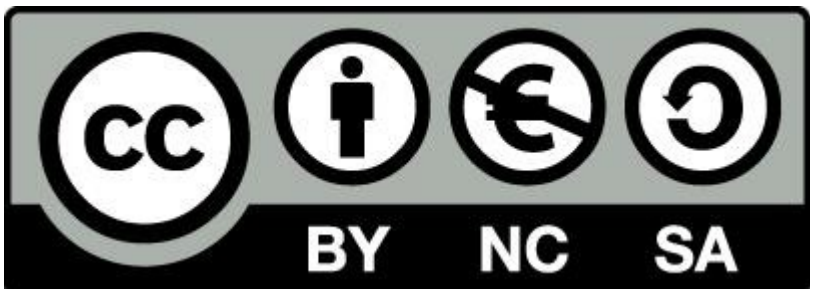
INTRODUCCIÓN A SISTEMAS LINUX

SISTEMAS OPERATIVOS MONOPUESTO

Mario García Alcázar



Última revisión Enero de 2016.

Índice de contenido

1. Introducción.

Este manual tiene como objetivo introducir al lector sobre los conceptos básicos necesarios para trabajar con sistemas operativos Linux basados en distribuciones Debian. Para ello trataremos diversos conceptos como las principales distribuciones existentes, el proceso de inicio del sistema o los principales ficheros de configuración.

2. Principales distribuciones Linux.

Linux es un sistema operativo de libre distribución. Esto quiere decir que, en principio, sus diversas implementaciones son gratuitas.

Para instalar un sistema operativo Linux, primero hemos de obtener una distribución. Una distribución es una recopilación de programas y ficheros organizados y preparados para su instalación. Estas distribuciones vienen normalmente con licencia GPL (Gnu Public License), lo cual implica que pueden obtenerse gratuitamente tanto de los ficheros binarios como del código fuente de dicha distribución. Además se permite también la modificación del código fuente para crear nuevas distribuciones, siempre que están sigan teniendo licencia GPL.

Desde la aparición de este sistema operativo, muchas empresas, universidades u organizaciones han creado sus propias distribuciones. En este capítulo analizaremos las más importantes.

- **Red Hat – Fedora**. Esta es una de las distribuciones más populares de Linux. Estas distribuciones fueron creadas por la empresa Red-Hat, la cual desarrolla para cada versión dos distribuciones alternativas, una libre con licencia GPL (llamada Fedora Core) y otra de pago, orientada a empresas llamada normalmente Red Hat Enterprise Linux. En principio ambas distribuciones tienen la misma potencia, pero la comercial se vende también una amplia documentación y soporte técnico.
- **SuSe**. Es una distribución alemana cuyo enfoque ha sido desde el principio básicamente comercial. Funciona de forma parecida a Red Hat, de forma que para cada versión saca una distribución comercial (SUSE Enterprise Server) y otra libre (OpenSUSE). Al igual que en el caso anterior, ambas distribuciones tienen la misma potencia, pero la comercial se vende también con una amplia documentación y soporte técnico.
- **Debian**. Es una distribución GNU no comercial, es decir, no depende de ninguna empresa para su desarrollo, sino de programadores individuales que cooperan entre si para crear un sistema operativo robusto. Actualmente es una de las distribuciones más extendidas de Linux, tanto en el ambito empresarial como en el campo de los ordenadores caseros.

  Existen otras distribuciones basadas en Debian. Las más importantes son **Ubuntu y Linux Mint.**

3. Primeros pasos con el modo comando.

En este capítulo se explicará el funcionamiento del entorno de trabajo en modo texto usado en sistemas Linux.

En concreto hablaremos de las consolas Linux y el programa shell.

3.1 Las consolas de Linux.

Cuando se inicia un sistema Linux, se crean las llamadas consolas virtuales. Una consola no es más que un interfaz que permite al usuario trabajar con el sistema.

