

Contenido

Historia De Linux	3
Ubuntu	3
Bash	3
Historial	4
Ayuda	4
Ficheros En Linux.....	5
Localización De Ficheros.....	5
Montaje de dispositivos	6
Acceso A Directorios Y Ficheros	7
Colores en el terminal	7
Caracteres Especiales.....	7
Expansión De Nombres En Bash (Comodines)	8
Ejecución De Comandos	8
Ejecución Especial De Comandos	8
Variables.....	9
Gestión De Ficheros Y Directorios	10
Compresión De Ficheros	10
Redirecciones	11
Tratamiento De Textos.....	11
Expresiones Regulares.....	13
Expresiones Regulares Extendidas	13
Grep.....	13
Sed.....	14
Niveles De Ejecución	15
Enlaces.....	15
Administración De Software En Linux.....	16
Herramientas Para La Administración De Paquetes Debian (Ubuntu).....	17
Dpkg	17
Apt-Get.....	17
Apt-Cache.....	18
Aptitude.....	18
Ficheros De Configuración De Repositorios	19
Añadir Un Nuevo Repositorio	19

Herramientas Para La Administración De Paquetes Rpm	19
Rpm	19
Yum	19
Instalación Manual De Software	20

HISTORIA DE LINUX

- Tiene su origen en el sistema operativo Unix, desarrollado a principios de los años 70 en los laboratorios Bell.
- Linux está escrito en el lenguaje de programación C
- Fue creado por Linus Torvalds en 1991 a partir de Minix, una variación de Unix para fines educativos.
- Linux es únicamente el núcleo del sistema operativo. No las aplicaciones de sistema ni de usuario.
- Linux fue licenciado como software libre, bajo licencia GNU GPL. Su unión con el proyecto GNU fue lo que le proveyó del resto de programas de sistema que lo completaron como sistema operativo.
- A las variantes de esta unión de programas y tecnologías, a las que se les adicionan diversos programas de aplicación se les denomina distribuciones. Su objetivo consiste en ofrecer ediciones que cumplan las necesidades de un determinado grupo de usuarios.

UBUNTU

- Dentro de las [distribuciones](#) existentes de Linux hemos escogido trabajar con Ubuntu.
- Ubuntu es una bifurcación del código base del proyecto [Debian](#).
- Está compuesta de software distribuido bajo una licencia libre o de código abierto. Es la distribución Linux de mayor uso como S.O. de escritorio (junto a Linux Mint)
- Tiene como respaldo la empresa Canonical que provee versiones para servidores, para empresas, televisores, tabletas, teléfonos...
- Se publica una nueva versión cada 6 meses. Con soporte para 9 meses.
- Las versiones LTS (Long Term Support) aparecen cada 2 años y reciben soporte durante 5 años.

BASH

- Bash Es el intérprete de comandos más usado (desarrollado por GNU)
- El intérprete se encarga de la ejecución de las órdenes, redirecciones, gestión de la entrada y salida de los comandos, incorporar comandos internos, etc.
- Existen otros (dash, zsh, csh). Suelen ser parecidos, pero puede cambiar la forma de uso de comandos internos, tuberías, redirecciones, etc.
- **Ubicación:** /bin/bash
- En muchas distros existe el enlace: /bin/sh → /bin/bash (o el usado por defecto)
- **Prompt:** indica que el intérprete espera órdenes y muestra cierta información (usuario, máquina, directorio actual, tipo de usuario, etc). Se guarda en \$PS1.
- PROMPT\$: usuario normal
- PROMPT#: usuario root (administrador)

ALT+F1-F6	Cambia entre las distintas consolas de texto
ALT+F7-F9	Cambia a las consolas gráficas
CTRL+ALT+F1-F6	Cambia a las consolas de texto desde una consola gráfica
reset	Reinicializa el terminal
clear	Borra el terminal
bash	Abre una nueva instancia de bash dentro de la actual
exit	Cierra la instancia del bash actual

HISTORIAL	
↑↓	Navega por el historial de comandos
CTRL+r	Realiza una búsqueda inversa por palabras en el historial Intro: ejecuta el comando encontrado Esc: edita el comando encontrado CTRL+G: cancela la búsqueda
history history 10 history -c	Muestra el historial del usuario almacenado en ~/.bash_history Muestra las últimas 10 líneas Borra el historial
(espacio) comando	Al ejecutar un comando con un espacio delante evita que se guarde en el historial
!cadena	Ejecuta el comando más reciente del historial que comience por cadena
!!	Ejecuta el último comando de nuevo

AYUDA	
<ul style="list-style-type: none"> ○ Las páginas de manual guardan la documentación de los programas instalados ○ Se organizan en secciones (número entre paréntesis al lado de cada página) <ul style="list-style-type: none"> - Programas ejecutables y comandos del Shell - Llamadas al sistema (proporcionadas por el kernel) - Bibliotecas (funciones en bibliotecas de programas) - Ficheros especiales (normalmente en /dev) - Formatos de ficheros de configuración - Juegos - Miscelánea - Comandos de administración del sistema (normalmente sólo para root) - Rutinas del kernel ○ Cada programa instala sus propias páginas en el manual ○ Activar páginas en castellano instalar los paquetes: manpages-es manpages-es extra 	
man comando	Busca la página del manual del comando <ul style="list-style-type: none"> - Buscar términos: /patrón - Avanzar: n - Retroceder: N - Salir: q
man n comando man fichero_config	Busca la página del manual del comando en la sección n Muestra información sobre el fichero de configuración
help comando_int	Muestra la ayuda sobre un comando interno del bash
type comando	Equivalente a man bash y buscar "SHELL BUILTIN COMMANDS" Comprueba el tipo de un comando y si es o no interno
apropos término man -k término	Muestra un listado de las páginas del manual en las que aparece el término
whatis	Muestra la descripción de la página de un comando
info comando	Similar a man pero utilizando una interfaz más avanzada que permite hipervínculos y combinación de teclas estilo emacs

