



UD 05.

INTRODUCCIÓN A LINUX

Sistemas Informáticos
DAW CFGS

Álvaro Maceda

a.macedaarranz@edu.gva.es

2022/2023

Versión:221025.1701

Licencia



Reconocimiento - No comercial - ShareAlike (by-nc-sa): No se permite el uso comercial de la obra original ni de ninguna obra derivada, cuya distribución debe realizarse bajo una licencia igual a la que rige la obra original.

Nomenclatura

A lo largo de esta unidad se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

ÿ Importante

ÿ Atención

ÿ Interesante

TABLA DE CONTENIDO

1. ¿Qué es Linux?	4	1.1 GNU/
Linux.....	4	1.2 ¿Qué es una
distribución?.....	5	1.3 ¿Dónde encontramos
Linux?.....	6	2. Modos de consola y
gráfico	6	3. El
terminal	7	3.1 Consolas
virtuales	7	3.2 La
concha.....	8	3.3 Introducción a la
consola.....	9	3.4 Comandos básicos de
Bash	11	4 El modo
gráfico	13	4.1 La pantalla de Linux
sistema.....	13	4.2 Servidores de
visualización	14	4.3 El gestor de
ventanas	15	4.4 Entornos de
escritorio.....	15	4.5 El administrador de
pantalla.. ..	17	5.
Registros	17	6. Complementario
material	19	

UT 05. INTRODUCCIÓN A LINUX

1.¿QUÉ ES LINUX?

La forma más sencilla de definir Linux es que es un sistema operativo de estilo Unix. A estas alturas del curso escolar ya sabemos que es un sistema operativo, pero ¿qué es Unix?

Aunque no es el primer sistema operativo, Unix es, sin duda, el primer gran sistema operativo. Sus características más importantes fueron que es portátil (puede ejecutarse en diferentes sistemas informáticos) multitarea y multiusuario. Además, arquitectónicamente hablando, se creó en base a conceptos como la simplicidad y la modularidad para que el código fuera fácilmente mantenido y ampliado por otros.

programadores

A lo largo de los años, la empresa creadora (Bell Labs¹) fue licenciando el producto a otras empresas que, para adaptarlo a entornos más específicos, iban realizando modificaciones y creando diferentes versiones. De ahí nacieron productos como Xenix (Microsoft), HP-UX (HP), IRIX (Silicon Graphics), SCO (Novell), AIX (IBM)....

• Todos los sistemas de la familia UNIX a menudo se denominan *IX

El problema de todos ellos es que son versiones propietarias por lo que el código no está disponible para su estudio (aparte de la gran cantidad de dinero que cuesta una licencia). Por eso, con un objetivo puramente educativo, a finales de los 80, un profesor de la Universidad de Ámsterdam decidió crear Minix, un sistema operativo basado en la filosofía UNIX pero reescrito desde cero y de código abierto. Debido a su carácter educativo, el autor decidió no permitir modificaciones que muy probablemente complicarían mucho más el código.

Es en este momento cuando un estudiante finlandés de informática decide, basándose en Minix, crear un clon libre que funcione en sistemas PC. Este estudiante fue Linus Torvalds² y llamó a su sistema operativo Linux.

1.1 GNU/Linux

En el apartado anterior comentábamos que Linux era un sistema operativo, pero eso no es exactamente así. Todo sistema operativo consta de un kernel o núcleo y un conjunto de aplicaciones que ayudan a hacer posible el funcionamiento del sistema operativo. Una posible clasificación de los diferentes programas que acompañan al kernel en un sistema operativo podría ser: shell o terminal (que permite la interacción con el usuario en modo texto), servicios o daemons (que son programas que se ejecutan en segundo plano), servidor (que permite dibujar elementos en pantalla) o un escritorio (que aprovecha las funciones del servidor gráfico para proporcionar acceso gráfico al usuario).

• Linux es simplemente el kernel del sistema. Para formar el sistema operativo, se acompaña de muchas utilidades GNU³. Ese sistema operativo no se llama Linux, se llama GNU/Linux

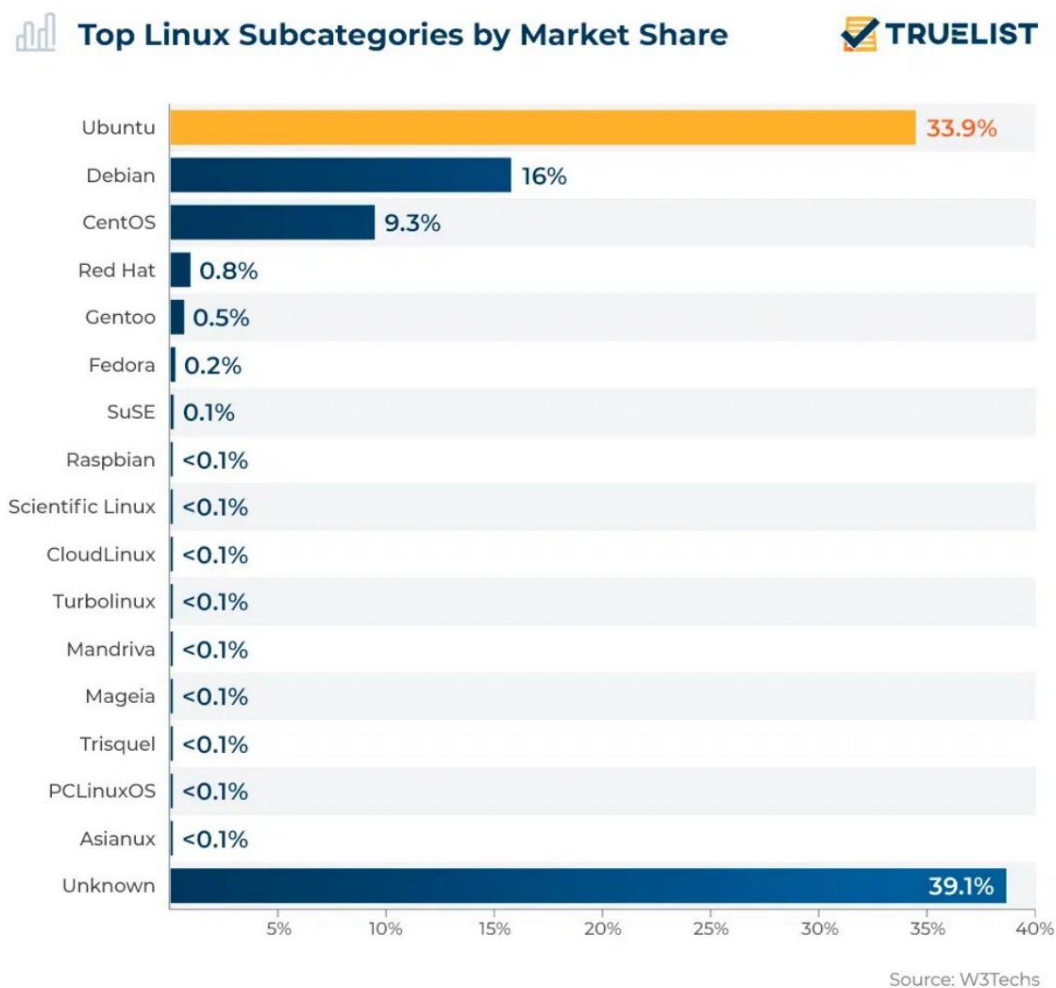
¹ Fundado por el inventor del teléfono, Graham Bell ² <http://www.comunidadhosting.com/t/primer-mensaje-de-linus-torvalds-y-comienzo-de-linux.8904/> ³ GNU es una Proyecto Linux cuyo objetivo es crear un sistema operativo libre. El proyecto ha desarrollado muchos de los

