Cómo pasar de DOS/Windows a Linux

Por Guido Gonzato *BORRAMEguido@ibogeo.df.unibo.it* (Elimine el BORRAME)
Traducido por David Marín Carreño, *davefx@bigfoot.com* v1.3.2 22 de Febrero de 1999. Traducción: 8 de Junio de 1999.

Este documento *Cómo* está dedicado a todos los (pronto ex-)usuarios de DOS y Windows que acaban de decidir pasarse a Linux, el clónico gratuito de UNIX. El propósito de este documento es ayudar al lector a trasladar su conocimiento de DOS y Windows al entorno Linux, así como dar indicaciones acerca de intercambiar archivos y recursos entre ambos sistemas operativos.

Índice General

1 Introducción **2**

1.1 ¿Es Linux adecuado para usted? 2

1.2 Ya está. Ahora cuénteme más. 3

1.2.1 Conceptos introductorios 4

1.2.2 Obtención de ayuda 4

1.3 Convenciones 5

2 Para el Impaciente **5**

3 Encontrándose con bash **6**

4 Ficheros y Programas **6**

4.1 Ficheros: Nociones preliminares 6

4.2 Enlaces simbólicos 7

4.3 Permisos y Propiedades 7

4.4 Ficheros: órdenes equivalentes 9

4.5 Ejecución de programas: Multitarea y Sesiones 10

4.6 Ejecución de programas en sistemas remotos 11

5 Gestión de Directorios **12**

5.1 Directorios: Nociones preliminares 12

5.2 Permisos de directorios. 12

5.3 Equivalencia de órdenes 12

6 Disquetes, discos duros y similares. **13**

6.1 Administración de dispositivos en «modo DOS» 13

6.2 Administración de dispositivos en «modo UNIX» 13

6.3 Copias de seguridad 15

7 ¿Qué pasa con Windows? 15

8 Personalización del sistema 16

8.1 Ficheros de inicialización del sistema 16

9 Ficheros de inicialización de programas 17

10 Trabajo en red: conceptos 17

11 Un poco de programación 18

11.1 Rutinas del Intérprete de Comandos: Ficheros .BAT con esteroides 18

11.2 Sus programas en C 19

12 El restante 1% 20

12.1 Utilización de `tar` y `gzip` 21

12.2 Instalación de aplicaciones 21

12.3 Trucos imprescindibles 22

12.4 ¿Dónde encontrar aplicaciones? 22

12.5 Unas cuantas cosas que Usted no podía hacer 22

12.6 Practicando UNIX bajo DOS/Windows 24

12.7 Extensiones comunes y programas relacionados 24

12.8 Conversión de ficheros 25

12.9 Entornos ofimáticos gratuitos 26

13 Fin, por ahora 26

13.1 Copyright 26

13.2 Agradecimientos 26

13.3 Traducción 27

14 Anexo: El INSFLUG 27

1 Introducción

1.1 ¿Es Linux adecuado para usted?

¿Quiere pasar del mundo DOS (esta definición incluye Windows) a Linux? Buena idea: Linux es técnicamente superior a DOS, Windows 95 e incluso Windows NT. Pero cuidado: puede no ser útil para usted. Estas son las diferencias principales entre DOS y Linux:

- Windows ejecuta Microsoft Office y muchos juegos; está claro que es fácil de instalar y configurar; es notablemente inestable; su rendimiento es pobre; los bloqueos del ordenador son frecuentes.
- Linux ejecuta StarOffice, mucho software técnico y unos pocos juegos; puede ser difícil de instalar y configurar; es sólido como una roca; rinde al máximo; los bloqueos de la máquina son muy raros.

De usted depende el decidir qué necesita. Linux le da potencia, pero lleva algún tiempo aprender a aprovecharla. Es decir, si necesita software comercial, o si no le apetece aprender órdenes y conceptos nuevos, sería mejor que buscara en otra parte. Sepa que muchos novatos lo dejan debido a las dificultades iniciales.

Se está trabajando para hacer Linux más facil de usar, pero *no espere ser un experto en Linux a menos que lea mucha documentación y lo use durante al menos unos meses*. Linux no le dará resultados instantáneos. A pesar de estas advertencias, tengo un 100% de seguridad en que si usted es el usuario correcto, entonces encontrará en Linux su Nirvana informático. Por cierto, Linux + DOS/Win pueden coexistir sin problemas en la misma máquina.