FICHEROS EN LINUX

- Los ficheros son sensibles a mayúsculas y minúsculas: Fichero1 es distinto a fichero1
- Linux trata a todo como un fichero
 - Un directorio es un tipo especial de fichero
 - Los terminales (dev/pts/4, dev/tty1, ...)
 - Los discos y particiones (/dev/sda1)
 - Los procesos son tratados como ficheros en /proc/PID-proceso

LOCALIZACIÓN DE FICHEROS

FHS: estandarización de la jerarquía de directorios Linux

/bin/	Comandos binarios esenciales para todos los usuarios
/boot/	Archivos del arranque: núcleo, initrd, grub, etc.
/dev/	Archivos de dispositivos (virtuales y reales)
/etc/	Archivos de configuración locales
/etc/opt/	Ficheros de configuración de los programas ubicados /opt
/home/	Directorios de trabajo de los usuarios (excepto root)
/root/	Directorio de trabajo del usuario root
/lib/	Bibliotecas esenciales compartidas por los programas
/media/	Puntos de montaje de los medios extraíbles
/mnt/	Sistemas de archivos montados temporalmente por los usuarios
/opt/	Programas instalados de forma manual y compartidos
/sbin/	Binarios del sistema usados por el root
/srv/	Datos para los servicios del sistema
/tmp/	Archivos temporales escritos por los usuarios

- El primer carácter al hacer ls -l muestra el tipo de fichero
d: directorio | **-**: fichero regular | **l**: enlace simbólico | **s**: socket **b**:
 fichero especial de bloques | **c**: fichero especial de carácter | **p**: tubería

RUTAS ABSOLUTAS

- Cualquier ruta que comience por / es una ruta absoluta
- El directorio / representa la raíz de la jerarquía de ficheros en Linux
- En Linux no existe una jerarquía para cada unidad. Todo cuelga de /

RUTAS RELATIVAS

- Cualquier ruta que no comience por / es una ruta relativa
- Una ruta relativa depende del directorio actual: la ruta equivalente es igual a la ruta actual más la ruta original
- Ruta relativa: documentos/facturas /utilizada desde /home/juan)
- Ruta absoluta equivalente: /home/juan/documentos/facturas

/usr/	Utilidades y aplicaciones multiusuarios Es una jerarquía secundaria (/usr/bin, /usr/lib...)
/var/	Archivos con información y variables: logs, bases de datos, directorio raíz de HTTP, FTP, colas de correo, ficheros temporales, etc. Sistema de archivos virtual con información del núcleo y los procesos en ejecución
/proc/	
find ruta expresión	Busca ficheros y directorios según sus atributos en forma recursiva Busca por el nombre del directorio según un patrón

find ... -name "patrón"	Usar comillas dobles para evitar que el bash procese comodines
find ... -maxdepth n	Limita la profundidad de búsqueda
find ... -type tipo	Busca por tipo de archivo
find ... -size t	Busca por tamaño de t*512B prefijos +/- Sufijos para especificar tamaño: c, k, M, G
find ... -atime n	Busca los ficheros accedidos en las últimas n*24 horas
find ... -ctime n	Busca los ficheros modificados en las últimas n*24 horas
locate nombre	Utiliza una BD (de rutas más frecuentes) para encontrar ficheros
updatedb	Actualiza la BD que usa locatedb
whereis f1	Muestra las rutas del binario, fuente y manual de un comando. Sólo busca en algunos directorios (/bin, /sbin, /etc, /usr/lib, /usr/bin...)
which comando	Muestra las rutas del path con las que se ejecutaría un comando
type comando	Indica si una orden es un ejecutable, alias, comando interno...

MONTAJE DE DISPOSITIVOS

- Todos los discos de almacenamiento secundario o unidades externas deben estar en un mismo árbol.
- Al añadir una nueva partición al sistema, debemos indicar en donde queremos que esta sea montada.
- Montar una partición significa indicar en qué punto del sistema de archivos queremos que esta sea accesible.

mount	Sirve para montar una nueva partición de datos
mount -t ext4 /dev/sdXX /mnt/usb	Monta en /mnt/usb la partición sdXX con el sistema de archivos ext4
mount -f	Monta de manera "ficticia", para comprobar si se puede realizar la acción
mount -r	Monta en modo "sólo lectura"
Umount /mnt/usr	Desmonta la partición montada en /mnt/usr

Fichero fstab

- Contiene la información de montaje de los distintos dispositivos del sistema (recordad que las particiones se tratan en Linux como dispositivos, /dev/xxx).
- Podemos realizar esta tarea de forma directa o mediante el uso del administrador de discos que Linux nos ofrece
- Una nueva línea por cada dispositivo, que debe contener:
 - Nombre del dispositivo (o su UUID): si usamos el UUID debemos poner "UUID="antes del valor.
 - Punto de montaje: ruta del sistema de archivos.
 - Sistema de archivos: ext4, ext3, ntfs, etc.
 - Opciones.
 - Dump-freq: opción para guardar log sobre errores del sistema de archivos (0 – desactivado, 1 – activado).
 - Pass-num: indicar el orden en que la aplicación fsck escaneará la partición en busca de errores (0 = nunca).