¿ Todos los días, todo el mundo simplifica y llama a Linux el sistema operativo (erróneamente).

1.2 ¿Qué es una distribución?

El kernel y las utilidades básicas conforman el núcleo del sistema operativo, pero sabemos que hoy en día se lanza un sistema operativo con mucho software no necesario para el propio ordenador (programas de dibujo, editores de texto, etc.). El hecho de que el kernel y las utilidades básicas se distribuyan libremente permite que cualquier persona pueda tomar estos elementos y acompañarlos de otras utilidades (básicas o no) según su gusto o sus necesidades. Así surgen las distribuciones de Linux o, más comúnmente, las distros.

Hay cientos de ellos en el mercado, pero por mencionar algunos de los más conocidos: Ubuntu (y sus diferentes "flavors" como LUbuntu, KUbuntu, Ubuntu Server...), Open Suse, Arch, Fedora, Debian, Red Hat, Mint, Lliurex.



Estadísticas sobre las distribuciones de Linux más utilizadas. Fuente: truelist.co

utilidades que acompañan al kernel, pero a día de hoy no han conseguido tener un kernel lo suficientemente estable.

1.3 ¿Dónde encontramos Linux?

Una de las ideas más extendidas es que Unix es un sistema operativo que sólo se utiliza en entornos académicos o de alto nivel técnico, y nada más lejos de la realidad. Los sistemas basados en versiones libres de Unix están implantados en muchos sistemas informáticos. Aunque posiblemente en entornos de escritorio los sistemas de Microsoft siguen siendo los que dominan la mayor parte del mercado, tanto en dispositivos móviles como en servidores la realidad es bien distinta.

Por ejemplo, los dos sistemas operativos móviles por excelencia (Android e iOS), son sistemas derivados de Linux o FreeBSD (además de sistemas como MacOS).

2. MODOS DE CONSOLA Y GRÁFICOS

Necesitamos una manera de interactuar con una computadora. En los sistemas de escritorio, lo habitual es interactuar con el sistema mediante una interfaz gráfica. Sin embargo, a veces no lo necesitamos o no podemos usar una interfaz gráfica: por ejemplo, para computadoras servidor. Linux permite usar ambos tipos de interfaces: podemos configurar cuál se iniciará cuando arranque el sistema, y también instalar o quitar la interfaz gráfica del equipo.

Para cambiar entre el modo de gráficos y el de consola, podemos usar *systemd*. Muchas distribuciones de Linux usan *systemd* para administrar la configuración y los servicios del sistema. Decide qué tareas ejecutará el sistema al arrancar (unidades) y las organiza en objetivos. Un destino es un conjunto de unidades con dependencias entre ellas. Los objetivos más importantes de *systemd* son:

- **default.target**: el destino que se inicia de forma predeterminada. Este es un enlace simbólico a otro objetivo, como **graphical.target**.
- **Emergency.target**: inicia un shell de emergencia en la consola.
- **graphical.target**: inicia un sistema con red, soporte multiusuario y una pantalla gerente.
- **halt.target**: cierra el sistema.
- **multiusuario.objetivo**: inicia un sistema multiusuario con conexión en red y sin gráficos ambiente.
- **reboot.target**: reinicia el sistema.
- **Rescue.target**: inicia un sistema de un solo usuario sin red.

Veremos esto con más detalle en una unidad posterior. El punto importante aquí es que si ejecutamos **graphical.target** terminaremos con un entorno de gráficos, y si ejecutamos **multi-user.target** tendremos un entorno de texto con una consola para interactuar con el sistema.

Puede ver cuál es el objetivo actual con:

```
systemctl get-default
```

3.EL TERMINAL

Las personas necesitan una interfaz para enviar información a la computadora y leer la salida de ella. Los dispositivos de entrada que usamos hoy en día incluyen teclados, ratones, micrófonos y cámaras web. Los dispositivos de salida suelen ser una o varias pantallas y un altavoz. Esos se llaman dispositivos periféricos, pero en el mundo de Unix y Linux se llaman terminales.

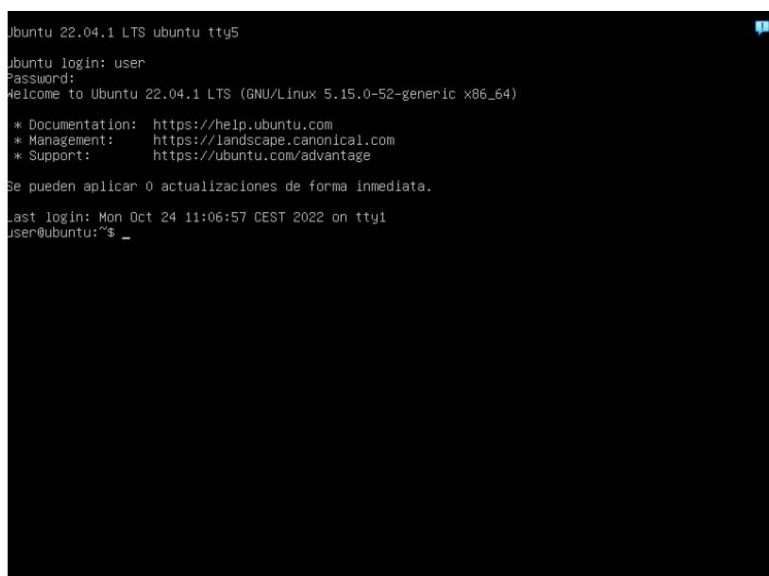
Una *terminal* se refiere a un dispositivo electrónico con un monitor y un teclado que se utiliza para interactuar con una computadora. Este dispositivo es un tipo de terminal físico en el que los programas están activos en modo texto, un lugar desde el que se monitoriza y controla un sistema. Hoy en día existen las llamadas consolas virtuales, lo que significa que la consola puede ser completamente software. En términos prácticos, no se hace diferencia entre consola y terminal y utilizaremos ambos términos indistintamente.

