina_madrit
```

La directiva de la empresa ha decidido cerrar la oficina de Jaén para ahorrar costes y pasar los recursos a la oficina de Málaga, así que no hará falta el grupo oficina_jaen. Lo podemos borrar con groupdel.

```
$ sudo groupdel oficina_jaen
```

11. GESTIÓN DE USUARIOS (adduser, userdel, usermod)

La gestión de usuarios, al igual que la de grupos, exige que los comandos se ejecuten con los privilegios del administrador del sistema. Se puede escribir sudo antes de cada comando, o se puede hacer lo siguiente:

```
$ sudo bash
```

Note el lector que el prompt ha cambiado. Ahora se muestra un carácter “#” en lugar de un “\$”. A partir de ahora, todos los comandos se ejecutarán con privilegios de administrador del sistema. Hay que acordarse de volver al usuario inicial mediante exit. Es necesario dar de alta a dos usuarios para el grupo oficina_malaga y uno para oficina_madrid. Habrá un cuarto usuario que estará yendo y viniendo de una oficina a otra, por tanto se le dará de alta en las dos.

```
# adduser pedro --ingroup oficina_malaga
# adduser ana --ingroup oficina_malaga
# adduser berta --ingroup oficina_madrid
# adduser laura --ingroup oficina_malaga
# adduser laura oficina_madrid
```

Hemos matado dos pájaros de un tiro. Hemos creado los usuarios y al mismo tiempo los hemos incluido dentro de los grupos correspondientes. Estos dos pasos se pueden hacer de forma independiente. El usuario laura pertenece a dos grupos. En primer lugar se ha creado el usuario y al mismo tiempo se ha añadido al grupo oficina_malaga con la opción –ingroup. Para añadir un usuario existente a un grupo, se utiliza adduser sin opciones.

```
# groups ana berta laura
ana : oficina_malaga
berta : oficina_madrid
laura : oficina_malaga oficina_madrid
```

Es importante destacar que se ha utilizado `adduser` no `useradd`. Este último se considera un comando de bajo nivel y se recomienda utilizar el primero.

Al crear los usuarios, se nos han pedido las claves, no obstante estas claves se pueden cambiar con el comando `passwd`.

```
# passwd pedro
# passwd ana
# passwd laura
```

Recuerde el lector salir del modo `root` con el comando `exit` cuando no tenga que hacer tareas que requieran privilegios de administrador.

```
# exit
```

De ahora en adelante, simplemente se indicará con el carácter “\$” que se trabaja como usuario sin privilegios y con el carácter “#” que se trabaja como `root`.

Cabe señalar que para cada usuario, se crea por defecto un directorio dentro de `/home`. Cuando un usuario se conecta al sistema, “atteriza” en ese directorio. Es lo que hemos denominado anteriormente como el directorio de trabajo.

```
$ ls /home/
Alumno  ana  berta  ftp  laura  luisjose  pedro
```

12 CAMBIO DE GRUPO Y DE DUEÑO (`chown`, `chgrp`)

Imaginemos que el fichero `informe.txt` ha sido creado por el usuario `pedro`. Por defecto, el dueño

de un archivo es el usuario que lo crea, en este caso `pedro`. El grupo del usuario `pedro`, como hemos

visto antes es `oficina_malaga`.

```
$ su pedro
$ cd
$ pwd
/home/pedro
$ touch informe.txt
$ ls -l
-rw-r--r-- 1 pedro oficina_malaga 0 2009-03-19 12:46 informe.txt
```

Todo esto se puede cambiar. Moveremos el fichero al directorio de trabajo del usuario laura y le cambiaremos el dueño.

```
# mv informe.txt /home/laura/
# cd /home/laura/
# chown laura informe.txt
# ls -l
-rw-r--r-- 1 laura oficina_malaga 0 2009-03-19 12:46 informe.txt
```

Ahora el fichero tiene al usuario laura como propietario.

Tanto `chown` como `chgrp` se pueden usar con la opción `-R` para cambiar el dueño o el grupo en un directorio completo, de forma recursiva.

13 CAMBIO DE PRIVILEGIOS (`chmod`)

El comando `chmod` sirve para cambiar los permisos de uno o varios ficheros. Esos mismos permisos que se pueden ver con `ls -l`.

```
$ ls -l
-rw-r--r-- 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
$ chmod +x hola_mundo.rb
$ ls -l
-rwxr-xr-x 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
```

Hemos añadido el permiso de ejecución al fichero `hola_mundo.rb`. Vemos que ahora hay tres x, la

que corresponde al dueño del fichero, la de todos los usuarios que pertenecen al grupo y la del resto de

usuarios. Cuando no se especifica ninguna de estas tres letras correspondientes a los usuarios (u, g, o) como en el ejemplo anterior, se sobreentiende que nos referimos a todos ellos. Se puede indicar de forma explícita con el carácter `a` (all).

Para entenderlo mejor, en la siguiente tabla, se muestran de forma esquemática, los parámetros del comando `chmod`:

u	g	o	+ -	r	w	x
(user) dueño del fichero	(group) usuarios que pertenecen al mismo grupo	(others) el resto de usuarios	dar permiso quitar permiso	(read) lectura	(write) escritura	(execution) ejecución

Quitaremos ahora el permiso de ejecución para el resto de usuarios (others) y daremos permiso de escritura (write) a los usuarios del mismo grupo (group).