○ EJEMPLOS:

/dev/sda1	/boot	vfat	defaults	0	2
/dev/sda2	/	ext4	defaults	0	1
/dev/sda3	/home	ext4	defaults	0	2
/dev/sda4	none	swap	defaults	0	0

ACCESO A DIRECTORIOS Y FICHEROS

ls ruta/fichero	Lista los ficheros de un directorio
ls -l	Muestra en modo lista detallada (permisos, fechas, tamaño, ...)
ls -h	Muestra los tamaños en modo human-readable
ls -d	Muestra el nombre de los directorios en lugar de su contenido
ls -a	Muestra todos los ficheros (ocultos incluidos) (Un fichero es oculto si su nombre empieza por un punto: .fichero)
cd [ruta]	Cambia del directorio actual a la ruta indicada
cd	Se dirige al directorio personal del usuario actual
cd ~	Se dirige al directorio personal del usuario actual
cd -	Se dirige al último directorio en que se encontraba el usuario
.	Representa el directorio actual
..	Representa el directorio padre del directorio actual
pwd	Muestra la ruta completa actual
<tab>	Permite completar rutas y nombre de ficheros sin errores Si existen varias opciones, completa la parte común y al pulsar otra vez <tab> muestra una lista con las posibilidades
cat fichero	Muestra por la salida el contenido de un fichero

COLORES EN EL TERMINAL

<ul style="list-style-type: none"> ○ En Linux todo es un archivo ○ Dependiendo de las propiedades de cada archivo, se distingue qué acciones pueden realizarse sobre ellos y quién puede hacerlas ○ Para distinguir estos distintos tipos de archivos, éstos se muestran en distintos colores dentro del terminal 	
Azul	Directorio
Verde	Archivo ejecutable
Cyan (Azul cielo)	Enlace simbólico (similar al acceso directo de Windows)
Amarillo con fondo negro	Dispositivo
Magenta (rosa)	Archivo de imagen o vídeo
Rojo	Archivo de tipo reconocido
Rojo con fondo negro	Enlace simbólico roto

CARACTERES ESPECIALES

CTRL+SHIFT+u+(cod hex)	Escribe un carácter UNICODE dado su código hexadecimal
AltGr+4	~
\t	Tabulación
\a	Alerta
\n	Nueva línea
\r	Retroceso
\v	Tabulación vertical
\\	Barra invertida
“texto con CE”	Evita las sustituciones especiales de Bash de metacaracteres No evita ~variables, el comando ! o comillas simples “ echo “!hola” #Dará un error echo ‘Soy \$USER’ #Escribirá el nombre del usuario actual

'texto'	Evita todas las sustituciones especiales de Bash de metacaracteres y órdenes. No pueden anidarse echo '!hola' #Escribirá: !hola echo 'Soy \$USER' #Escribirá: Soy \$USER
\(carácter)	Escapa el carácter especial char y lo trata como texto echo "Soy \ \$USER" #Escribirá: Soy \$USER cd Documents\ and\ Settings #Entrará al directorio con espacios
\$'texto'	Se sustituye por l texto considerando los caracteres especiales \a, \b, \n, \t, \v, \\
\ (al final de línea)	Divide una orden en varias líneas, pero se trata como una sola echo "Hola\ como estas" #Escribirá: Holacomo estás

EXPANSIÓN DE NOMBRES EN BASH (COMODINES)

<ul style="list-style-type: none"> Una expansión de Bash es una sustitución automática de un fragmento de texto especial por otro equivalente (ejemplo: ~por el directorio personal) Las expansiones de ficheros se sustituyen por los ficheros encontrados en ./ o si se comenzó a escribir una ruta absoluta o relativa por los coincidentes en dicha ruta Un comodín nunca sustituye al . inicial de un fichero oculto ni los directorios "." y ".." La sustitución la realiza el Bash antes de llamar al comando El comando recibe el texto sustituido 	
*	Sustituye a cualquier cadena (incluida cadena vacía) Doc* coincide con Doc, Documentos, Documento.txt, Doc345... Doc*.txt coincide con Doc.txt, Documento.txt, DocA.txt, Doc23.txt
?	Sustituye a un carácter cualquier (sólo 1) Doc? Coincide con DocA o Doc1 Doc? NO coincide con Doc o DocA.txt
[caracteres]	Sustituye a uno de los caracteres entre [] (sólo 1) Doc[A1j] coincide con DocA, Doc1, Docj Doc[0-9].txt coincide con Doc0.txt, Doc1.txt, Doc2.txt.. Doc[a-z] coincide con Doca, Docb, Docc Doc[0-9][0-9] coincide con Doc00, Doc45, Doc32...
[!caracteres]	Sustituye a un carácter que no esté entre [] (sólo 1) Doc[!123] coincide con Doc0, Doc4, Doc>, DocA, Docb.txt Doc[!123] NO coincide con Doc1, Doc2, Doc3 o Doc
echo * echo * echo "" echo ""	Muestra por pantalla el nombre de los ficheros del directorio ./ Evita la expansión. Muestra por pantalla un * Evita la expansión. Muestra por pantalla un * Evita la expansión. Muestra por pantalla un *

EJECUCIÓN DE COMANDOS

comando	Ejecuta el comando si tiene permisos de ejecución y está en el PATH
./comando	Ejecuta el comando si tiene permisos de ejecución y está en ./
exec comando	Ejecuta el comando sustituyendo el Shell actual y cerrándolo

comando a “b c”	Ejecuta el comando con dos parámetros: a y “b c” Los espacios blancos se descartan (si no se utiliza “, ‘ ó \) Todo el texto entrecomillado forma un único parámetro Antes de llamar al comando se resuelven las expansiones del bash
alias alias l1='ls -laF'	Muestra los alias Crea un alias para el comando l1