La consola, entonces, emula un teclado y un monitor conectados directamente al sistema. Esto tomaría el lugar de una terminal serial en sistemas Unix de minicomputadora y mainframe.

3.1 Consolas virtuales

Una consola virtual es como tener un emulador de terminal Linux en pantalla completa. En los sistemas de escritorio, con frecuencia verá mensajes de inicio antes de que se inicie el administrador de pantalla. Cuando ejecuta el sistema en modo multiusuario (modo de texto), arranca en una consola virtual. Cuando el sistema se está iniciando, los mensajes de inicio se muestran en una consola virtual.

Hay varias consolas virtuales disponibles. Para cambiar entre las consolas virtuales, puede usar el sistema Alt + tecla de función. Por lo general, hay siete consolas virtuales que puede usar. Si está ejecutando un sistema de escritorio con X11, generalmente comienza en la séptima consola virtual. Para cambiar a la primera consola virtual, deberá presionar Alt + F1. Si no funciona para usted, intente Ctrl + Alt + F1 en su lugar. Para volver al sistema de escritorio, puede presionar Alt + F7. Si está ejecutando el sistema operativo dentro de una máquina Virtual Box, puede cambiar entre consolas virtuales presionando [la tecla host + Fn](#).

A screenshot of a virtual terminal window. The title bar at the top reads "ubuntu 22.04.1 LTS ubuntu tty5". The terminal content shows the login process: "ubuntu login: user", "Password:", and "Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)". It then lists links for documentation, management, and support. Below that, it says "Se pueden aplicar 0 actualizaciones de forma inmediata." and "Last login: Mon Oct 24 11:06:57 CEST 2022 on tty1". The prompt "user@ubuntu:~\$" is visible at the bottom.

Ejemplo de una consola virtual dentro de una máquina virtual, con un usuario que acaba de iniciar sesión. Puede ver que esta es la quinta consola virtual porque hay un mensaje "ubuntu tty5" en la parte superior de la pantalla

Si bien las consolas virtuales se usan con menos frecuencia ahora que los escritorios gráficos con administradores de pantalla son más comunes, todavía tienen sus usos en Linux en la actualidad. Si su sesión de escritorio se bloquea o algo más sale mal, puede cambiar a una consola virtual e iniciar sesión para tratar de solucionarlo, o puede apagar o reiniciar su máquina sin problemas.

Si la sesión de escritorio no se inicia, de todos modos se le volcará a una sesión de consola virtual. Esta es otra razón más por la que vale la pena estar familiarizado con la línea de comandos de Linux.

Las consolas virtuales todavía se usan en algunos programas de instalación de distribución, en particular los que se ejecutan en modo de texto, como las distribuciones de servidor. Puede cambiar a otra consola virtual y ver cómo funciona el va la instalación o por qué una operación parece estar tardando mucho. También puede ejecutar otros comandos en un shell que pueden ser útiles si se encuentra con algún error.

3.2 La concha

El *shell* es la interfaz o programa que interpreta las líneas de comando enviadas a la terminal. Funciona procesando los comandos e interpretando los resultados, es decir, toma los comandos escritos a mano y le dice al sistema operativo que debe ejecutarlos. Cuando está ejecutando una terminal, o está en una consola virtual, lo que realmente ve es la salida del shell. Los comandos que ingresa se envían al shell, que los interpreta. También toma secuencias de comandos para interpretar la lógica entre ellos. Su función principal es iniciar otros programas.

Por ejemplo: si abre una consola virtual y ejecuta el comando `whoami`, verá su nombre de usuario como respuesta. Lo que realmente sucede cuando haces eso es:

1. El shell recibe el comando (`whoami`) y lo interpreta como un comando que debe ser ejecutado
2. El shell ejecuta el programa `whoami`, redirigiendo la entrada del usuario a ese programa (en este caso, el programa no necesita una entrada) y la salida a la pantalla.
3. Una vez finalizado el programa, espera el siguiente comando.

En los sistemas Linux, el shell es configurable. El administrador del sistema puede instalar nuevos shells y configurar el shell predeterminado para cada usuario.

Hay cientos de shells disponibles para Linux. No obstante, los más utilizados son:

- **bash** (*Bourne-again shell*): es el shell de inicio de sesión predeterminado en la mayoría de los sistemas. Tiene características como un historial de comandos, alias de comandos, expansión, control de trabajos, etc.
- **ksh** (*Korn Shell*): Fue desarrollado muchos años antes que bash. Se ejecuta a mayor velocidad y tiene mejor soporte para algunas estructuras de programación como bucles o matrices asociativas.
- **zsh** (*Z-shell*): Tiene muchas similitudes con bash y ksh e incorpora muchas de sus características principales. Pero también incorpora muchas mejoras como cambiar el directorio sin entrar en `cd`, autocompletado y corrección ortográfica, y soporte de plugins y temas.
- **dash** (*Debian Almquist Shell*): Es un shell diseñado para ser un shell minimalista con una gran velocidad de ejecución. Es pequeño y estable, y se usa como shell predeterminado en algunos sistemas.