Normalmente al arrancar el sistema se crean siete consolas virtuales. Seis son en modo texto (tty1, tty2... tty6) y una es en modo gráfico (tty7). El objeto de tener varias consolas es el de permitir que varios usuarios puedan entrar a la vez al sistema y operar con el.

Para acceder a estas consolas desde modo gráfico debemos teclear CTRL + ALT + F1, para la consola 1, CTRL + ALT + F2 para la consola 2, etc.

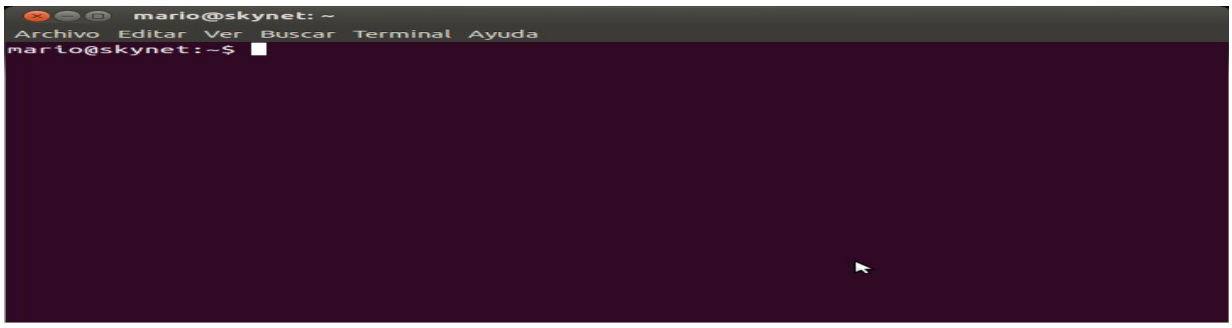
Si estamos en modo texto, para conmutar de consola basta con pulsar ALT + F1 para la primera consola, ALT + F2 para la segunda, ect.

Por último si estamos en una consola modo texto y queremos volver a la consola modo gráfico, debemos pulsar ALT + F7.

3.2 La Shell.

La shell es un programa que recoge los comandos introducidos por el usuario, los interpreta, y si son correctos los ejecuta. La shell nos muestra por pantalla el llamado prompt o indicador de comandos, el cual es configurable para el sistema o para cada usuario particular.

Para abrir un interprete shell en nuestro sistema, solo tenemos que abrir un terminal desde el menú **aplicaciones → accesorios → terminal.**



Como podemos ver en la imagen, el shell tiene un encabezado de texto "mario@skynet: $" donde se indica como información el **nombre del usuario** con el que estamos trabajando (mario), el **nombre del equipo** (skynet) y por último un **carácter** que nos indica si estamos trabajando como un usuario normal ($) o con permisos de administrador (#).

Existen diversas versiones de shell, no obstante los más utilizados son **sh** (Bourne Shell) o una mejora posterior llamada **bash** (Bourne Again Shell). Para ver la versión de shell que estamos usando basta teclear en una consola el comando **echo $SHELL.**

La shell nos ofrece además dos funcionalidades para facilitar nuestro trabajo:

- **El historial**. La Shell almacena el historial de órdenes tecleadas por el usuario en el fichero $HOME/.bash_history. Para visualizar los comandos ejecutados basta con pulsar las teclas de las flechas de arriba y abajo.
- **La función de expansión**. Se utiliza para completar el nombre de un fichero que se quiera utilizar en un comando. Para ello basta con teclear las primeras letras del nombre de dichos fichero y pulsar la tecla TAB.

Por otra parte, hay que indicar la secuencia de scripts que lanza el shell en el momento de la conexión o desconexión de un usuario al sistema.

Concretamente en la conexión se ejecutan los siguientes scripts en el orden indicado:

1. /etc/profile
2. $HOME/.bash_profile
3. $HOME/.bash_login
4. $HOME/.profile

En el momento de desconectarse, intenta ejecutar:

1. $HOME/.bash_logout

4. Conceptos básicos.

Una vez dominamos el uso de las consolas Linux, trataremos una serie de conceptos básicos necesarios para entender el funcionamiento del sistema operativo Linux.