Requisitos previos para la lectura de este Cómo. Asumiré que:

- conoce las órdenes y conceptos básicos del DOS;
- Linux, posiblemente con el sistema X Window (X11 para abreviar), está adecuadamente instalado en su ordenador;
- su intérprete de órdenes (el equivalente de `COMMAND.COM`) es `bash`;

A menos que se especifique lo contrario, toda la información de este trabajo está dirigida al antiguo DOS. Hay información acerca de Windows en algunos lugares de este documento, pero tenga en cuenta que Windows y Linux son completamente distintos, al contrario que DOS, que es una especie de hermano pobre de UNIX.

Dese cuenta también de que este trabajo no es ni una guía de configuración ni un manual completo para los no iniciados.

La última versión de este documento está disponible en muchos formatos en *ftp://sunsite.unc.edu/pub/Linux/docs/howto*.

1.2 Ya está. Ahora cuénteme más.

Ha instalado Linux y los programas que necesita en su ordenador. Se ha proporcionado una cuenta (si no, ¡teclee `adduser suNombre` *ahora mismo*!) y Linux está ejecutándose. Acaba de introducir su nombre y su contraseña, y ahora está mirando a la pantalla pensando... *¿Y ahora qué?*

Bien, no se desespere. Puede hacer casi las mismas cosas que solía hacer con DOS/Win, y muchas más. Si estuviera ejecutando DOS/Win en lugar de Linux, realizaría alguna de las siguientes tareas:

- ejecutar programas y crear, copiar, ver, borrar, imprimir, renombrar ficheros;
- cambiar de directorio, crearlos, borrarlos y listar sus contenidos;
- formatear disquetes y copiar ficheros de/hacia ellos;
- configuración del sistema;
- navegar por Internet;
- escribir ficheros `.BAT` y programas en su lenguaje favorito;
- el restante 1%.

Estará contento al saber que todas esas tareas pueden llevarse a cabo bajo Linux de una manera similar a como se hacen bajo DOS. Bajo DOS, el usuario medio usa muy pocos de las más de 100 órdenes disponibles: lo mismo, hasta cierto punto, sucede con el Linux.

1.2.1 Conceptos introductorios

La mejor manera de aprender algo es mojándose los pies. Desde aquí le animo a experimentar y «jugar» con Linux: a menos que entre como usuario «root», no puede llegar a dañar el sistema de esa manera. Unos cuantos conceptos:

- Primero, cómo salir de Linux de manera segura. Si ve una pantalla en modo texto, pulse <Ctrl-Alt-Supr>, espere a que el sistema realice unas tareas y le diga que todo está bien, y entonces apague el ordenador. Si está trabajando bajo el sistema X Window, pulse primero <Ctrl-Alt-Retroceder>, y después <Ctrl-Alt-Supr>. *Nunca* apague o reinicie el ordenador directamente: podría dañar el sistema de ficheros;
- Al contrario que en DOS o Windows, Linux tiene mecanismos de seguridad intrínsecos, debido a su naturaleza multiusuario. Los ficheros y los directorios tienen permisos asociados, y por lo tanto el usuario normal puede no tener acceso a alguno de ellos; vea la sección 4.3 (Permisos). DOS y Windows, por el contrario, le dejarían machacar todo el contenido de su disco duro;
- Hay un usuario especial denominado «root»: el administrador del sistema, que tiene todo el poder sobre la máquina. Si usted trabaja con su propio PC, también será root. Trabajar como root es *peligroso*: cualquier error puede dañar seriamente o destruir el sistema de la misma manera que con DOS/Win. No trabaje como root a menos que sea absolutamente necesario;
- Mucha de la complejidad de Linux viene de su extrema configurabilidad: prácticamente cada característica y aplicación puede ser configurada mediante uno o más archivos de configuración. La complejidad es el precio a pagar por la potencia;
- La redirección y el entubamiento (*piping*) son características «secundarias» de DOS. En Linux pasan a ser muy importantes y mucho más potentes. Se pueden enlazar órdenes para ejecutar tareas complejas. Le recomiendo sobremanera que aprenda a usarlas.