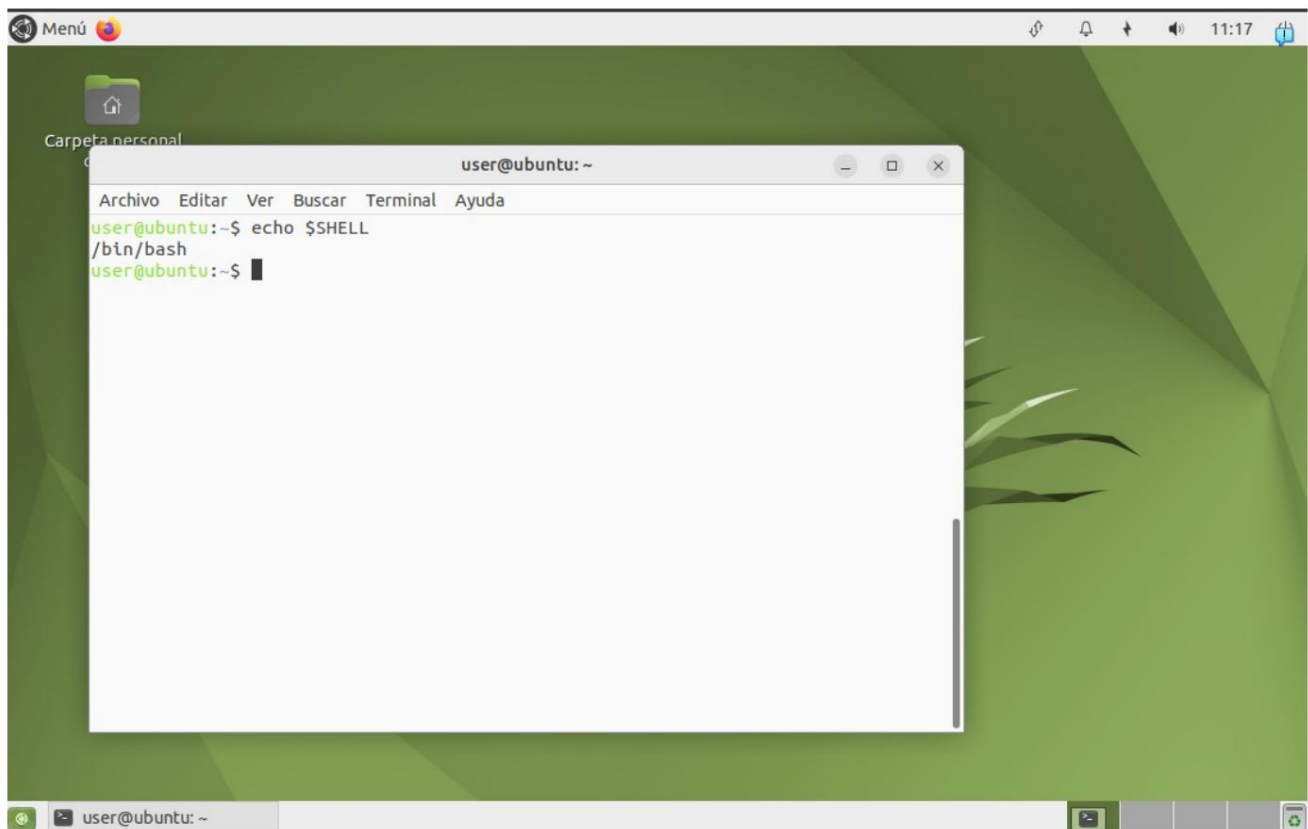
En la mayoría de los sistemas, `sh` es el shell del sistema predeterminado. Pero este comando apunta a otro shell, generalmente uno liviano como dash. Para saber qué shell está utilizando, puede ejecutar el comando:


```
echo $SHELL
```

El administrador del sistema puede cambiar el shell predeterminado para un usuario con el comando:

```
usermod --shell /bin/ksh miusuario
```

Esto cambiaría el shell predeterminado de myuser a ksh. Algunos usuarios no tienen un shell, porque no deberían acceder al sistema para ingresar comandos. Suelen ser usuarios que ejecutan servicios del sistema.



Un shell que se ejecuta en un entorno gráfico dentro de un terminal virtual, una aplicación que se ejecuta como una consola virtual o un Terminal real conectado al sistema. El aspecto es similar a la terminal virtual porque ambas ejecutan el mismo programa: la shell (bash, en este caso)

3.3 Introducción a la consola

En la consola, ingresa comandos con argumentos. Los comandos se ejecutan y el resultado se imprime en la pantalla. Por ejemplo:

```
user@ubuntu:~$ echo "Hello, World!"
Hello, World!
user@ubuntu:~$ ls -l /home/user
total 36
drwxr-xr-x 2 user user 4096 oct 21 13:29 Descargas
drwxr-xr-x 2 user user 4096 oct 21 13:29 Documentos
drwxr-xr-x 2 user user 4096 oct 21 13:29 Escritorio
drwxr-xr-x 2 user user 4096 oct 21 13:29 Imágenes
drwxr-xr-x 2 user user 4096 oct 21 13:29 Música
drwxr-xr-x 2 user user 4096 oct 21 13:29 Plantillas
drwxr-xr-x 2 user user 4096 oct 21 13:29 Público
drwx----- 4 user user 4096 oct 21 13:29 snap
drwxr-xr-x 2 user user 4096 oct 21 13:29 Videos
user@ubuntu:~$
```

El usuario ha ejecutado dos comandos:

- `echo "¡Hola mundo!"`: El comando es `echo` y el parámetro es `"¡Hola mundo!"`
- `ls -l /home/user`: el comando es `ls`. Los parámetros son `-l` y `/home/user`

Los parámetros modifican el comportamiento de los comandos indicando información adicional (p. ej., el `"¡Hola mundo!"` del primer comando indica lo que se debe mostrar en pantalla) o seleccionando diferentes modos de comportamiento (p. ej., el `-l` del segundo comando indica que los archivos deben mostrarse en un formato determinado).

Hay dos tipos de comandos de shell:

- **Comandos no incorporados**: le dicen al shell que ejecute un programa externo y muestre el resultado.

Por ejemplo, `ls` es un comando no incorporado: el shell ejecuta el programa `/usr/bin/ls` y muestra los resultados

- **Comandos incorporados**: un comando incorporado es simplemente un comando que el shell ejecuta por sí mismo, en lugar de interpretarlo como una solicitud para cargar y ejecutar algún otro programa. Por lo general, son más rápidos (porque cargar y ejecutar un programa lleva tiempo) y pueden cambiar el estado interno del shell. Por ejemplo, el comando incorporado `cd` cambia el directorio donde se ejecuta el shell. `echo` también es un comando incorporado debido a problemas de rendimiento.

Puedes diferenciarlos usando el comando `which`

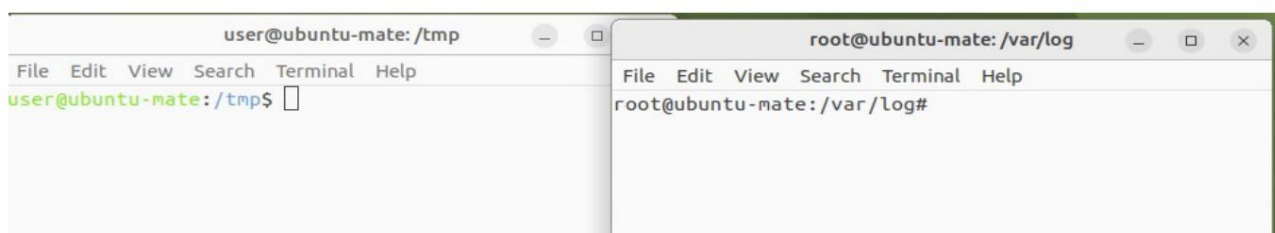
```
que gato
/usr/bin/gato

que pwd
pwd: comando integrado de shell
```

El segundo, `pwd`, es un comando incorporado.