```
$ ls -l
-rwxr-xr-x 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
$ chmod o-x hola_mundo.rb
$ chmod g+w hola_mundo.rb
$ ls -l
-rwxrwxr-- 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
```

A este método, que utiliza los caracteres rwxse le denomina método simbólico. Podemos utilizar de forma análoga el método numérico.

4	2	1	Total
r	w	x	4 + 2 + 1 = 7
r	w	-	4 + 2 + 0 = 6
r	-	x	4 + 0 + 1 = 5
r	-	-	4 + 0 + 0 = 4
-	w	x	0 + 2 + 1 = 3
-	w	-	0 + 2 + 0 = 2
-	-	x	0 + 0 + 1 = 1

De esta forma, esta línea

```
$ chmod 755 hola_mundo.rb
sería equivalente a estas tres
$ chmod u+rwx hola_mundo.rb
$ chmod g+rx-w hola_mundo.rb
$ chmod o+rx-w hola_mundo.rb
```

En efecto

```
$ ls -l
```

```
-rwxr-xr-x 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
```

Los permisos de los directorios se pueden cambiar de la misma forma que los ficheros, aunque el significado es algo diferente. Si un directorio tiene el permiso de lectura quiere decir que se puede ver su contenido. Si tiene permiso de escritura, quiere decir que se pueden crear ficheros dentro y si tiene permiso de ejecución quiere decir que se puede entrar dentro.

<i>Comando</i>	<i>Acción</i>
ls -l	Muestra, entre otras cosas, información sobre los permisos, el usuario y el grupo al que pertenece el fichero.
sudo	Permite ejecutar comandos como root.
su	Cambia de usuario.
whoami	Muestra el nombre del usuario actual.
groups	Muestra el/los grupos/s a los que pertenece el usuario actual.
groupadd	Añade un nuevo grupo.
groupdel	Borra un grupo.
groupmod	Modifica las características de un grupo.
adduser	Añade un nuevo usuario.
userdel	Borra un usuario.
usermod	Modifica las características de un usuario.
passwd	Asigna o cambia la clave de un usuario.
chown	Cambia el dueño de un archivo.
chgrp	Cambia el grupo al que pertenece un archivo.
chmod	Cambia los permisos.

14. GESTIÓN DE SERVICIOS EN UBUNTU

systemctl , que es la herramienta de administración central para controlar el sistema init. Cubriremos cómo administrar servicios, verificar estados, cambiar estados del sistema y trabajar con los archivos de configuración.

Para las tareas de administración de servicios, la unidad de destino será unidades de servicio, que tienen archivos de unidad con un sufijo de .service . Sin embargo, para la mayoría de los comandos de administración de servicios, puede dejar el sufijo .service , ya que systemd es lo suficientemente inteligente como para saber que probablemente quiera operar en un servicio cuando utiliza comandos de administración de servicios.

- **sudo systemctl status** -> muestra el status de todos los servicios
- **sudo systemctl start <application>** -> Inicia Servicio
- **sudo systemctl stop <application>** -> Detiene Servicio
- **sudo systemctl restart <application>** -> Para reiniciar un servicio
- **sudo systemctl reload <application>** -> Recarga (sin reiniciar)
- **sudo systemctl enable <application>** -> habilita inicio automatico
- **sudo systemctl disable <application>** -> desabilita inicio automatico
- **sudo systemctl status <application>** -> muestra estado de servicio

```

root@ip-172-31-8-231:~# systemctl status apache2
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Sun 2017-10-15 00:05:06 UTC; 7h ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1253 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCESS)
    Tasks: 55
   Memory: 10.6M
      CPU: 11.194s
   CGroup: /system.slice/apache2.service
           └─1327 /usr/sbin/apache2 -k start
             └─1330 /usr/sbin/apache2 -k start
               └─1331 /usr/sbin/apache2 -k start

Oct 15 00:05:05 ip-172-31-8-231 systemd[1]: Starting LSB: Apache2 web server...
Oct 15 00:05:05 ip-172-31-8-231 apache2[1253]: * Starting Apache httpd web server apache2
Oct 15 00:05:06 ip-172-31-8-231 apache2[1253]: *
Oct 15 00:05:06 ip-172-31-8-231 systemd[1]: Started LSB: Apache2 web server.

```