

Introducción a Linux

Antes de poder trabajar con línea de comandos del Shell de Linux, primero debe de comprender que es Linux, de donde procede y como trabaja. En esta sección se explica que es Linux y donde encaja la línea de comandos del Shell dentro del sistema operativo.

¿Qué es Linux?

Si usted nunca ha trabajado con Linux, puede ser confuso entender porque hay disponible tantas versiones diferentes. Al intentar buscar software en ambiente Linux, seguramente ha estado confundido al escuchar varios términos como distribución, LiveCD y GNU. Sumergirse en el mundo de Linux por primera vez es una experiencia un poco confusa. Esta sección explica algunos de los fundamentos básicos del sistema Linux que son necesarios comprender antes de iniciar a trabajar con los comandos y los Shell scripts.

Los cuatro componentes básicos del sistema Linux son:

- El kernel de Linux
- Las utilidades GNU
- El ambiente de escritorio gráfico
- El software de aplicación

Cada de uno de estos componentes desempeñan un trabajo específico en el sistema. Ningún componente es útil por sí mismo. La figura 3.1 muestra un diagrama de como encajan para formar juntos el sistema completo de Linux:

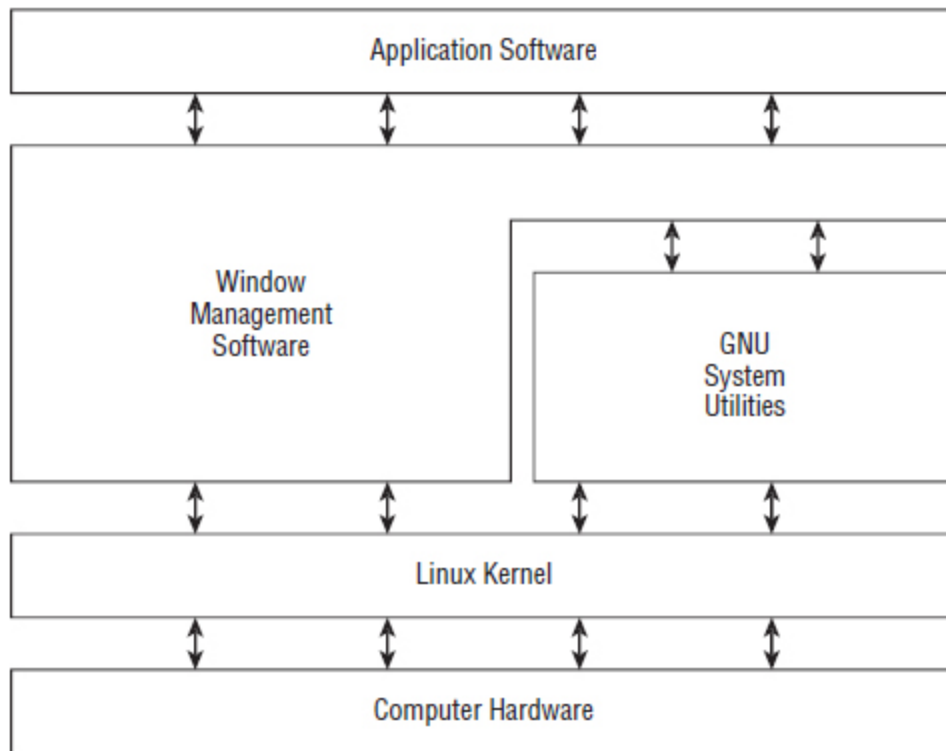


Figura 3.1 Componentes del sistema Linux

Esta sección describe en detalle las cuatro partes principales y da un vistazo de como trabajan juntas para crear sistema completo de Linux.

El kernel de Linux

El corazón del sistema Linux es el kernel. El kernel controla todo el hardware del sistema computacional, reservando hardware cuando es necesario y ejecutando software cuando se requiere.

Si alguna vez a seguido el mundo de Linux probablemente haya escuchado el nombre Linus Torvalds. Linus es la persona responsable de la creación del primer kernel de Linux, siendo estudiante de la universidad de Helsinski. El pretendía hacer una copia del sistema Unix, un sistema operativo que gozaba de mucha popularidad en el momento, utilizado por muchas universidades.