3.3.1 El aviso

El indicador de shell es el texto que aparece antes de que ingresemos un comando. Por lo general, contiene el usuario actual, el nombre de la computadora y el directorio de trabajo actual.



Dos terminales, uno de ellos con el usuario root logueado

La mayoría de los shells permiten configurar el aviso. En bash, por ejemplo, puede configurar el aviso utilizando las variables `PS1`, `PS2` y `PS3`. Si, por ejemplo, ejecuta este comando:

```
PS1='\[\e[0m\]d\[\e[0m\]@\[\e[0m\]h\[\e[0m\]:\[\e[0m\]'
```

El aviso mostrará la fecha actual y el nombre de la computadora.

La sintaxis del indicador de shell no es trivial, pero puede usar generadores como <https://scriptim.github.io/bash-prompt-generator/> para crear un nuevo aviso.

3.4 Comandos básicos de bash

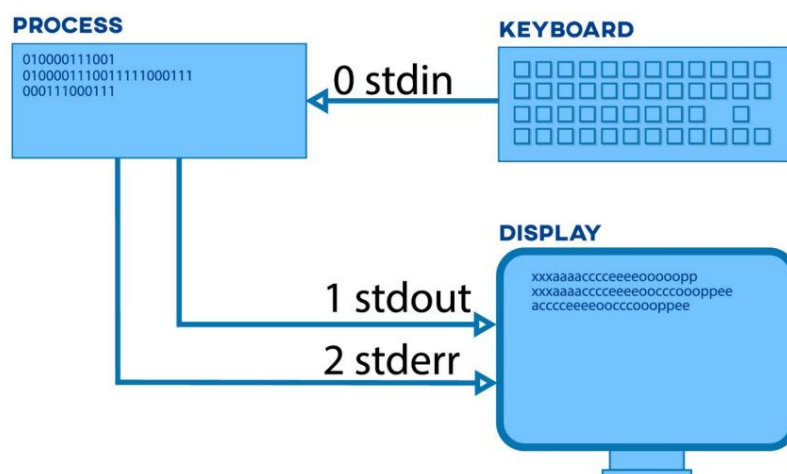
Aquí se enumeran algunos de los comandos de shell más básicos y útiles y los que se usan en los ejemplos. Los comandos para administrar el sistema de archivos los veremos más en profundidad en otra unidad:

- **pwd**: muestra el directorio actual
- **ls**: lista de archivos. Por ejemplo, **ls -l** muestra los archivos en el directorio actual
- **cd**: cambia el directorio actual del shell. Por ejemplo, **cd /** irá a la raíz del sistema. **cd /home/user** irá al directorio del usuario
- **man**: Muestra información sobre un comando: cómo funciona un comando y sus parámetros.
- **cat**: muestra el contenido de un archivo. Por ejemplo, **cat /etc/hosts**
- **menos**: muestra el contenido de un archivo pero una página a la vez. Es útil para archivos grandes. Por ejemplo, **menos /etc/grupo**
- **grep**: busca contenido en un archivo o en la entrada del programa. Por ejemplo, **el usuario grep /etc/passwd** buscará el texto "usuario" dentro del archivo **/etc/passwd**.
- **ordenar**: ordenar el contenido de un archivo. Por ejemplo, **ordene /etc/passwd**.
- **wc**: cuenta líneas y palabras de archivos. Por ejemplo, **wc -l /proc/devices** contará las líneas en el archivo **/proc/devices**

Puede obtener más información sobre los comandos que ejecutan **man <comando>**.

3.4.1 Tuberías y redirección

En Unix, los programas obtienen su entrada de la entrada estándar. Esa entrada se llama **stdin** y suele ser el teclado. La salida se envía a la salida estándar, llamada **stdout**. Si hay un error, se envía al error estándar, **stderr**. Tanto **stdout** como **stderr** son la pantalla, por defecto.



- **stdin**: toma la entrada del usuario a través del teclado. Tiene el identificador 0 dentro del proceso.
- **stdout**: muestra la salida en la pantalla. Tiene el identificador 1 dentro del proceso.

- `stderr`: Muestra información de error en la pantalla. Tiene el identificador 2 dentro del proceso.

Redirección

Podemos redirigir las entradas y salidas de un proceso a un archivo o a otro proceso. Por ejemplo, si ejecutamos esto:

```
ls 1>listado.txt
```

La salida de `ls -l` se redirigirá al archivo `listado.txt`. Entonces, si ejecutamos `cat listado.txt`, veremos el resultado del comando.

Si queremos redirigir la salida de error a un archivo, podemos usar:

```
ls "esto generará un error" 2> errores.txt
```

También podemos redirigir ambas salidas (`>` es equivalente a `1>`):

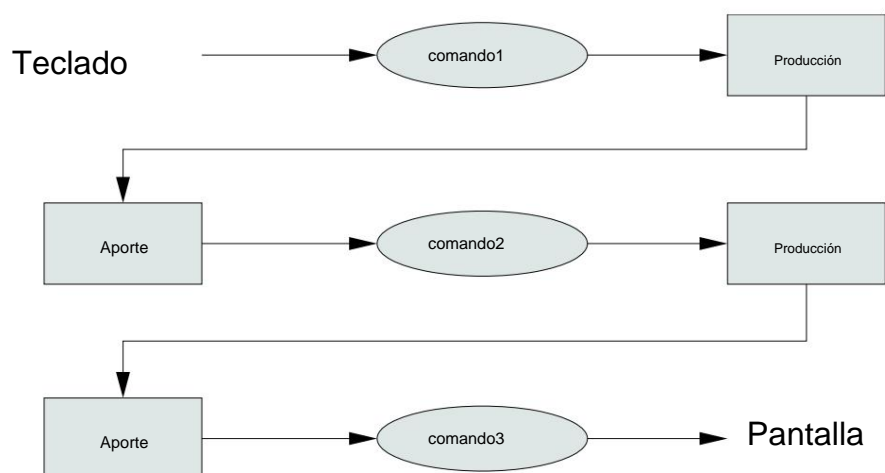
```
ls /this_is_an_error > listado.txt 2> errores.txt
```

Tubería

Las tuberías permiten redirigir la salida de un comando a la entrada de otro comando. Es como la redirección, pero enviamos la salida a otro programa en lugar de a un archivo. Hacemos eso usando el signo `|` entre mandos. Por ejemplo, si ejecutamos:

```
comando1 | comando2 | comando3
```

Esto es lo que pasa:



La filosofía de Linux es utilizar pequeños comandos que se pueden combinar para realizar tareas más complejas. Usando tuberías, podemos combinar comandos para que la salida de uno sea la entrada de otro. Por ejemplo, si ejecutamos:

```
ls -l /etc | grep 'en' | wc --líneas
```

1. Se ejecuta el comando `ls -l /etc`. Produce un listado

2. El listado se pasa como entrada a `grep`. `grep` genera solo las líneas en la lista que contiene

la cuerda en `en`

3. Las líneas producidas por `grep` se pasan como entrada a `wc`. `wc` contará el número de líneas

y mostrar el resultado en la pantalla

También podríamos haber redirigido la salida del comando anterior a un archivo agregando `> <archivo>` en el final del comando anterior.