1.2.2 Obtención de ayuda

Hay muchas maneras de conseguir ayuda con Linux. Las más importantes son:

- *Leyendo la documentación* Aunque el documento CÓMO que está leyendo puede servir como introducción a Linux, hay unos cuantos libros que debería leer: *Linux: Instalación y Primeros Pasos* de Matt Welsh (*http://sunsite.unc.edu/mdw/LDP/gs/gs.html*), la *Guía del Usuario de Linux* de Larry Greenfield (*http://sunsite.unc.edu/pub/Linux/docs/linux-doc-project/users-guide*), y la lista PUF de Linux (respuestas a Preguntas de Uso Frecuente) (*http://sunsite.unc.edu/mdw/FAQ/Linux-FAQ.html*. Todos ellos pueden conseguirse en *http://lucas.hispalinux.es* en su versión traducida. Mantenga un sentimiento de culpabilidad hasta haber leído al menos uno de ellos;
- La documentacion de los paquetes instalados en la máquina puede encontrarse usualmente en los directorios bajo `/usr/doc/`;
- Para conseguir ayuda acerca de las «órdenes internas» del intérprete de órdenes, teclee `help` o, mejor, `man bash` o `info bash`;
- Para conseguir ayuda acerca de una orden, teclee `man orden`, lo cual invoca a la página del manual («man») correspondiente a `orden`. Alternativamente, teclee `info orden`, lo cual mostrará la página info relativa a `orden`. info es un sistema de documentacion basado en hipertexto. Quizá no es intuitivo en un principio. Finalmente, puede intentar con `apropos orden` o `whatis orden`. Con todas estas órdenes, pulse 'q' para salir.

- Finalmente, en Internet: el lugar más adecuado para conseguir ayuda es Usenet, en el foro *news:comp.os.linux.setup*. Por favor, no me envíe mensajes solicitando ayuda, porque tengo bastante trabajo.

1.3 Convenciones

A lo largo de este documento, los ejemplos seguirán el siguiente formato: $<\ldots>$ significa algo que debe ser especificado, mientras que `[...]` es algo opcional. Ejemplo:

```
$ tar -tf <fichero.tar> [> fichero_redir]
```

Debe indicar `fichero.tar`, pero la redirección a `fichero_redir` es opcional.

Desde ahora «LPM» significa «para más información Lea las Páginas del Manual». No puedo dejar de recordar lo importante que es leer la documentación. «DosWin» significa «DOS/Windows».

Cuando el simbolo de la línea de órdenes en un ejemplo sea '#', la orden solo puede ser ejecutada por el root.

2 Para el Impaciente

¿Quiere empezar ya? Eche un vistazo a esta tabla:

```
DOS                     Linux                          Notas
------------------------------------------------------------------------------

ATTRIB (+-)attr fich    chmod <modo> fich              completamente distinto
BACKUP                  tar -Mcvf dispositivo dir/     ídem
CD nombredir\           cd nombredir/                  casi la misma sintaxis
COPY fich1 fich2        cp fich1 fich2                 idem
DEL fichero             rm fichero                     cuidado no hay undelete
DELTREE directorio      rm -R directorio/              idem
DIR                     ls                             no exact. misma sintaxis
DIR fich /S             find . -name fich              completamente distinto
EDIT fichero            vi fichero                     creo que no le gustara
                        joe fichero                    mas parecido al EDIT
EDLIN fichero           ed fichero                     olvídelo
FORMAT                  fdformat
                        mount, umount                  sintaxis muy distinta
HELP orden              man orden                      misma filosofia
                        info orden
MD directorio           mkdir directorio/              casi la misma sintaxis
MOVE fich1 fich2        mv fich1 fich2                 idem
NUL                     /dev/null                      idem
PRINT fichero           lpr fichero                    idem
PRN                     /dev/lp0,
                        /dev/lp1                       idem
RD directorio           rmdir directorio/              idem
REN fich1 fich2         mv fich1 fich2                 no para varios ficheros
RESTORE                 tar -Mxpvf device              sintaxis distinta
TYPE fichero            less fichero                   mucho mejor
WIN                     startx                         ¡polos opuestos!
```

Si necesita más que una tabla de órdenes, continúe con las secciones siguientes.

3 Encontrándose con bash

Buenas noticias: con Linux usted escribe mucho menos en la línea de órdenes, porque el intérprete `bash` escribe por usted siempre que sea posible, y goza de muy buenas características de edición de línea. Para empezar, la tecla flecha-arriba llama a las últimas líneas ejecutadas; pero aún hay más. Pulsar <TAB> completa los nombres de ficheros y directorios, por lo que teclear

```
$ ls /uTABloTABbTAB
```

es como teclear

```
$ ls /usr/local/bin
```

Si hubiera ambigüedades, como al teclear

```
$ ls /uTABloTABiTAB
```

`bash` para porque no sabe si usted se refiere a `/usr/local/info` o a `/usr/local/include`. Proporcione más caracteres y pulse <TAB> de nuevo.

Otras combinaciones de teclas útiles son <ESC-BACKSPACE> que borra una palabra a la izquierda, mientras que <ESC-D> borra una palabra a la derecha; <ESC-F> mueve el cursor una palabra a la derecha, <ESC-B> a la izquierda; <CTRL-A> mueve el cursor al principio de la línea, <CTRL-E> al final. La tecla <ALT> es equivalente a <ESC>.