Después de desarrollar el kernel de Linux, Linus lo liberó a la comunidad de Internet y solicitó sugerencias para mejorarlo. Este simple proceso inició una revolución en el mundo de los sistemas operativos de computadoras.

Permitir que cualquiera hiciera cambios en el código de programación del kernel resultaría en un completo caos. Para simplificar las cosas, Linus actuó como el responsable de la coordinación ante cualquier sugerencia de mejora. La decisión final de incorporar o no cualquier código en el kernel estaba en las manos de Linus.

Este mismo concepto todavía se aplica en la codificación del kernel de Linux, sin embargo, en lugar de ser controlado por Linus, un equipo de desarrollo se encarga de coordinar dicha tarea.

Las cuatro funciones principales del kernel son:

1. Administración de la memoria:

El kernel no solamente se encarga de gestionar la memoria sino también es quien crea y administra la memoria virtual, para ello utiliza un espacio sobre el disco duro llamado memoria swap o memoria de intercambio. El kernel intercambia el contenido localizado en la memoria virtual hacia y desde la memoria física. Esto le permite al sistema creer que hay más memoria disponible de la que realmente existe. La figura 3.2 describe la memoria de intercambio:

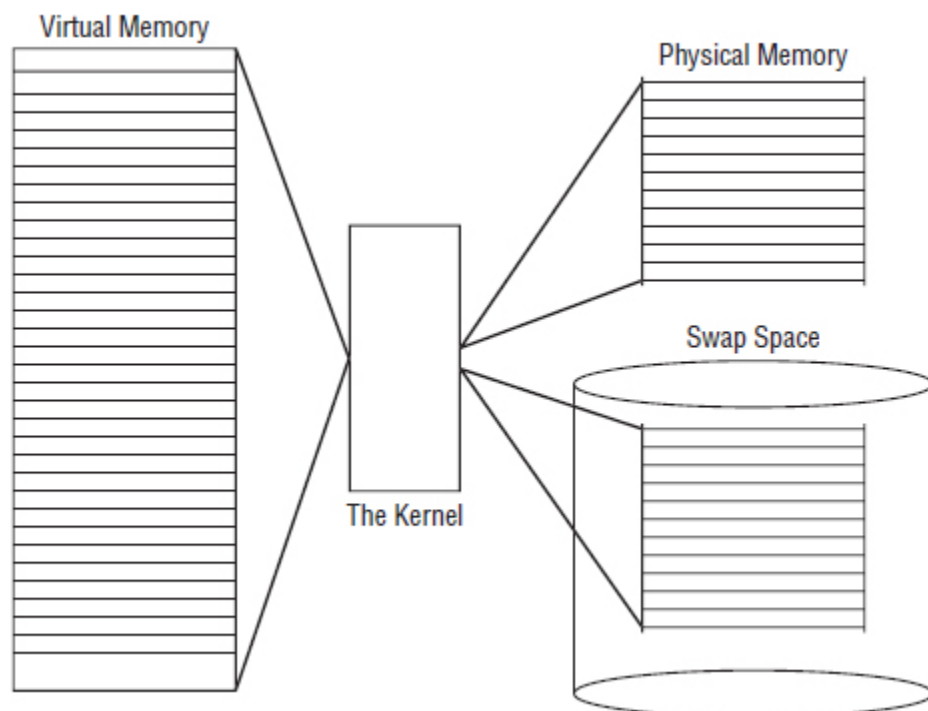


Figura 3.2 mapa del sistema de memoria en Linux

Las ubicaciones en memoria son agrupadas en bloques llamados páginas. El kernel ubica cada espacio de memoria ya sea en la memoria física o en el espacio de intercambio. El

kernel mantiene una tabla de las páginas de memoria que indica cual página está en memoria física y cual página está en espacio de intercambio o disco duro.