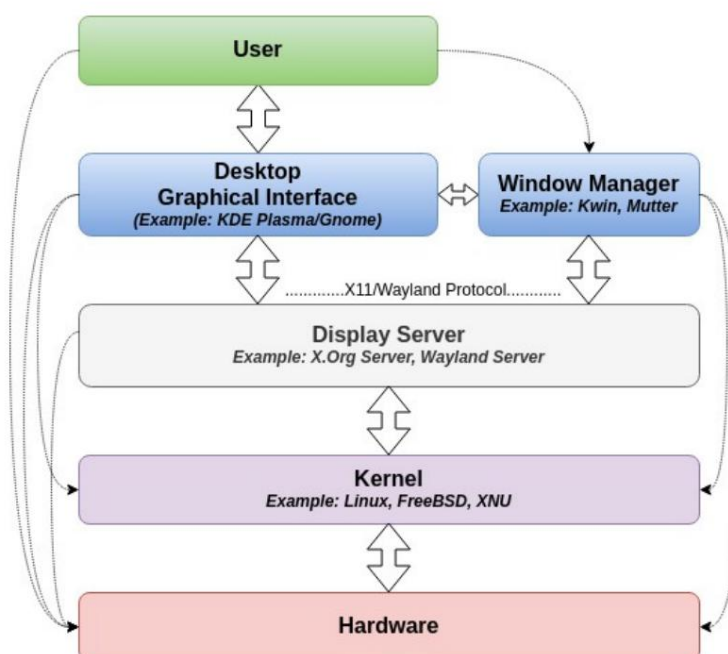
4. EL MODO GRÁFICO

Linux también puede ejecutarse en modo gráfico. Cuando usamos Linux como una máquina de escritorio, es el modo que usan casi todos los usuarios: Nos permite interactuar con el sistema de una manera fácil y cómoda.

El propósito de este tema no es enseñar cómo usar un sistema gráfico, ya que es bastante intuitivo y depende de la versión de escritorio de Linux utilizada, sino aprender sobre la arquitectura del modo gráfico.

4.1 El sistema de visualización de Linux

Esta es una descripción general de la arquitectura gráfica de Linux:



Revisaremos este diagrama desde abajo (los componentes de nivel más bajo) hasta arriba (los componentes de alto nivel).

Las capas inferiores son el hardware y el kernel. El hardware son los dispositivos que tenemos en nuestro ordenador: tarjetas gráficas, pantallas, ratón, teclado, etc. Solo se puede acceder a esos dispositivos a través del kernel. El kernel recibe los comandos del servidor de visualización, que es la capa que discutiremos a continuación.

4.2 Servidores de visualización

Un servidor de visualización es un programa cuya tarea principal es coordinar la entrada y salida de sus clientes hacia y desde el resto del sistema operativo, el hardware y entre sí. El servidor de visualización se comunica con sus clientes a través del protocolo del servidor de visualización.

Por ejemplo, los servidores de visualización administran el mouse y ayudan a hacer coincidir los movimientos del mouse con el cursor y los eventos de GUI causados por el cursor. El servidor de visualización también proporciona varios protocolos y se comunica directamente con el kernel. Existen diferentes conjuntos de protocolos de servidores de visualización y diferentes servidores de visualización que implementan un protocolo específico. En este documento hablaremos sobre el sistema X windows y Wayland.

El servidor de visualización es algo interno a las aplicaciones. El usuario probablemente no sepa qué servidor de visualización se está utilizando.

4.2.1 Sistema de ventanas X

El sistema X windows tiene dos partes: el protocolo X y la implementación X. Todas las aplicaciones y servidores que implementen el protocolo X deben ser compatibles.

El sistema de ventanas X se llama así porque es el sucesor del sistema de ventanas W.

El protocolo X11

X es un protocolo de red. Ha estado en la versión 11 desde 1987, por lo que generalmente se llama X11. Describe cómo se intercambian los mensajes entre un cliente (aplicación) y la pantalla (servidor). Estos mensajes suelen llevar comandos de dibujo primitivos como "dibujar un cuadro", "escribir estos caracteres en esta posición", "se ha hecho clic en el botón izquierdo del ratón", etc.

En X11, los clientes son las aplicaciones y el servidor es donde se dibujan las aplicaciones.

Cuando se desarrolló X11, los usuarios iniciaron un servidor X en sus escritorios (donde se visualizaron las aplicaciones) y ejecutaron los clientes X (las aplicaciones) en una computadora más potente. Actualmente, los clientes X y el servidor X normalmente se ejecutan en la misma máquina.

La implementación de X

Implementar el protocolo X11 es una tarea muy compleja. El proyecto X.Org proporciona una implementación de referencia para el protocolo que se puede integrar en los sistemas operativos.

4.2.2 Wayland

Wayland es un reemplazo para el sistema Windows X11. Su objetivo es que sea más fácil de desarrollar, ampliar y mantener. La principal diferencia entre las arquitecturas del sistema X windows y Wayland está en el papel del compositor. El compositor es el software que combina todas las ventanas para crear la imagen final que se mostrará al usuario.