EJECUCIÓN ESPECIAL DE COMANDOS

a;b	Ejecuta el comando a y luego el b
a && b	Ejecuta a. Si no falla ejecuta después b
a b	Ejecuta a. Si falla ejecuta b
echo \$?	Muestra el código de salida del último comando ejecutado Código 0: finalización correcta Código distinto de 0: error (mirar el man del comando)
com1 ‘com2’	Sustituye la salida del comando2 y lo utiliza como parámetro para com1 No anidable
com1 \$(com2)	Idéntico al anterior Anidable: lista=\$(ls \$(cat directorios.txt))
comando& nohup comando&	Ejecuta el comando en segundo plano y continúa el bash Ejecuta el comando en segundo plano y no muere al terminar el bash
tee fichero	Lee la entrada estándar y escribe en la salida y en el fichero ls tee salida.txt grep f1
xargs comando	Ejecuta el comando tantas veces como palabras en la entrada cat nombre xargs mkdir La opción -l texto permite indicar la posición en el comando find /bin -size 2k xargs -l XXX cp XXX ./

VARIABLES

echo texto echo -e texto echo -n texto	Muestra por la salida el texto Interpreta los caracteres especiales No inserta un salto de línea al final del texto
num_var=valor	Asigna el valor a la variable num_var Sin espacios entre el = El nombre de la variable sin \$ Si valor es una variable sí llevaría \$: a=\$b
\$variable	Recupera el valor de la variable para utilizarlo
\${variable}	Sintaxis adecuada para evitar ambigüedades: a=1; b=2; ab=3; echo\${a}b
“\$variable”	En ocasiones hay que colocar “” para tener en cuenta los valores con espacios a=”1 3”; echo \$a; echo “\$a”
unset variable	Borra una variable unset a

export variable	<p>Mete una variable en el entorno</p> <p>Será visible por el Shell actual y sus hijos</p> <p>Por convención se utilizan nombres en mayúsculas</p> <p>Para hacer una variable global permanente hacer export en:</p> <p>~/.bash_profile (ó ~/.profile) #Al hacer login</p> <p>~/.bashrc #Al abrir cada nueva instancia de bash</p> <p>Para hacer una variable global permanente a todos los usuarios:</p> <p>/etc/bashrc</p> <p>/etc/profile</p> <p>/etc/ambiente</p>
env	Muestra la lista de variables de entorno de un usuario

GESTIÓN DE FICHEROS Y DIRECTORIOS

mkdir ruta_dir	Crea un directorio dada su ruta y nombre
mkdir -p ruta_dir	Crea un directorios y sus antecesores aunque no existan
rmdir ruta_dir	Borra un directorio vacío dada su ruta
file fichero	Indica el tipo del fichero utilizando ciertos patrones
file -s fich_especial	<p>Más información: man 5 magic</p> <p>Obtiene información adicional en ficheros especiales (dev...)</p>
touch ficheros	<p>Si no existe el fichero crea un nuevo fichero vacío</p> <p>Si existe modifica la fecha de acceso y mod por la actual.</p> <p>Para cambiarla por una fecha concreta:</p> <p>touch -t [YY]YYMMDDhhmm[.ss] ficheros</p>
stat fichero	Muestra la fecha de acceso y modificación del fichero
rm ruta	Borra ficheros
rmdir ruta	Borra directorios
rm -rf rutas	Borra directorios y ficheros recursivamente sin preguntar
cp ruta_origen ruta_dest	<p>Copia el fichero origen a destino</p> <p>Si en la ruta destino se indica el nombre se copia y renombra</p> <p>Si el destino existe y es un directorio lo copia dentro</p> <p>Si el destino existe y es un fichero lo sobrescribe</p>
cp rutas_origen ruta_dest	Copia múltiples ficheros de origen al destino
cp -r ruta_ori ruta_dest	Copia recursivamente ficheros y directorios de origen a destino
cp -rf ruta_ori ruta_dest	Copia recursivamente y fuerza la sobrescritura
cp -i ruta_ori ruta_dest	Copia y pregunta en caso de sobrescrituras
mv ruta_ori ruta_dest	<p>Mueve ficheros y directorios del origen al destino</p> <p>Si el directorio origen y destino es el mismo renombra</p>

COMPRESIÓN DE FICHEROS

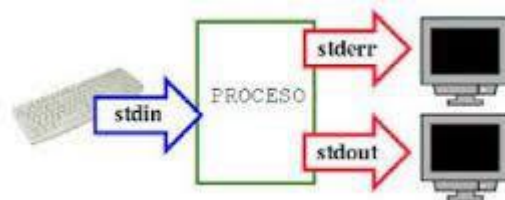
tar -xvzf file.tgz	Desempaqueta (x) y descomprime (z) mostrando información (v) el fichero (f)
tar -czvf file.tgz f1 f2 f3	Empaqueta (c) y comprime (z) en el fichero file.tgz todos los ficheros restantes
zip -r fichero.zip ruta	Comprime en zip la ruta en el fichero.zip
zip -r -[0-9] fichero.zip ruta	Indica el nivel de compresión
unzip fichero.zip	Descomprime fichero.zip en el directorio actual
unzip fichero.zip -d ruta	Descomprime el fichero.zip en la ruta indicada

unzip -l fichero.zip	Lista los archivos contenidos en fichero.zip
unzip -v fichero.zip	Lista los archivos de fichero.zip en modo detallado