2. Administración de los programas:

Linux llama proceso a todo programa que corre en memoria. Un proceso puede correr en primer plano, por ejemplo, mostrando su salida a través de la pantalla, o puede estar corriendo en segundo plano sin mostrar ninguna actividad al usuario. El kernel controla la forma en que Linux administra todos los procesos que corren en el sistema.

El kernel crea un primer proceso, llamado proceso init, encargado de iniciar todos los otros procesos del sistema. Cuando inicia el kernel, este inicia el proceso init dentro de la memoria virtual. A medida que el kernel inicia cada proceso adicional, este le asigna un área única dentro de la memoria virtual para almacenar los datos y el código que usa el proceso.

Algunas implementaciones de Linux contienen una tabla de procesos que inician automáticamente en el arranque del sistema. En sistemas Linux, normalmente esta tabla está localizada en el archivo especial `/etc/inittabs`.

Otros sistemas, tales como la distribución utilizada en este curso (Ubuntu) utilizan la carpeta `/etc/init.d`, la cual contiene scripts para arrancar y detener en tiempo de inicio aplicaciones individuales. Los scripts son iniciados a través de entradas ubicadas bajo las carpetas `/etc/rcX.d`, en donde X corresponde al nivel de inicio (run level).

El sistema operativo Linux usa un sistema init que utiliza niveles de arranque (run levels). Un nivel de arranque puede ser usado para indicarle al proceso init que corra solo ciertos tipos de procesos, así como es definido en el archivo `/etc/inittabs` o los directorios `/etc/rcX.d`. Existen 5 niveles de arranque en los sistemas operativos Linux.

En el nivel de arranque 1, solamente son iniciados los procesos básicos del sistema, así como una consola o terminal de procesos. Este modo es llamado modo usuario único (single-user mode). El modo de usuario único es utilizado en casos de mantenimientos de emergencia del sistema de archivo. Obviamente, en este modo, solo una persona (comumente el administrador o usuario root) puede hacer login al sistema para la manipulación de los datos.

El nivel de arranque estándar es el 3. En este nivel de arranque, son iniciados la mayoría de programas, tales como el software de soporte de red. Otro nivel de arranque popular es el 5. Este es el nivel de arranque en donde el sistema inicia el programa gráfico X Windows, este es un software que permite a los usuarios utilizar el escritorio gráfico de ventanas.

Linux puede controlar las funcionalidades del sistema completo controlando los niveles de arranque. Por ejemplo, al intercambiar el nivel 3 por el 5, el sistema puede cambiar de un

sistema basado en consola de comandos a un sistema avanzado basado en interfaz gráfica X Windows.

Mas adelante, verá como usar el comando `ps` para observar los procesos que actualmente corren en el sistema Linux.

3. Administración del hardware

Otra responsabilidad del kernel es administrar el hardware. Cualquier dispositivo que deba comunicarse con el sistema Linux necesita un código controlador (driver) insertado dentro del kernel. El código controlador le permite al kernel transferir datos hacia y desde el dispositivo, actuando como un middleware (software intermedio) entre las aplicaciones y el hardware. Se usan dos métodos para insertar código controlador de dispositivos en el kernel:

- Controladores compilados en el kernel
- Módulos controladores agregados al kernel

En el pasado, la única forma de insertar controladores era a través de la recompilación del kernel. Cada vez que se agregaba un nuevo dispositivo al sistema, se tenía que recompilar el código del kernel. Este proceso se volvió mas ineficiente a medida que el kernel de Linux soportaba mas hardware. Afortunadamente, los desarrolladores de Linux propusieron una mejor solución para insertar código controlador dentro de cualquier kernel en ejecución.

Los programadores desarrollaron el concepto de módulos del kernel para permitirle al usuario final insertar código controlador dentro del kernel en ejecución, todo esto sin necesidad de recompilarlo. Igualmente, un modulo del kernel puede ser removido cuando el usuario ha terminado de utilizar el dispositivo. Este cambio significativo simplificó y expandió enormemente el uso de más hardware en Linux.