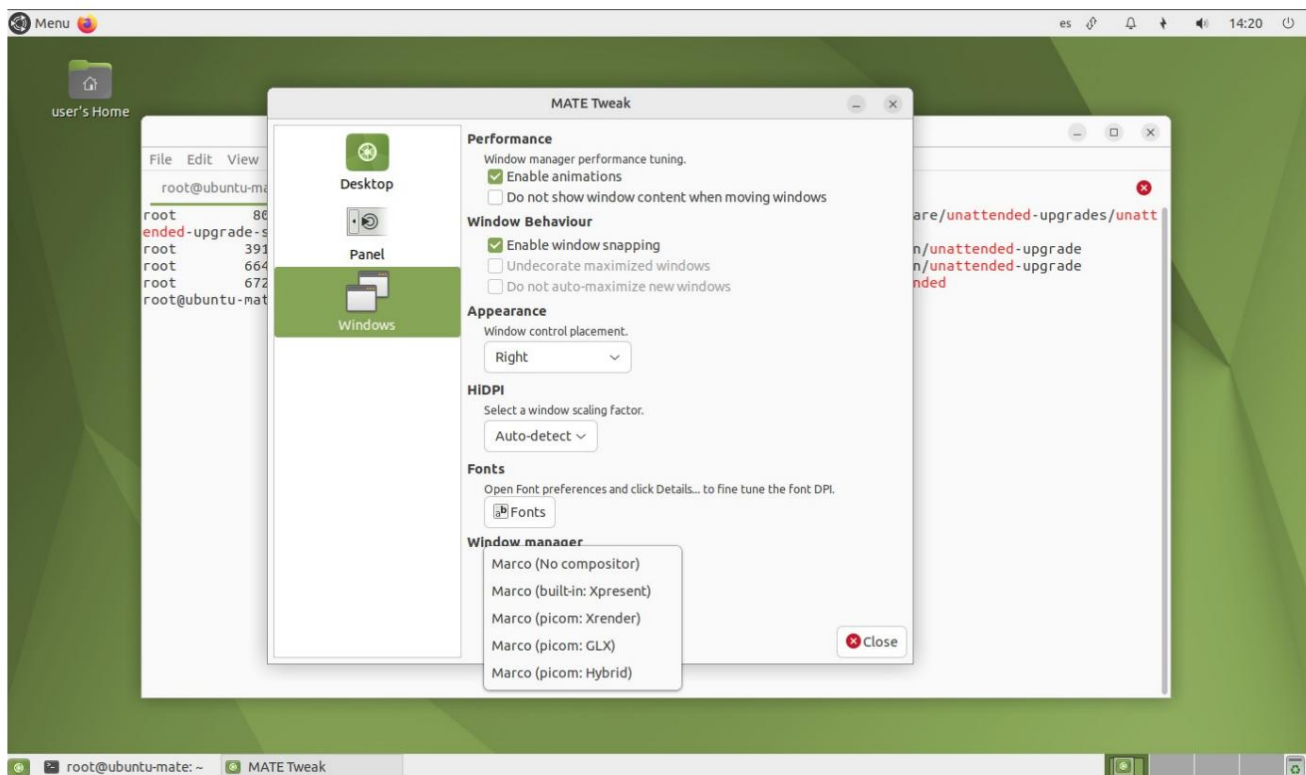
La mayoría de las distribuciones modernas de Linux han adoptado Wayland como su servidor de visualización predeterminado, pero proporciona una capa de compatibilidad con el sistema X Windows.

Si quieres saber más sobre cómo X difiere de Wayland, puedes consultar este enlace: <https://wayland.freedesktop.org/architecture.html>

4.3 El gestor de ventanas

El servidor de visualización proporciona las primitivas básicas para dibujar en la pantalla. Pero las interfaces gráficas modernas usan ventanas para mostrar aplicaciones. El administrador de ventanas controla la colocación y decoración de las ventanas. Es el gestor de ventanas el que añade a la ventana la barra de título, los botones de control —normalmente asociados a las acciones de minimizar, maximizar y cerrar— y gestiona el cambio entre ventanas abiertas.

El administrador de ventanas es utilizado por el entorno de escritorio, pero generalmente puede instalarlo y elegirlo. otro.

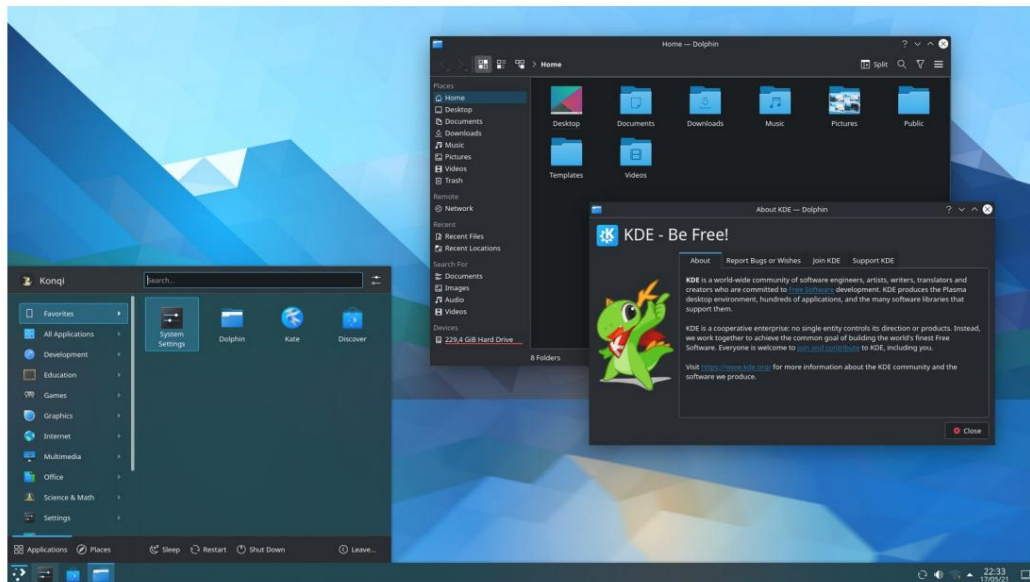


Selección de un administrador de ventanas para un entorno de escritorio Mate

4.4 Entornos de escritorio

Un entorno de escritorio es lo que ve cuando ingresa a una interfaz gráfica en los sistemas Linux. Podemos ejecutar programas individuales en un servidor de visualización, pero lo que normalmente queremos es un entorno integrado que nos permita ejecutar varias aplicaciones, que tenga una barra de herramientas y un menú, etc. Todo esto lo proporciona el entorno de escritorio. Proporcionan elementos comunes de la interfaz gráfica de usuario (GUI), como iconos, barras de herramientas, fondos de pantalla y widgets de escritorio. Gracias al entorno de escritorio, puede usar Linux gráficamente usando su mouse y teclado como lo hace en otros sistemas operativos como Windows y macOS.

La mayoría de los entornos de escritorio tienen su propio conjunto de aplicaciones y utilidades integradas para que los usuarios obtengan una sensación uniforme mientras usan el sistema operativo. Entonces, obtienes un explorador de archivos, búsqueda en el escritorio, menú de aplicaciones, editores de texto y más.