REDIRECCIONES

Al ejecutar un comando se abren tres flujos:

- 0:** entrada estándar (**stdin**, por defecto el teclado)
- 1:** salida estándar (**stdout**, por defecto la pantalla)
- 2:** salida estándar de error (**stderr**, por defecto la pantalla)



Las redirecciones modifican el lugar al que se conectan estos flujos

comando >f	Envía la salida del comando al fichero f y si existe lo sobrescribe
comando >>f	Envía la salida del comando al final del fichero f
comando 2>f	Envía la salida de error al fichero f y lo sobrescribe
comando 2>>f	Envía la salida de error al final del fichero f
comando &> f	Envía la salida estándar y de error al fichero f y lo sobrescribe
comando &>> f	Envía la salida estándar y de error al final del fichero f
comando < f	Utiliza el contenido de f como entrada estándar
comando <<XXX texto1 texto2 XXX	Ejecuta el comando y le pasa por entrada estándar las líneas texto1 y texto2 XXX puede sustituirse por cualquier cadena, únicamente marca el comienzo y fin
comando 2>&1	Redirecciona la salida de error a la estándar comando >/dev/null 2>&1 == programa >/dev/null 2>/dev/null
comando 1>&2	Redirecciona la salida estándar a la de error echo "Esto se mostrará en stderr" 1>&2
comando1 comando2	Tubería: envía la salida del comando 1 a la entrada del comando 2 apropos lines grep "(1)"
/dev/null	Fichero de dispositivo virtual con nulos infinitos
/dev/urandom	Fichero de dispositivo virtual con aleatorios infinitos

TRATAMIENTO DE TEXTOS

- La mayoría de comandos de texto pueden recibir la entrada desde la entrada estándar (teclado o desde un comando mediante tubería) o desde un fichero como parámetro.
- A veces ambas: el carácter - se utiliza como parámetro y sustituye a la EE

ls car - f1 # El cat muestra la salida de ls y luego la de f1	
cat f1 [f2, f3...] cat -n f1	Muestra por pantalla el contenido del fichero f1, f2, f3... Numera las líneas
tac archivo	Igual que cat pero en orden inverso
less fichero	Visualiza el contenido de un fichero de forma paginada Funciona como el man
more fichero	Similar a less pero no permite moverse arriba/abajo
head f1 head -3	Muestra las primeras líneas de un fichero Muestra las 3 primeras líneas
tail tail -f	Muestra el final de un fichero Mantiene abierta el fichero y muestra las nuevas líneas añadidas
hexdump od	Visualiza el contenido de un fichero en hexadecimal y octal
sort f1 [f2, f3...] sort -r f1 sort -n f1 sort -k 3 f1 sort -k 3 -t ":" f1	Ordena las líneas de un fichero Orden inverso Orden numérico en lugar de alfabético Orden según el campo 3 (separador de campo " ") Orden según el campo 3 utilizando como separador de campo ":"
paste f1 f3 paste -d ":" f1 f2 cat f1 paste - f2 ls paste - - -	Fusiona f1 y f2 pegando las líneas de ambos Usa ":" como separador entre líneas (por defecto TAB) Sustituye - por la entrada estándar Divide en columnas
join f1 f2 join -1 3 -2 2 f1 f2 join -t ":" f1 f2	Combina las líneas de f1 con las de f2 según la coincidencia de un campo. Las líneas deben estar ordenadas Combina las líneas según el campo 3 de f1 y el campo 2 de f2 Usa : como separador de campo (por defecto espacio)
expand	Convierte tabulaciones en espacios (en 8 por defecto)
unexpand	Convierte espacios en tabulaciones
split [opciones] f1 split -b x f1 split -C x f1 split -l x f1	Divide contenido del fichero f1 en varios Divide en ficheros de x bytes Divide en ficheros de x bytes sin mezclar líneas Divide en ficheros de x líneas
tr [opciones] c1 tr abc def c1 tr -d "a" c1 tr -s "a" "b"	Cambia caracteres individuales de la EE (no de fichero) Cambia "a" por "d", "b" por "e" y c por "f" Borra todas las apariciones del carácter "a" Sustituye cada aparición del carácter "a" repetido por una sola "b" Puede usarse [:lower:] [:upper:] [:digit:] [alnum:]
uniq f1 uniq -d f1 uniq -c f1	Elimina las líneas duplicadas de f1. f1 tiene que estar ordenado Solo muestra las líneas duplicadas Muestra al lado de cada línea el número de repeticiones
nl f1 nl -b a f1 nl -b Pexpr	Numera las líneas de un fichero no vacías (ídem a cat -b f1) Numera todas las líneas (ídem a cat -n f1) Numera sólo las líneas que cumplen cierta ER no extendida
cut f1 cut -b n f1 cut -c 2-4 f1 cut -f 3 f1 cut -f 2,4-5 cut -d ":"	Extrae partes de f1 Extrae los bytes indicados de cada línea Extra los caracteres del 2 al 4 de cada línea Extrae la tercera columna Extrae las columnas 2 y de la 4 a la 5 Usa ":" como separador de campo (por defecto usa TAB)
wc f1	Cuenta las líneas, palabras y bytes de un fichero
awk	Procesa, filtra y reformatea texto

	<pre>awk '/patrón_busqueda/ {acción;acción;} archivo awk '/^UUID/ {print \$1;} archivo /etc/fstab awk 'BEGIN {FS=":";} /root/ {print \$1, \$2} /etc/passwd'</pre>
--	---