El sistema Linux identifica los dispositivos hardware como archivos especiales llamados archivos de dispositivo (device files). Los archivos de dispositivo se clasifican en tres tipos:

- Carácter: son utilizados para dispositivos que solo pueden manejar un carácter a la vez. Muchos tipos de modems y terminales son creados como archivos de carácter.
- Bloque: los archivos de tipo bloque son para dispositivos que pueden manejar bloques largos de datos al mismo tiempo, tales como unidades de disco.
- Red: los archivos de tipo red son usados por dispositivos que utilizan paquetes para enviar y recibir datos. Estos incluyen las tarjetas de red, y un dispositivo especial loopback que le permite al sistema Linux comunicarse consigo mismo usando protocolos de red comunes como TCP/IP.

4. Administración del sistema de archivos

A diferencia de otros sistemas operativos, el kernel de Linux puede soportar diferentes tipos de sistemas de archivos para realizar las tareas típicas de lectura y escritura de datos. Además de tener mas de 12 sistemas de archivos nativos, Linux puede leer y escribir desde y hacia sistemas de archivos usados por otros sistemas operativos, tales como Microsoft Windows. La tabla 3.1 lista los sistemas de archivos estándares que un sistema Linux puede utilizar para leer y escribir datos.

Sistema de Archivos	Descripción
ext	Linux extended filesystem – Sistema de archivos original de Linux.
ext2	Second extended filesystem, provee caracterisitcas avanzadas sobre ext.
ext3	Third extended filesystem, soporte de journaling.
ext4	Fourth extended filesystem, soporte de journaling Avanzado.
hpfs	OS/2 high-performance filesystem.
Jfs	IBM's journaling filesystem.
iso9660	Sistema de archivos ISO 9660 (CD-ROMs).
minix	Sistema de archivos de MINIX.
msdos	Microsoft FAT32.
ncp	Netware filesystem.
nfs	Network File System.
ntfs	Soporte para sistemas de archivos Microsoft NT.
proc	Acceso a información del sistema.
ReiserFS	Sistema de achivos avanzado de Linux para recuperación y rendimiento mejorado.
smb	Sistema de achivos Samba SMB para acceso en red.
sysv	Viejo sistema de achivos en sistemas Unix.
ufs	Sistema de achivos de BSD.
umsdos	Sistema de achivo tipo Unix que recide sobre sistemas msdos.
Vfat	Sistema de achivos de Windows 95(FAT32).
XFS	Sistema de archivos journaling de 64 bits de alto rendimiento.

Tabla 3.1 Sistemas de archivos en Linux

Distribuciones de Linux

Ahora que ya conoce los cuatro componentes principales de un sistema Linux, puede ser que usted mismo se pregunte ¿Cómo junto todas las partes para formar un sistema Linux completo? Afortunadamente otras personas ya han hecho eso para usted.

Un paquete de un sistema Linux completo es llamado distribución. Existen muchas distribuciones de Linux que cumplen cualquier requerimiento computacional que usted pueda tener. La mayoría de distribuciones están personalizadas para un grupo específico de usuarios, tales como usuarios de negocios, entusiastas de multimedia, desarrolladores de software o usuarios domésticos. Cada distribución personalizada incluye el paquete de software requerido para soportar funciones especializadas, tales como software de edición de audio y video para entusiastas de multimedia, o compiladores y ambientes de desarrollo integrado (IDE) para desarrolladores de software.

Frecuentemente las distintas distribuciones de Linux se dividen en tres categorías:

- Distribuciones completas de Linux
- Distribuciones especializadas de Linux
- Distribuciones de prueba LiveCD

La siguiente sección describe estos tipos de distribuciones de Linux y muestra algunos ejemplos de distribuciones de Linux en cada categoría.

Distribuciones completas de Linux

Una distribución completa de Linux contiene un kernel, uno o más ambientes de escritorio gráfico y muchas aplicaciones de usuario, precompiladas para el kernel. La tabla 3.2 muestra algunas de las distribuciones completas de Linux más populares.