4.1 Estructura de procesos en Linux.

Actualmente, todos los sistemas operativos, incluido Linux, son sistemas multitarea, es decir, permiten la ejecución de varios procesos de forma concurrente.
En este punto hemos de diferenciar dos conceptos fundamentales:

- **Programa**. Es un conjunto de instrucciones escrito en un lenguaje de programación y almacenado en uno o varios ficheros del disco duro.
- **Proceso**. Es un programa en ejecución, ubicado en la memoria RAM del sistema.

En Linux, la estructura de los procesos en memoria, es jerárquica, esto quiere decir que los procesos se organizan con una estructura en forma de árbol, igual que ocurre con los directorios y ficheros.
Para ello, se crea la relación **padre-hijo**, que consiste en que cada proceso debe haber sido creado por otro, el cual será su proceso padre.

El primer proceso que se crea en Linux cuando arrancamos el proceso es el proceso INIT. A partir INIT, se crean el resto de procesos de usuario o sistema.

Por último hay que indicar que todo proceso de Linux tiene un identificador numérico único llamado **PID** (Process ID) en el caso del proceso INIT su PID es siempre el 1.

4.2 Los usuarios de Linux.

Linux tiene tres tipos básicos de usuarios:

- **Usuarios comunes**. Son los usuarios corrientes del sistema. Tienen acceso restringido a determinados ficheros y recursos del sistema, así como a los archivos del resto de usuarios. Normalmente solo tienen permisos para trabajar en su directorio de trabajo, también llamado directorio HOME. Este directorio es /home/usuario, donde usuario es el login del usuario. Ej.- /home/mario
- **Usuarios de sistema**. Son usuarios utilizados por el sistema operativo para lanzar procesos con unos determinados permisos. Por ejemplo existe un usuario 'lp' encargado de la gestión de las impresoras. Ninguna persona puede entrar en el

sistema haciéndose pasar por un usuario de sistema, ya que la seguridad de Linux lo impide.

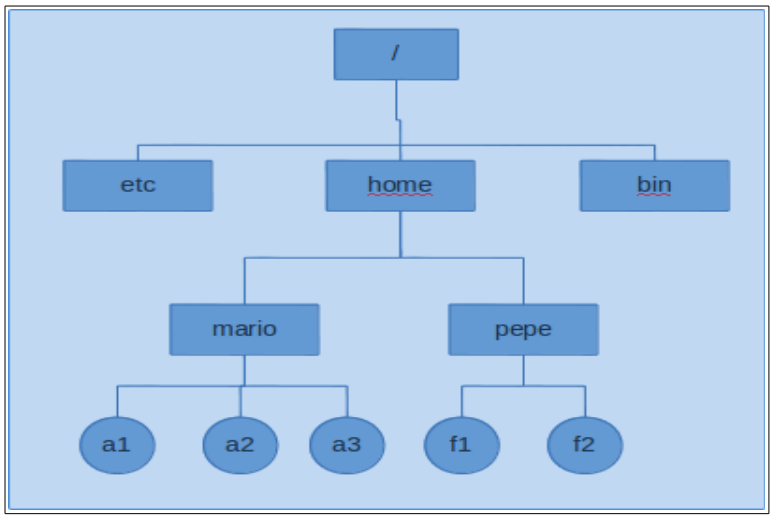
- **Usuario root**. También llamado superusuario. Root puede hacerlo todo. Puede ver y modificar cualquier fichero del computador. Por tanto, su uso debe realizarse de modo responsable por los administradores del sistema. Para acceder al sistema como root, entramos con otro usuario, escribiremos en una consola el comando su root e introducimos la contraseña de root. Por otra parte, podemos saber en todo momento si estamos operando como usuario root, ya que su prompt termina con el carácter '#'.

4.3 El árbol de directorios.

Linux utiliza igual que otros sistemas operativos, ficheros y directorios para organizar la información. Los directorios forman una estructura en modo de árbol, cuyo origen se llama directorio raíz. Este directorio raíz se muestra en Linux con el carácter '/' y en Windows con las siglas C: D:, etc.