EXPRESIONES REGULARES

	<ul style="list-style-type: none"> Una expresión regular determina un conjunto de cadenas de caracteres Sirve para seleccionar cadenas de caracteres (nombres de archivos, partes de un texto, etc.) según un patrón. Pueden ser muy complejas, pudiendo combinar expresiones regulares elementales, que se equiparan con un solo carácter.
. a.b	Un carácter cualquiera, excepto un salto de línea Concuerda con axa, aab, abb, aSb, a#b...
[abc] Hol[abc]	Un carácter a, b o c Concuerda con Hola, Holb, Holc
^ '^m' [^abc] 1[^789]	Delimitador. Indica inicio de línea. Líneas que empiecen por "m" Si está en un corchete es una expresión negativa, cambia el sentido Un carácter que NO sea a, b o c Concuerda con 10,11,12,13,14,15,16
\$ 'a\$'	Delimitador. Indica final de línea Líneas que finalicen por "a"
[0-9][a-z][A-Z] Hol[a-c] 1[1-3]	Cualquier carácter del intervalo indicado. Funcionan con el código ASCII, si lo ponemos al revés [Z-A] no encuentra nada Coincide con Hola, Holb, Holc Coincide con 11, 12, 13
\	Hay que ponerlo antes de caracteres especiales (*,[\,^,\$,/)
* r*	Repetición, uno o más. Una o más apariciones del carácter "r"

EXPRESIONES REGULARES EXTENDIDAS

+ [0-9]+	1 o más ocurrencias Coincide con 0, 1, 9, 00, 99, 123, 415, 3423465345...
? [0-9]?	0 o una ocurrencia de la ER y no más Coincide con 0, 1, 2, 3... 9 y con la cadena vacía
r{n} r{n,} r{,m} r{n,m}	n ocurrencias de la ER r n o más ocurrencias de la ER r 0 o a lo sumo m ocurrencias de la ER r n o más ocurrencias de la ER r, pero a lo sumo m
r1 r2	la ER 1 o la ER2 (alternativa)
(r)	ER anidada
"r"	evita que los caracteres de la ER r sean interpretados por el shell

GREP

- Busca las líneas que coinciden con un patrón especificado
- Utiliza como entrada stdin o los ficheros pasados como parámetro
- Devuelve las líneas que contienen dicho patrón
- Como patrón utiliza ER no extendidas (o extendidas usando -E o usando egrep)
- Al recibir un ER es conveniente pasarla entre "" para que bash no intente resolverla

grep p f1

Busca en el fichero f1 las líneas coincidentes con el patrón p

c1 | grep p

Busca en la salida del comando c1 las líneas coincidentes con p

grep -i p f1	Ignora mayúsculas y minúsculas
grep -v p f1	Invierte el sentido de la ER (líneas no coincidentes)
grep -r p f1	Busca recursivamente en directorios
grep -E p f1	Utiliza ER extendidas
grep -l p f1	Sólo lista el nombre de los ficheros coincidentes (no las líneas)
grep -c p f1	Sólo cuenta las líneas coincidentes
grep -o p f1	Muestra solo la parte coincidente (no la línea completa)

SED

- Utiliza como entrada stdin o los ficheros pasados como parámetro y devuelve una versión modificada del texto según ciertos patrones y texto de remplazo
- Utiliza ER no extendidas por defecto (con -r para extendidas)
- Estructura general:
 - sed
 - Comilla simple
 - Cadena a buscar o números de línea
 - Instrucción
 - Argumentos (cadena para sustituir, por ejemplo)
 - Comilla simple
 - Ficheros
- Podemos encadenar más de una orden en sed separándolas con “;”

sed '1d' f1	Elimina la línea 1 del fichero f1
sed '/^\$/d' f1	Elimina líneas vacías (el ^marca una cadena vacía y el \$ fin de línea)
sed '/^#/d' f1	Elimina las líneas que empiezan por # de f1
sed "/er/d" f1	Elimina del fichero f1 toda ocurrencia de la ER er
sed -n '/[de]/p' f1	Muestra las líneas de f1 que empiezan por d o e
sed 's/xx/yy/' f1	Sustituye la cadena “xx” por “yy” en el fichero f1
sed 's/[0-9]+/&Kg/g f1	Añade el sufijo Kg a cualquier número que encuentre en f1
sed '1 iHola' f1	Inserta el texto f1 al principio (antes de la línea 1) del fichero f1
Más ejemplos	

NIVELES DE EJECUCIÓN

<ul style="list-style-type: none"> ○ Modos de operación de los sistemas operativos tipo Unix ○ Los niveles de ejecución existentes para cada distribución se encuentran en /etc/inittab ○ 7 niveles en Ubuntu. El 0 es apagado y el 6 reinicio ○ Según el nivel de ejecución, se ejecutan unas tareas u otras al inicio. Estas se corresponden con los scripts (enlaces simbólicos) existentes en cada uno de los directorios /etc/rcX.d 	
Nivel 0 (Alto)	Cierre del sistema (Apagado)
Nivel 1 (monousuario)	Sólo usuario root. No hay interfaz de red. Es un nivel para reparar problemas
Nivel 2 (multiusuario)	Multiusuario sin soporte de red
Nivel 3 (multiusuario con soporte de red)	Inicio normal
Nivel 4 (multiusuario con soporte de red)	No se usa por defecto, existe para poder personalizar el nivel 3 añadiendo nuevos servicios
Nivel 5 (multiusuario con inicio gráfico)	Nivel 3 + X11
Nivel 6 (reinicio)	Se reinicia el sistema