Distribución	Descripción
Slackware	Una de las distribuciones originales de Linux, popular dentro de la comunidad geek de Linux
Red Hat	Distribución comercial para negocios usada principalmente para servidores de Internet
Fedora	Un subproducto de Red Hat pero diseñado para usuarios domésticos
Gentoo	Una distribución diseñada para usuarios avanzados, contiene solamente el código fuente de Linux
openSUSE	Distribución para distintos usos, utilizada en ambientes comerciales como servidores y domésticos.
Debian	Popular entre los usuarios expertos de Linux y productos comerciales

Tabla 3.2 Distribuciones completas de Linux

En los primeros días de Linux, una distribución era liberada como un set de discos floppy. Se descargaba un grupo de archivos y se copiaban en el set de discos. Comúnmente se requerían 20 discos para hacer una distribución completa, no es necesario decirlo, pero esto era una experiencia dolorosa.

Hoy en día, con computadoras de escritorio comúnmente utilizan dispositivos USB 3.0 y en algunos casos unidades de CD/DVD, las distribuciones de Linux son liberadas en formato ISO para ser copiada en CD/DVD y unidades de almacenamiento USB 3.0. Esto hace que instalar Linux sea una tarea mucho más fácil. Sin embargo, los usuarios principiantes pueden enfrentan algunos problemas al instalar distribuciones completas de Linux. Para cubrir cualquier necesidad de un usuario en particular, una distribución debe incluir gran cantidad de software, desde sistemas de base de datos hasta aplicaciones de juego de usuario final, es por eso que una distribución completa normalmente requiere de varias unidades de CD/DVD o en su defecto varios Gigabytes de almacenamiento.

Aunque tener gran cantidad de distribuciones disponibles es importante para usuarios avanzados de Linux, puede ser una pesadilla para usuarios principiantes. La mayoría de distribuciones hacen una serie de preguntas durante el proceso de instalación para determinar que aplicaciones se deben cargar por defecto, que hardware está conectado a la PC y como configurarlo. Los principiantes a menudo se ven confundidos por estas preguntas. Como resultado, frecuentemente sobrecargan el sistema de software que nunca utilizaran.

Afortunadamente par los principiantes, existen formas más simples de instalar Linux.

Distribuciones especializadas de Linux

En los últimos años han aparecido un nuevo subgrupo de distribuciones Linux. Típicamente están basadas en una de las distribuciones principales, pero solamente contienen un subset de aplicaciones, esto con el objetivo de cumplir necesidades en áreas específicas de uso.

Además de proveer software especializado (tales como productos de oficina para usuarios de negocios), las distribuciones personalizadas de Linux también intentan ayudar a los usuarios principiantes a través de la autodetección y autoconfiguración de los dispositivos hardware más comunes. Esto hace que la instalación de Linux sea un proceso mucho más amigable.

La tabla 3.3 muestra algunos de las distribuciones especializadas de Linux más utilizadas, así como su uso más común:

Distribución	Descripción
Centos	Distribución libre creada a partir del código fuente de Red Hat Enterprise Linux
ubuntu	Distribución libre dirigida para el sector educativo y usuarios domésticos
PCLinuxOs	Distribución libre para usuarios de oficina y domésticos.
mint	Distribución libre para entrenamiento en el hogar.
Dyne:bolic	Distribución libre diseñada para audio y aplicaciones MIDI
Puppy linux	Distribución libre muy ligera que corre eficientemente en PCs con hardware antiguo.

Tabla 3.3 Distribuciones especializadas de Linux

En la tabla anterior solamente se muestran algunas de las distribuciones especializadas más comunes, literalmente hay cientos de distribuciones especializadas de Linux, y con el paso del tiempo su número va en incremento. No importa cuál sea su especialidad, es muy probable que exista una distribución específica para su campo de estudio.

Muchas de las distribuciones especializadas de Linux están basadas en Debian, la utilizada en este curso no es la excepción (Ubuntu). Ubuntu utiliza los mismos archivos de instalación de Debian, pero los paquetes base del sistema son solo una pequeña fracción del utilizado por una versión completa de Linux como lo es Debian.