No obstante, utiliza solamente en un árbol de directorios, y no uno por disco o uno por partición como hace Windows. Si se añade un nuevo disco al sistema, debe “colgarse” de un directorio del árbol. Esto se hace mediante un proceso denominado **montaje** que se realiza mediante el comando **mount** el cual trataremos posteriormente.
Por ejemplo, cuando conectamos una memoria usb o un disco externo al ordenador, automáticamente Linux no crea otra unidad (D: F: o G:) sino que lo **monta** en un directorio con su nombre ubicado dentro del directorio **/media**

Por otra parte para hacer referencia a un fichero se utiliza lo que se llama **ruta**, es decir, la descripción de todos los directorios que hay que recorrer para llegar hasta él.



En este punto, hemos de indicar la diferencia entre ruta absoluta y ruta relativa.

- **Ruta absoluta**: Es la ruta completa hasta el archivo desde el directorio raíz del sistema.
  Por ejemplo, *según la imagen anterior una ruta absoluta válida sería: /home/mario/a1*

- **Ruta relativa**: Es la ruta hasta el archivo, desde el directorio en el que se encuentra el usuario. Para trabajar con rutas relativas, se usa el carácter '.' para referenciar al directorio actual y '..' para subir un nivel en el árbol de directorios.
  Siguiendo el ejemplo anterior, si estamos en el directorio */home/mario*, podríamos acceder al fichero f1 con la ruta relativa *../pepe/f1*

Por último, para acabar con este apartado, se listan los principales directorios de sistema en Linux, con la información que contienen.

DIRECTORIO	USO
/bin	Binarios de los principales comandos del sistema.
/boot	Ficheros utilizados durante el arranque del sistema
/dev	Dispositivos conectados al sistema (discos, impresoras, etc).
/etc	Ficheros de configuración generales del sistema y de servicios instalados en este.
/home	Directorios de inicios de los usuarios
/lib	Bibliotecas compartidas esenciales para los binarios de /bin
/media	Puntos de montaje para dispositivos externos como memorias USB.

/proc	Sistema de ficheros virtual que documenta sucesos y estados del núcleo. Contiene principalmente ficheros de texto.
/root	Directorio de trabajo del usuario root.
/sbin	Binarios asociados a comandos de administración del sistema.
/tmp	Ficheros temporales
/usr	Directorio donde se instalan las aplicaciones instaladas.
/usr/bin	Programas ejecutables de las aplicaciones instaladas.
/usr/lib	Bibliotecas compartidas.
/usr/share	Datos compartidos independientes de la arquitectura del sistema. Imágenes, ficheros de texto, etc.
/usr/src	Códigos fuente del sistema operativo.
/var	Ficheros variables, como son logs, bases de datos, directorio raíz de servidores HTTP y FTP, colas de correo, ficheros temporales, etc.
/var/log	Ficheros y directorios de registro de sucesos.

4.4 Las propiedades de los archivos.

En un sistema Linux, cada archivo tiene una serie de permisos asociados. Estos permisos determinan que usuarios pueden leer, modificar o ejecutar dicho archivo o directorio.

Para ver los permisos de un archivo, sólo tenemos que posicionarnos en el directorio que lo contiene y ejecutar el comando 'ls -l'.

Al ejecutar este comando, el sistema nos muestra una línea por cada archivo o directorio con información detallada de este. La primera columna de cada fila nos muestra las propiedades de las que hablamos antes.

La interpretación de estos símbolos es la siguiente:

- El primer carácter indica el **tipo de archivo**:
  - **_** Archivo normal.
  - **d** Directorio.
  - **l** Enlace simbólico.
  - **c** Archivo de dispositivo tipo carácter.