ENLACES

ENLACES SIMBÓLICOS

- Un enlace simbólico guarda un acceso directo con la ruta al archivo enlazado
- Pueden crearse entre distintos dispositivos
- Los permisos del enlace se muestran siempre como lrwxrwxrwx
- Sin embargo, los permisos efectivos son los del fichero real
- Al borrar el original el enlace sigue existiendo pero no apunta a nada
- Pueden crearse enlace simbólicos a directorios

ENLACES DUROS (hard links o enlaces físicos)

- En un enlace físico dos entradas de directorio apuntan a un mismo inodo (no se crea un nuevo inodo). Ambos son igual de auténticos (ls -li obtiene el número de inodo)
- No pueden crearse entre distintos sistemas de ficheros
- Al cambiar los permisos de uno cambian los del otro
- Al borrar uno no se borra el otro
- Casi ningún sistema de fichero soporta enlaces de referencia a directorios

ln -s origen destino	Crea un enlace simbólico del fichero origen al destino
ln origen destino	Crea un enlace físico del fichero origen al destino

ADMINISTRACIÓN DE SOFTWARE EN LINUX

- La mayoría de distribuciones gestionan el software mediante paquetes.
- **PAQUETE:** software comprimido y preparado para su instalación automatizada. El paquete incluye, además del software, la configuración, paquetes dependientes necesarios, ubicación donde debe instalarse, órdenes que deben ejecutarse para preparar el software antes y después de la instalación, etc.
- **TIPOS DE PAQUETES**
 - **DEB:** utilizado por distribuciones Debian, Ubuntu y derivadas.
 - Herramientas de administración: dpkg, apt-get, aptitude
 - **RPM:** utilizado por distribuciones RedHat y derivadas (Fedora, CentOS, Suse, etc.)
 - Herramientas de administración: rpm, yum
 - **PACMAN:** utilizado por la distribución ArchLinux
- **REPOSITORIO:** para instalar y encontrar los paquetes de forma sencilla se almacenan y agrupan en repositorios, normalmente accedidos remotamente.
 - Al instalar un paquete éste se busca en el repositorio, se descarga junto a sus paquetes dependientes y se instala de forma automatizada.
 - Cada distribución utiliza una lista de repositorios con el software disponible para instalar y adaptado a dicha distribución
 - No deben mezclarse repositorios de distintas distribuciones o incluir repositorios extraños
 - Si no se utiliza un repositorio es necesario descargar el fichero del paquete y sus paquetes dependientes y utilizarlos para la instalación local

HERRAMIENTAS PARA LA ADMINISTRACIÓN DE PAQUETES DEBIAN (UBUNTU)

dpkg	Gestiona los paquetes Debian (.deb) instalados localmente o descargados. No utiliza los repositorios
apt-get	Gestiona los paquetes .deb de los repositorios y locales Actúa como front-end de dpkg y back-end de algunas herramientas gráficas
apt-cache	Consulta información de los paquetes deb en los repositorios
aptitude	Alternativa a apt que facilita su uso: realiza acciones y propuestas sobre los paquetes a mayor nivel
synaptic	Herramienta gráfica para la gestión de paquetes Debian (también puede ser usado para RPM). Es un front-end para apt. Podemos instalar y eliminar software y modificar y actualizar nuestros repositorios
Centro software	Es la herramienta gráfica existente por defecto en Ubuntu. Hasta la versión 15 fue una aplicación propia desarrollada por Canonical, a partir de la 16 es la versión de GNOME. Al igual que synaptic permite gestionar la instalación y actualización de software así como administrar nuestros repositorios de programas.

DPKG

<ul style="list-style-type: none"> ○ Gestiona los paquetes Debian (.deb) instalados localmente o descargados ○ No utiliza los repositorios 	
dpkg -i fichero.deb	Instala el paquete .deb desde un fichero local
dpkg -l	Muestra la lista de paquetes instalados ii (instalado), rc (desinstalado y con configuración)
dpkg -l patrón	Muestra los paquetes instalados que coinciden con un patrón
dpkg -r paquete	Desinstala el paquete dado su nombre
dpkg -P paquete	Purga el paquete (lo desinstala y borra archivos de configuración)
dpkg -s paquete	Muestra la información sobre un paquete instalado
dpkg -I fichero.deb	Muestra información de un fichero .deb no instalado
dpkg -C	Muestra los paquete parcialmente instalados
dpkg -L paquete	Muestra los ficheros instalados por un paquete
dpkg -S ruta/fichero	Muestra los paquetes que incluyen el fichero/ruta
dpkg-reconfigure paquete	Vuelve a lanzar el script de postinstalación del paquete

APT-GET

<ul style="list-style-type: none"> ○ Gestiona los paquetes .deb de los repositorios y locales ○ La ubicación de los repositorios donde se encuentran los paquetes se configura en /etc/apt/sources.list (o los ficheros ubicados en /etc/apt/sources.list.d/*.list) ○ Utiliza una BD que contiene un índice con los paquetes disponibles, versión, dependencias, ubicación de los ficheros, etc. ○ Al instalar un paquete la BD debe encontrarse actualizada o se producirá un error. ○ Cada vez que se instala un paquete conserva el .deb en /var/cache/apt/archives/ ○ Una vez descargado el paquete .deb y sus dependencias se instala y se configura, 	
--	--

estando listo para su ejecución	
apt-get update	Resincroniza la BD de los paquetes disponibles
apt-get upgrade	Actualiza todos los paquetes del sistema (siempre que no se necesite instalar nuevos o eliminar paquetes)
apt-get -s upgrade	Simula una actualización
apt-get dist-upgrade	Actualiza los paquetes del sistema y si se requiere instala nuevas dependencias o las elimina
apt-get install paquete	Instala o actualiza el paquete desde los repositorios
apt-get install pkg=ver	Instala una versión específica del paquete
apt-get remove paquete	Elimina el paquete local instalado
apt-get autoremove	Elimina los paquetes huérfanos (dependencias que fueron instaladas automáticamente con un paquete ya eliminado) (aptitude lo hace por defecto)
apt-get check	Comprueba las dependencias rotas de los paquetes instalados
apt-get clean	Borra la cache de .deb (directorio /var/cache/apt/archives/*)
apt-get changelog paquete	Muestra la lista de cambios de las últimas versiones