Distribuciones de prueba LiveCD

Las distribuciones Linux de arranque Live CD son un fenómeno relativamente nuevo en el mundo de Linux. Estas permiten ver y explorar un sistema Linux sin necesidad de instalarlo. La mayoría de las PCs modernas permiten el arranque desde un CD/USB en lugar del disco duro. Para tomar ventaja de esta característica, algunas distribuciones Linux crean una imagen de arranque que contiene una copia del sistema Linux, a esto se le llama Linux LiveCD. Debido a las limitaciones en tamaño de una unidad de CD, la imagen no puede contener una copia completa del sistema Linux, pero es realmente sorprendente la cantidad de software que puede ofrecer. El resultado es que usted puede arrancar su PC desde una distribución de Linux LiveCD sin tener que instalar el sistema completo en su disco duro.

Esta es una forma excelente de probar varias distribuciones de Linux sin necesidad de reescribir el disco principal de su PC. Usted puede descargar gran cantidad de distribuciones LiveCDs desde

Internet, posteriormente copiarlas en una unidad CD/USB y finalmente probarla en su equipo de trabajo.

La tabla 3.4 muestra algunas de las distribuciones Linux LiveCD disponibles:

Distribución	Descripción
knoppix	Distribución alemana, el primer Linux LiveCD desarrollado
PClinuxos	Distribución completa de Linux disponible en LiveCD
Ubuntu	Un proyecto mundial de Linux, diseñada para muchos lenguajes
slax	Linux liveCD basada en Slackware Linux
Puppy linux	Distribución libre muy completa diseñada para correr en PCs con hardware antiguo

Tabla 3.4 Distribuciones Linux LiveCD

La tabla anterior le puede ser muy parecida con la tabla 3.3, muchas distribuciones especializadas de Linux también cuentan con una versión en LiveCD. Algunas distribuciones Linux LiveCD como Ubuntu, permiten instalar la distribución directamente desde el disco LiveCD. Esto le permite arrancar con el CD, probar la distribución Linux, y si lo desea instalarla en el disco duro. Esta es una característica muy practica y amigable al usuario.

En las distribuciones de prueba LiveCD no todo son ventajas, tambien tienen algunos inconvenientes. Debido a que todo se carga desde el CD, las aplicaciones pueden correr más lento, especialmente si está usando un equipo viejo con unidades de CD lentas. También, debido a que no se puede escribir en la unidad de CD, cualquier cambio hecho por Linux será ignorado en el próximo reinicio del sistema.

Muchos avances se han hecho en el mundo de los LiveCD de Linux que pueden ayudar a solventar algunos de estos problemas. Estos avances incluyen la habilidad para:

- Copiar archivos del sistema desde el CD hacia la memoria.
- Copiar archivos del sistema a un archivo en el disco duro.
- Almacenar configuraciones del sistema en una unidad de memoria USB
- Almacenar configuraciones de usuario en una unidad de memoria USB

Algunas distribuciones LiveCDs como Puppy Linux son diseñadas con un número mínimo de archivos de sistema. Los scripts de arranque del disco LiveCD copian los archivos directamente a la memoria. Esto permite quitar el CD/USB de la computadora tan pronto como inicie el sistema. Como

consecuencia el sistema correrá las aplicaciones mucho más rápido (las aplicaciones corren más rápido desde la memoria física).

Otras distribuciones LiveCD usan un método alternativo que permite remover el CD de la bandeja después del arranque. Esto implica copiar los archivos principales de Linux hacia un archivo ubicado en el disco duro. Después del arranque del CD, el sistema lee el archivo guardado en el disco duro. La versión dyne:bolic usa esta técnica, la cual es llamada docking. Por supuesto, antes de arrancar desde el CD deberá copiar el archivo de sistema al disco duro.

Es importante señalar que las técnicas antes mencionadas son aplicadas a unidades de almacenamiento USB, con la ventaja de que el Linux LiveCD puede escribir directamente sobre ellas.