Suficiente por ahora. Una vez que se acostumbre a estas combinaciones de teclas, encontrará bastante molesto el intérprete de órdenes de DOS.

4 Ficheros y Programas

4.1 Ficheros: Nociones preliminares

Linux tiene una estructura de directorios y ficheros muy similar a la de DOSWin. Los ficheros tienen nombres que obedecen unas normas especiales, están guardados en directorios y algunos son ejecutables, y entre éstos, la mayoría tiene opciones en la línea de órdenes. Incluso puede utilizar comodines, redirección y tuberías. Sólo hay unas pocas diferencias:

- Bajo DOS, los nombres de ficheros están en el llamado formato 8.3; por ejemplo `NOTENOUG.TXT`. Bajo Linux podemos hacerlo mejor. Si instaló Linux usando un sistema de ficheros tal como ext2 o umsdos, puede utilizar nombres más largos (hasta 255 caracteres), y con más de un punto en ellos: por ejemplo, `Este_es.un.nombre_de.fichero.MUY.largo`. Fíjese que he utilizado tanto mayúsculas como minúsculas: en efecto...
- Las mayúsculas y las minúsculas en nombres de ficheros u órdenes son distintas. Por lo tanto, `FICHERO.tar.gz`, y `fichero.tar.gz` son dos ficheros distintos. `ls` es una orden, `LS` es un error;
- Usuarios de Windows, cuidado al usar nombres de ficheros largos bajo Linux. Si un nombre de fichero contiene espacios (no recomendado, pero posible), debe encerrar el nombre de fichero entre comillas dobles cada vez que se refiera a él. Por ejemplo:

```
$ # la siguiente orden crea un directorio llamado "Mis ficheros antiguos"
$ mkdir "Mis ficheros antiguos"
$ ls
Mis ficheros antiguos          bin            tmp
```

No deben usarse ciertos caracteres: algunos son `!*$&#`.

- No hay extensiones obligadas como .COM y .EXE para los programas, o .BAT para los ficheros de procesamiento por lotes. Los ficheros ejecutables se marcan con un asterisco cuando se ejecuta la orden `ls -F`. Por ejemplo:

```
$ ls -F
cindy.jpg    cjpg*    Soy_un_directorio/  mi_1er_script*  old~
```

Los ficheros `cjpg*` y `mi_1er_script*` son «programas» ejecutables. Bajo DOS, las copias de seguridad de los ficheros suelen acabar en `.BAK`, mientras que bajo Linux acaban con un virgulilla ~. Un fichero cuyo nombre empieza con un punto es considerado como oculto. Ejemplo: el fichero `.Soy.un.fichero.oculto` no se mostrará a un orden `ls`;

- Las opciones de los programas bajo DOS se obtienen con `/opción`. En Linux se obtienen con `-opción` o `-opción`. Ejemplo: `dir /s` se convierte en `ls -R`. Fíjese en que muchos programas DOS (como `PKZIP` o `ARJ`) utilizan opciones de tipo Unix.

Ahora puede saltar a la sección 4.4 (de equivalencias), pero yo de usted seguiría leyendo.

4.2 Enlaces simbólicos

Unix tiene un tipo de fichero que no existe bajo DOS: el enlace simbólico. Puede pensar en él como un puntero o enlace a un fichero o a un directorio y que puede utilizarse en lugar del fichero o del directorio al que apunta; es similar a los «Accesos Directos» de Windows. Ejemplos de enlaces simbólicos son `/usr/X11`, que apunta a `/usr/X11R6`; `/dev/modem`, que apunta a `/dev/ttyS0` o a `/dev/ttyS1`.

Para crear un enlace simbólico:

```
$ ln -s <fichero_o_directorio> <nombre_del_enlace>
```

Ejemplo:

```
$ ln -s /usr/doc/g77/DOC g77manual.txt
```

Ahora puede referirse a `g77manual.txt` en lugar de `/usr/doc/g77/DOC`. Los enlaces aparecen en los listados de directorio así:

```
$ ls -F
g77manual.txt@
$ ls -l
l(muchas cosas...)            g77manual.txt -> /usr/doc/g77/DOC
```

4.3 Permisos y Propiedades

Los ficheros y directorios de DOS tienen los siguientes atributos: `A` (fichero), `H` (oculto), `R` (sólo-lectura), y `S` (sistema). Sólo `H` y `R` tienen sentido bajo Linux: los ficheros ocultos comienzan con un punto, y los de sólo lectura, tienen activado el permiso `r`, pero nada más.