APT-CACHE

○ Consulta información de los paquetes deb de los repositorios (instalados o no)	
apt-cache search paquete	Busca un paquete en la lista de paquetes de los repositorios
apt-cache show paquete	Muestra los campos de información de un paquete
apt-cache showpkg paquete	Muestra las versiones disponibles y su repositorio entre () y las dependencias
apt-cache depends paquete	Muestra las dependencias de un paquete
apt-cache rdepends paquete	Muestra las dependencias inversas de un paquete
apt-cache pkgnames	Muestra la lista de todos los paquetes que apt conoce
apt-file search ruta/fichero	Muestra los paquetes del repositorio que incluyen el fichero

APTITUDE

○ Alternativa a apt (apt-get, apt-cache, apt-file, etc.) que facilita su uso: realiza acciones y propuestas sobre los paquetes a mayor nivel en un único comando	
aptitude update	Resincroniza la BD de los paquetes disponibles
aptitude upgrade aptitude safe-upgrade	Obsoleto. Realiza un safe-upgrade Actualiza todos los paquetes del sistema (siempre que no necesite instalar nuevos o eliminar paquetes)
aptitude full-upgrade	Actualiza los paquetes del sistema y si se requiere instala nuevas dependencias o las elimina
aptitude install paquete	Instala o actualiza el paquete desde los repositorios
aptitude search paquete aptitude search “~U” aptitude search “~U nombre”	Muestra las dependencias inversas de un paquete Busca los paquetes susceptibles de actualización ídem pero que contengan el nombre indicado
aptitude remove paquete	Elimina el paquete local instalado
aptitude purge paquete	Purga el paquete (desinstala y borra su configuración)

FICHEROS DE CONFIGURACIÓN DE REPOSITORIOS

/etc/apt/sources.list	Fichero con las direcciones de los repositorios disponibles Este fichero es utilizado por apt para saber dónde encontrar los paquetes a descargar y actualizar la BD de índices
/etc/apt/sources.list.d/*	Cada fichero .list en este directorio puede contener direcciones de repositorios al igual que sources.list Sirve para separar y agrupar los repositorios
apt-key add fichero_clave	Añade la clave pública de un repositorio al almacén de confianza
/var/lib/apt/lists/*	Ficheros con la BD de los paquetes de cada repositorio Mostrar todos los paquetes de un repositorio concreto grep “^Package” /var/lib/apt/lists/(nombre_repo)*Packages
/var/cache/apt/archives/*	Directorio con los .deb guardados al instalar con apt

AÑADIR UN NUEVO REPOSITORIO

- Incluir en sources.list la línea deb o deb-src con la URL del repositorio:
deb http://download.virtualbox.org/virtualbox/debian jessie contrib
- Descargar el fichero con la clave pública del repositorio:
wget http://<url_fichero_clave_publica>
- Añadir la clave pública apt-key add <fichero_clave_publica>
- Actualizar la BD de índices
apt-get update

HERRAMIENTAS PARA LA ADMINISTRACIÓN DE PAQUETES RPM

rpm	Gestiona los paquetes RedHat (.rpm) instalados localmente o descargados No utiliza los repositorios
yum	Gestiona los paquetes rpm de los repositorios y locales

RPM

rpm -i fichero.rpm	Instala el paquete .rpm desde un fichero local
rpm -e paquete	Desinstala un paquete
rpm -U paquete	Reinstala un paquete
rpm -F fichero.rpm	Actualiza sólo si existe una versión anterior
rpm -qa	Muestra la lista con todos los paquetes instalados
rpm -qi paquete	Muestra información sobre un paquete local
rpm -qf fichero	Muestra el paquete al que pertenece el fichero
rpm -ql paquete	Muestra los ficheros instalados por un paquete

YUM

yum install paquete	Instala el paquete desde las fuentes de repositorios
yum remove paquete	Elimina el paquete local instalado
yum update paquete	Actualiza el paquete desde las fuentes de repositorios
yum search patrón	Busca paquetes en los repositorios según un patrón
yum info paquete	Muestra información sobre un paquete instalado o no
/var/cache/yum	Directorio con los .rpm guardados al instalar con yum

INSTALACIÓN MANUAL DE SOFTWARE

- Al instalar software que no se encuentra en un paquete se utiliza la orden make que se encarga de realizar los pasos preconfigurados para la compilación e instalación
- Es habitual que ocurran múltiples fallos: falta de dependencias, bibliotecas, etc.
- PASOS INSTALACIÓN
 - tar -xzf archivo_software.tgz #Descomprime el software
 - ./configure #Ejecuta un script que comprueba el sistema y las dependencias
 - make
 - make install
- En otros casos el software se distribuye con un instalador que únicamente hay que ejecutar (por ejemplo Guest Additions de VB, el paquete XAMPP, etc.):
 - chmod +x instalador.bin
 - ./instalador.bin