    - **b** Archivo de dispositivo tipo bloque.
- Los siguientes tres caracteres indican los permisos que tiene sobre el fichero **su usuario propietario**.
    - Si tiene una **r** en la primera posición es que puede leerlo.
    - Si tiene una **w** en la segunda posición es que puede modificarlo.
    - Si tiene un **x** en la tercera posición es que puede ejecutarlo.
- Los siguientes tres caracteres indican los permisos que tiene sobre el fichero **los usuarios de su grupo propietario**.
    - Si tiene una **r** en la primera posición es que puede leerlo.
    - Si tiene una **w** en la segunda posición es que puede modificarlo.
    - Si tiene un **x** en la tercera posición es que puede ejecutarlo.
- Los siguientes tres caracteres indican los permisos que tiene sobre el fichero **el resto de usuarios del sistema**.
    - Si tiene una **r** en la primera posición es que puede leerlo.
    - Si tiene una **w** en la segunda posición es que puede modificarlo.
    - Si tiene un **x** en la tercera posición es que puede ejecutarlo.
- Existen también dos atributos especiales:
    - **S**: los atributos suid y sgid, otorgan a un "fichero" los permisos de su dueño o grupo respectivamente, cada vez que se ejecute ,sea quien sea el que lo ejecute. El atributo s nunca puede aplicarse al resto de usuarios, solo al propietario o al grupo. Este atributo aparece en la posición de la 'x' de los permisos de usuario (suid) o en la posición de la 'x' de los permisos del grupo (guid).
    - **T**: el atributo sticky (pegajoso) hace que sólo el propietario del fichero pueda borrarlo. Este atributo aparece solamente en la posición de la ‘x’ de los permisos del reto de usuarios.

Independientemente de los permisos, hay que recordar que root siempre puede realizar cualquier acción sobre cualquier fichero o directorio

4.5 Las variables de entorno.

Una variable de entorno no es más que una etiqueta capaz de almacenar un valor determinado.
El ámbito de dicha variable es el de la sesión que hayamos establecido si estamos en

modo texto, o la consola de comandos en la que nos encontremos, si estamos en modo gráfico.
Para crear una variable de entorno, sólo tendremos que ejecutar el siguiente comando:
nombre_variable=valor

Ej.- A1=casa

Para obtener el valor de una variable de entorno, pondremos el símbolo $ delante de dicha variable.

Ej.- $A1

Cuando shell encuentra el símbolo $ delante de un conjunto de caracteres determina que se trata de una variable de entorno y cambia esa etiqueta por el valor que se le dió, antes de llevar a cabo la ejecución del comando.

Ej.- echo $A1 daría como resultado – casa

4.5.1 Variables de entorno predefinidas.

Existen algunas variables de entorno definidas en los scripts de arranque y configuración del sistema.
Para poder ver la lista completa de todas las variables de entorno podemos utilizar el comando **set.**
Algunas de las variables predefinidas más usuales son las siguientes:

- **DISPLAY**. Características del entorno gráfico
- **HOME**. Directorio de trabajo del usuario validado
- **HOSTNAME**. Nombre del host
- **PATH**. Ubicaciones donde shell buscara los ficheros ejecutables
- **PS1**. Prompt de la sesión
- **PWD**. Directorio actual
- **SHELL**. Versión utilizada de shell
- **UID**. ID de usuario validado
- **USER**. Nombre del usuario validado

5. Uso avanzado de comandos. Redirección de comandos y tuberías.

Los comandos Linux no lanzan los resultados de salida directamente a la pantalla. En lugar de esto, se los envían a una rutina especial denominada **salida estándar o stdout**. Esta rutina está configurada para enviar esos datos a la pantalla.
No obstante, el usuario puede configurar un determinado comando para que no envíe sus resultados a la salida estándar, sino a un fichero o dispositivo. Para ello, se usa el carácter '**>**'. Pongamos un ejemplo:

- cat /etc/passwd envía el contenido del fichero /etc/passwd a la rutina de la salida estándar, la cual lo muestra por pantalla.
- cat /etc/passwd > f1 guarda el contenido del fichero /etc/passwd dentro del fichero f1.