El Shell de Linux

En los primeros años de Linux, todo trabajo con el sistema era a través del Shell. Administradores de sistemas, programadores y usuarios, todos usaban algo llamado terminal de consola Linux, escribiendo comandos del shell y viendo salidas de texto. Hoy en día, con el ambiente de escritorio gráfico, se ha vuelto difícil encontrar un prompt del shell para escribir comandos. Esta sección discute que se requiere para poder acceder al ambiente de línea de comandos.

La línea de comandos

Antes de que se popularizaran los escritorios gráficos, la única forma de interactuar con un sistema Unix era a través de la interfaz de línea de comandos (CLI) proveída por el Shell. El CLI permitía solamente texto de entrada y solo podía desplegar texto y gráficos rudimentarios como salida.

Debido a estas restricciones, los dispositivos de salida no eran muy atractivos. Con frecuencia, solo se necesitaba de una terminal tonta para interactuar con el sistema Unix. Una terminal tonta no era mas que un monitor y un teclado conectado al sistema vía cables de comunicaciones (cable serie). Esta combinación simple proveía a Unix de una manera fácil de escribir texto y ver los resultados.

Como bien puede saber, en los ambientes Linux de hoy en día las cosas pueden ser muy diferentes. Casi todas las distribuciones Linux usan algún tipo ambiente de escritorio gráfico. Sin embargo, para escribir comandos del Shell, todavía se necesita tener acceso al CLI. El problema se

ha simplificado a un punto antes impensable, a veces encontrar una manera para conseguir un CLI en una distribución Linux no es una tarea fácil.

Accediendo a terminales de comando

Terminales de consola

Una manera de obtener una CLI es tener Linux fuera del modo gráfico de escritorio, estableciendo el modo texto. Este modo provee únicamente un Shell CLI, como se hacía antes que aparecieran los escritorios gráficos. Este modo es llamado consola de Linux porque emula los viejos tiempos de Linux, los días cuando se accedía a la consola a través del cable serie.

Cuando arranca el sistema, automáticamente se crean varias consolas virtuales. Una consola virtual es una sesión de terminal que corre en la memoria del sistema Linux. En lugar de tener muchas terminales tontas conectadas al equipo, la mayoría de distribuciones de Linux inician cinco o seis (a veces más) consolas virtuales que pueden ser accedidas utilizando el teclado y el monitor del equipo. Normalmente se pueden acceder las terminales de consola con las combinaciones del teclado `Alt+F1` | `Alt+F2` | `Alt+F3` | `Alt+F4` | `Alt+F5`.

Terminales gráficas

La alternativa a utilizar una terminal virtual de consola es usar un paquete de emulación de terminal, dentro del entorno de escritorio gráfico de Linux. Un paquete de emulación de terminal simula trabajar sobre una terminal de consola, pero dentro de una ventana gráfica de escritorio. La figura 3.3

muestra un ejemplo de un emulador de terminal corriendo sobre entorno de escritorio gráfico de Linux.