Entorno de escritorio Plasma 5.22



Entorno de escritorio Gnome 42

Hay varios entornos de escritorio disponibles para Linux, y estos entornos de escritorio determinan cómo se ve su sistema Linux y cómo interactúa con él. Algunos de los entornos de escritorio más utilizados son:

- **GNOME:** El entorno GNOME es el más utilizado en Linux, al punto que mucha gente cree que es parte de él. Las distribuciones como Debian y Fedora usan GNOME como su entorno predeterminado estándar
- **KDE plasma:** el entorno de escritorio Linux más utilizado después de GNOME. Se usa en distribuciones como Kubuntu o openSuse.
- **Mate:** proporciona la experiencia de escritorio más tradicional. Es la continuación de GNOME 2.
- **LXDE:** significa entorno de escritorio ligero X11. Utiliza una memoria RAM mínima en comparación con otros entornos de escritorio, ya que utiliza muy pocos recursos. Es más adecuado para computadoras viejas con hardware obsoleto.

Las distribuciones de Linux se envían con un entorno de escritorio predeterminado, pero puede instalar otros nuevos y cambiarlos en el momento del arranque.

4.5 El administrador de pantalla

Podemos iniciar la computadora en modo texto y luego lanzar el entorno gráfico. Pero muchas veces lo que queremos es arrancar directamente en modo gráfico. Aquí es donde entra en juego el administrador de pantalla.

Un gestor de sesiones es un programa que nos permite iniciar sesión en el ordenador. Por ejemplo: cuando la computadora se inicia en modo texto, inicia un programa (generalmente `getty` o `agetty`) que solicita al usuario un nombre de usuario y una contraseña. Validará si el usuario puede iniciar sesión en el sistema e iniciará el shell después de eso.

Si el administrador de sesión se ejecuta en modo gráfico y nos permite abrir una sesión gráfica en una computadora, se llama administrador de pantalla. Las distribuciones de Linux vienen con un administrador de pantalla, pero podemos cambiarlo si queremos. Algunos de los gestores de pantalla más utilizados son:

- **GNOME Display Manager (GDM):** es el administrador de pantalla predeterminado para GNOME
- **Simple Desktop Display Manager (SDDM):** es un administrador de pantalla liviano que se escribió desde cero en C++ y admite la creación de temas. Este administrador está ganando popularidad con el tiempo. Es el sucesor de KDE y se utiliza en colaboración con KDE plasma.
- **Administrador de visualización de KDE (KDM):** este era el antiguo administrador de visualización de KDE4, que se basaba en XDM (un administrador de pantalla más antiguo)
- **Light Display Manager (LightDM):** es un administrador de pantalla de escritorio cruzado. Admite diferentes tecnologías de escritorio, diferentes tecnologías de visualización (X, Wayland...) y es liviano, tiene un bajo uso de memoria y un alto rendimiento.

5. REGISTROS

Los logs o bitácoras del sistema son archivos de texto que registran cronológicamente todas las actividades y eventos importantes que ocurren en el sistema operativo o en la red. Linux registra absolutamente todo lo que sucede en el sistema operativo a través de registros.

Algunos ejemplos de la información contenida en los registros de Linux son los siguientes:

- Paquetes que se instalan y desinstalan en el sistema operativo.
- Información sobre accesos remotos a su ordenador.
- Intentos fallidos de autenticar a los usuarios en la computadora.
- Registro de errores que se produzcan en los programas o servicios que utilizamos.
- Accesos o salidas bloqueadas por nuestro firewall.
- Etc.

Casi todos los registros se almacenan en el directorio `/var/log` y sus subdirectorios. Algunos de los registros más relevantes son:

- `/var/log/syslog` o `/var/log/messages`: mensajes generales, así como información relacionada con el sistema. Esencialmente, este registro almacena todos los datos de actividad en todo el sistema global.
- `/var/log/auth.log` o `/var/log/secure`: almacenar registros de autenticación, incluidos los métodos de autenticación y los inicios de sesión exitosos y fallidos.
- `/var/log/boot.log`: registra información relacionada con el arranque y cualquier mensaje registrado durante puesta en marcha.
- `/var/log/kern`: almacena registros del Kernel y datos de advertencia. Este registro es valioso para la resolución de problemas núcleos personalizados también.
- `/var/log/dmesg`: mensajes relacionados con los controladores de dispositivos. El comando `dmesg` se puede usar para ver mensajes en este archivo.
- `/var/log/dpkg.log`: registro de administración de paquetes para Debian y Ubuntu, que registra qué paquetes se instalaron y/o desinstalaron en el sistema. Para entornos Red Hat y Fedora el archivo es el archivo: `/var/log/rpmpkgs`

Para mostrar un registro, puede usar el comando `menos`. Por ejemplo, para mostrar el contenido del `syslog` puedes usar:

```
menos /var/log/syslog
```

También puede usar el comando `cola`. Lo que hace `tail` es generar la última parte de los archivos. Entonces, si emite el comando:

```
cola /var/log/syslog
```

imprimirá solo las últimas líneas del archivo `syslog`. Cuando emite el comando con `-f` argumento:

```
cola -f /var/log/syslog
```

`tail` continuará viendo el archivo de registro e imprimirá la siguiente línea escrita en el archivo

Otro comando importante para consultar registros es el comando `dmesg`. Mostrará los mensajes del kernel. El argumento `-T` es para mostrar la hora del mensaje:

```
dmesg -T
```

Salida de ejemplo:

```
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: iniciando la carga del módulo kernel pstore_blk...
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: iniciando la carga del módulo kernel pstore_zone...
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: iniciando la carga de ramoops del módulo kernel...
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: la verificación de condición resultó en la omisión de la verificación del sistema de archivos en el dispositivo raíz.
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: iniciando la carga de módulos del kernel...
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: Inicio de Remontaje de sistemas de archivos raíz y kernel...
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: Iniciando Coldplug Todos los dispositivos udev...
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: sistema de archivos de páginas enormes montado.
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: sistema de archivos de cola de mensajes POSIX montado.
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: Sistema de archivos de depuración de kernel montado.
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: Sistema de archivos de seguimiento del kernel montado.
[Martes, 25 de octubre, 15:27:17, 2022] systemd[1]: finalizó la creación de una lista de nodos de dispositivos estáticos.
```

Puede combinar `dmesg` con `grep` y `menos` usando tuberías y buscar información:

```
dmesg | grep <algo> | menos
```

Este comando es invaluable para solucionar errores relacionados con el hardware, advertencias y para diagnosticar fallas en el dispositivo.

6.MATERIAL COMPLEMENTARIO

La terminal Linux:

- <https://www.makeuseof.com/what-are-linux-virtual-consoles/>
- <https://www.linuxbabe.com/command-line/linux-terminal>

Interfaces de usuario y escritorios:

- <https://unix.stackexchange.com/questions/596894/how-does-linuxs-display-work>
- https://learning.lpi.org/en/learning-materials/102-500/106/106.1/106.1_01/
- https://learning.lpi.org/en/learning-materials/102-500/106/106.2/106.2_01/