Por otra parte, cuando un comando Linux detecta un error al ejecutarse envía un mensaje de error a otra rutina llamada **salida de error estándar o stderr**. Esta rutina envía sus resultados a la pantalla, pero el usuario puede configurar el comando para que no envíe sus resultados a la **salida de error estándar**, sino a un fichero o dispositivo. Para ello, se usa el carácter '**2>**'.

- cat /etc/shadow 2> f2

También existe una **entrada estándar o stdin**, la cual puede redirigirse con el carácter '**<**'. Un ejemplo podría ser el siguiente:

- cat < /home/usuario/prueba1

Hay que indicar que los operadores **>** y **2>,** utilizados sobre archivos eliminan la información previa que tuviesen estos. Para conservar la información y que el operador añada la información de la redirección sin sobrescribir se usan los operadores **>>** y **2>>**.

Por último hablaremos de las **tuberías**. Una tubería es un sistema que permite encadenar la ejecución de varios comandos, de forma que la salida de un comando sea la entrada del siguiente. Para ello se usa el carácter '|'. Pongamos un ejemplo:

cat /etc/passwd | wc –l -> para interpretar este comando hemos de hacerlo de izquierda a derecha. Primero se ejecuta el comando **cat /etc/passwd**. Este comando muestra el contenido del fichero, pero al haber después una tubería **'|'**, la salida del cat no se muestra por pantalla, sino que pasa a ser la entra da del comando **wc –l** el cual muestra por pantalla el número de líneas de un texto. En este caso, se mostrará por pantalla el número de líneas del fichero /etc/passwd.

6. Arranque del sistema.

En todos los sistemas Linux, el proceso init, se encarga de realizar el arranque del sistema. Este proceso, distingue múltiples niveles de ejecución, cada uno de los cuales puede tener su propio conjunto de procesos que se inician. Los niveles de ejecución válidos son 0-6. Concretamente:

Nivel	Descripción	Configurable por el usuario
0	Modo de parada o apagado del equipo	no
1	Modo monousuario sin red. Suele usarse para tareas de mantenimiento.	no
2	Modo gráfico multiusuario. Suele estar por defecto.	si
3	Modo gráfico multiusuario. Igual que el 2.	si
4	Modo gráfico multiusuario. Igual que el 2.	si
5	Modo gráfico multiusuario. Igual que el 2.	si
6	Modo de reinicio	no

Para cambiar de un nivel a otro basta con ser root y usar el comando init. Por ejemplo init 6 reiniciaría el sistema.

En este punto, hay que diferenciar entre dos tipos de sistemas de arranque en Linux. Estos son **systemV** y **upstart**.

SystemV es un sistema antiguo que usaba un fichero de configuración (/etc/initab) para indicar que servicios debían arrancarse en cada runlevel. Actualmente ha caído en desuso, por lo que no lo trataremos en este curso. Nos centraremos por tanto en el sistema **upstart** que se utiliza en la actualidad.

6.1 Arranque de sistemas Linux con UPSTART.

Upstart es un *init daemon* creado por el Ubuntu-team que pretende ser sustituto del init de System V. Es el único proceso que arranca el núcleo y encargado de llamar al resto de programas/servicios del sistema. En Linux, todo proceso es hijo de init.

Ciertamente Upstart presenta interesantes mejoras respecto del init de System V: Arranque de servicios en paralelo, guiado por eventos, servicios que se rearrancan automáticamente si mueren, etc.

En el directorio */etc/event.d/* tenemos los ficheros de configuración de upstart que, funcionan como si del fichero */etc/inittab* se tratase. Estos ficheros son:

- Ficheros ttyX: que indican en qué nivel de ejecución ha de arrancarse cada consola virtual en modo texto.
- Ficheros rcX: que acaban *ajustando* el runlevel y llamando a */etc/init.d/rc* para ejecutar los scripts del nivel de arranque correspondiente.
- *ctrl-alt-del que indica las acciones a tomar cuando ejecutamos esta secuencia de teclas.*
- *rc-default* que define el nivel de inicio por defecto.