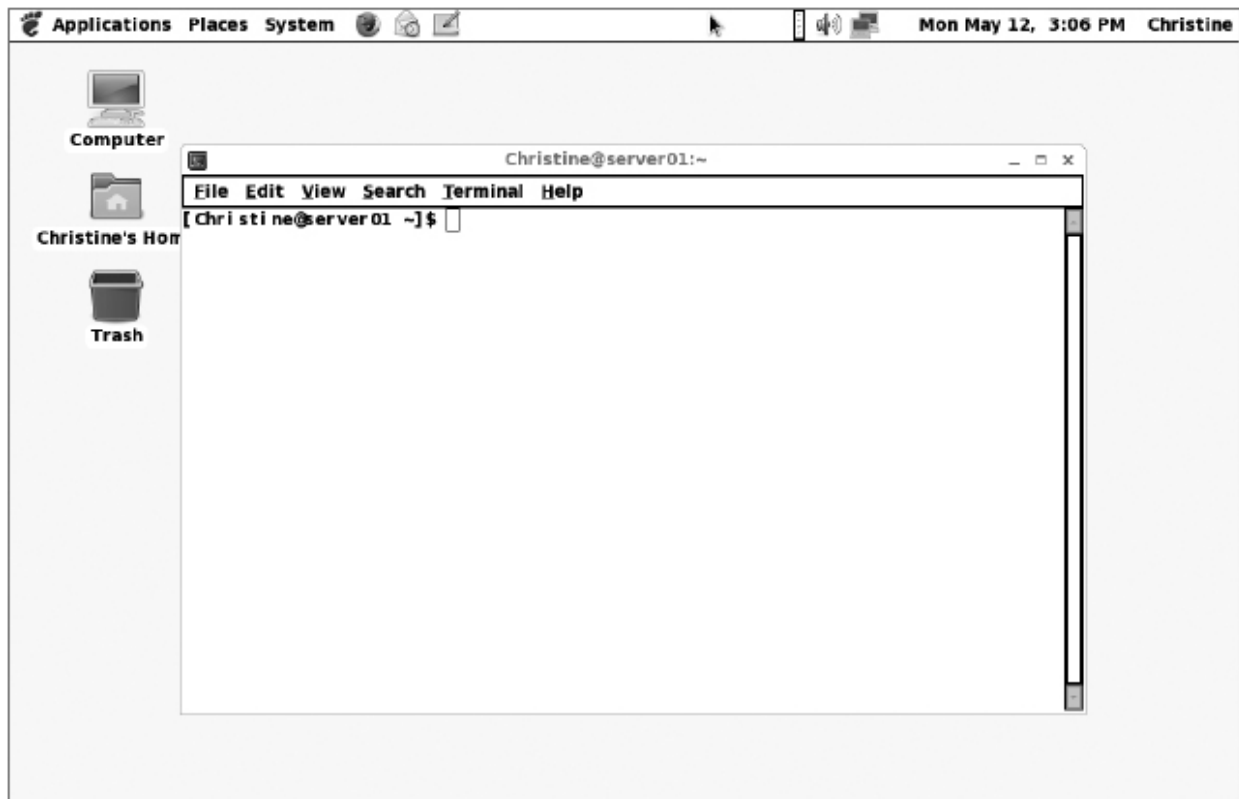


Figura 3.3 emulador de terminal que se ejecuta en un escritorio gráfico de Linux

La emulación de terminal gráfica es solamente responsable de la experiencia gráfica. El entorno completo está compuesto por varios componentes, incluyendo el software de emulación gráfica de terminales (llamado cliente). La tabla 3.5 muestra los distintos componentes del entorno de escritorio gráfico de Linux.

Nombre	Ejemplos	Descripción
Client	Emulador de terminal gráfica, ambiente de escritorio	Aplicación que solicita servicios gráficos
Display Server	Mir, Wayland compositor, Xserver	Elemento que administra la pantalla y los dispositivos de salida como el teclado y el mouse
Windows manager	Compiz, Metacity, Kwin	Elemento que agrega bordes a Windows y proporciona características para mover y administrar ventanas
Widgets library	Athena(Xaw), X Intrinsic	Elemento que agrega menús y otros ítems al cliente de escritorio.

Tabla 3.5 Elementos de la interfaz gráfica

Para comprender el funcionamiento de la línea de comandos del escritorio, usted debe enfocarse sobre el emulador de terminales gráficas. Puede relacionar los emuladores de terminales gráficas con las terminales CLI dentro del entorno GUI (Graphical User Interface) y las terminales de consola virtual con las terminales CLI que

están fuera del entorno GUI. Entender las múltiples terminales existentes y sus características puede mejorar su experiencia de usuario sobre la línea de comandos.

- ¿Qué es Linux?
- El kernel de Linux
- Distribuciones de Linux
- El Shell de Linux
- Accediendo a terminales de comandos

Bibliografía:

Richard Blum, Christine Bresnahan. (January 2015). Linux Command Line and Shell Scripting Bible. Indianapolis, IN 46256, United States: Wiley.

Rob Kennedy. (Noviembre 2018). Essential Linux Command Line. United States: Kennedy Projects .