Además, para cada nivel de ejecución hay un directorio llamado /etc/rcX.d donde tenemos enlaces soft a todos los servicios que han de iniciarse o detenerse en dicho nivel de ejecución. Si el nombre del enlace tiene una 'S' es que hay que iniciar ese servicio, y si tiene una 'K' es que hay que detenerlo. Además cada enlace tiene en su nombre un número que indica el orden de ejecución de los programas del directorio.

7. Ficheros de configuración.

En este apartado, trataremos las funciones y características de los principales ficheros de configuración del sistema. Estos ficheros son modificables por el usuario root, pero cualquier cambio ha de realizarse con cuidado ya que puede conllevar un mal funcionamiento del sistema.

- **/etc/passwd.** Este fichero mantiene una lista con todos los usuarios del sistema. Cada línea del fichero contiene la información de un usuario con los siguientes campos separados por el carácter ':'
  - Nombre del usuario
  - Este campo almacenaba la contraseña encriptada, pero ya no se usa. Si tiene el carácter **x**, es que es un usuario común. Si tiene el carácter '!' es que es un usuario de sistema, que no tiene contraseña, por lo cual no podemos validarnos en el sistema con él.
  - Identificador único del usuario (UID). Es un número que lo identifica unívocamente.
  - Identificador unívoco del grupo primario de ese usuario (GID). Todo usuario tiene un grupo primario, que es el que aparece en este campo. Aparte, puede estar incluido en más grupos, pero eso ya se refleja en el fichero /etc/group.
  - Información de contacto del usuario.

- Directorio $HOME del usuario.
- Tipo de shell del usuario.

- **/etc/shadow.** Este fichero almacena la contraseña de cada usuario y parámetros asociados con la validez de dichas contraseñas. Los usuarios corrientes no tienen permisos sobre este fichero. Los campos del fichero son los siguientes:

- **/etc/group.** En este fichero se guardan los datos de los grupos existentes en el sistema. Tiene los siguientes campos separados por el carácter ':'
  - Nombre del grupo.
  - Un campo con el carácter x, que ya no se utiliza.
  - El GID o identificador único de grupo.
  - La lista de usuarios secundarios que pertenecen al grupo, separados por comas. Todo usuario tiene un grupo primario, indicado en el fichero /etc/passwd, pero aparte puede estar incluido en más grupos, como se refleja en este campo.

- **/boot/grub/grub.cfg** Este fichero permite configurar la herramienta de inicio de sistema **GRUB.** Entre otras cosas permite modificar los siguientes parámetros:
  - Tiempo de espera para el arranque del sistema. Modificando el valor de la variable tiemout
  - Sistema operativo a arrancar por defecto. Modificando el valor de la variable default.
  - Colores del menú de GRUB. Modificando el valor de la variable color.

- **/etc/fstab.** Este fichero almacena la lista de sistemas de ficheros existentes en nuestro equipo, y de qué forma ha de realizarse el proceso de montaje de estos, en el arranque del sistema. Esta compuesto por Íineas con el siguientes campos separados por espacios en blanco:
  - Dispositivo a montar.
  - Punto de montaje.
  - Tipo de sistema de ficheros.
  - Opciones especiales de montaje.

  Para más información sobre este comando consultar el comando mount del manual de comandos Linux.

- **/etc/profile.** Este fichero es un sencillo programa shell script que se ejecuta cuando el usuario se valida en el sistema. Este es un fichero de configuración general para todos los usuarios. Los usuarios comunes NO pueden modificar este fichero.

- **$HOME/.profile**. Este fichero es un sencillo programa shell script que se ejecuta cuando el usuario se valida en el sistema, después de ejecutar el fichero de configuración general de usuarios /etc/profile. Los usuarios pueden modificar este fichero para introducir sus propios parámetros de configuración.

- **$HOME/.bash_logout**. Fichero que se ejecuta en la desconexión del usuario.

- **$HOME/.bash_history.** Este fichero guarda los últimos comandos introducidos ejecutados por un usuario determinado.