Bajo Unix un fichero tiene «permisos» y un propietario, que pertenece a un «grupo». Mire este ejemplo:

```
$ ls -l /bin/ls
-rwxr-xr-x  1  root  bin  27281 Aug 15 1995 /bin/ls*
```

El primer campo contiene los permisos del fichero `/bin/ls`, que pertenece a `root`, del grupo `bin`. Dejando la información restante a un lado (el libro de Matt esta ahí para ese propósito), sólo recordaré lo que significa `-rwxr-xr-x` (de izquierda a derecha):

`-` es el tipo de fichero (`-` = fichero normal, `d` = directorio, `l` = enlace, etc.); `rwx` son los permisos del propietario del fichero (leer, escribir, ejecutar); `r-x` son los permisos para el grupo del propietario del fichero (leer y ejecutar); (no tocaré el concepto de grupo, puede pasar sin él mientras sea un novato ;-) `r-x` son los permisos para todos los demás usuarios (leer, ejecutar).

Esa es la razón de que no pueda borrar el fichero `/bin/ls` a menos que sea root: no tiene el permiso de escritura para hacer eso. Para cambiar los permisos de un fichero, la orden es:

```
$ chmod <quiénXperm> <fichero>
```

donde `quién` podría ser:

- `u` usuario, que es el propietario,
- `g` (grupo),
- `o` (otros).

`X` puede ser tanto `+` como `-`, y `perm` puede ser:

- `r` (lectura),
- `w` (escritura)
- `x` (ejecución).

Ejemplos:

```
$ chmod u+x fichero
```

esto habilita el permiso de ejecución para el propietario del fichero. Atajo: `chmod +x fichero`.

```
$ chmod go-wx fichero
```

esto quita el permiso de escritura y de ejecución para todo el mundo menos al usuario.

```
$ chmod ugo+rwx fichero
```

esto le da a todo el mundo el permiso de lectura, escritura y ejecución.

```
$ chmod +s fichero
```

esto convierte al fichero en *setuid* o *suid*; esto es, un fichero que todo el mundo puede ejecutar con los privilegios de su dueño. Normalmente, se encontrará con archivos suid del root; estos son, a menudo, ficheros importantes del sistema, como el servidor X.

Una manera más corta de referirse a los permisos es con números: `rwxr-xr-x` puede ser expresado como `755` (cada letra corresponde a un bit: `--` es 0, `-x` es 1, `-w-` es 2, `-wx` es 3...). Parece difícil, pero con algo de práctica el concepto se domina. root, el administrador y superusuario, puede cambiar los permisos de los ficheros de todo el mundo. Hay mucha más información acerca de esto: LPM.

4.4 Ficheros: órdenes equivalentes

A la izquierda, las órdenes de DOS; a la derecha, sus equivalentes en Linux.

```
COPY:         cp
DEL:          rm
MOVE:         mv
REN:          mv
TYPE:         more, less, cat
```

Operadores de redirección y de tuberías:

$<$ $>$ $>>$ $|$

Comodines: `*` `?`

```
nul: /dev/null

prn, lpt1:  /dev/lp0 o /dev/lp1; lpr
```

EJEMPLOS

```
DOS                                    Linux
---------------------------------------------------------------------

C:\GUIDO>ATTRIB +R FICH.TXT            $ chmod 400 fich.txt
C:\GUIDO>COPY JOE.TXT JOE.DOC          $ cp joe.txt joe.doc
C:\GUIDO>COPY *.* TOTAL                $ cat * > total
C:\GUIDO>COPY FRACTALS.DOC PRN         $ lpr fractals.doc
C:\GUIDO>DEL TEMP                      $ rm temp
C:\GUIDO>DEL *.BAK                     $ rm *~
C:\GUIDO>MOVE PAPER.TXT TMP\           $ mv paper.txt tmp/
C:\GUIDO>REN PAPER.TXT PAPER.ASC       $ mv paper.txt paper.asc
C:\GUIDO>PRINT LETTER.TXT              $ lpr letter.txt
C:\GUIDO>TYPE LETTER.TXT               $ more letter.txt
C:\GUIDO>TYPE LETTER.TXT               $ less letter.txt
C:\GUIDO>TYPE LETTER.TXT > NUL         $ cat letter.txt > /dev/null
        no disponible                  $ more *.txt *.asc
        no disponible                  $ cat section*.txt | less
```

Notas:

- `*` es más inteligente bajo Linux: `*` equivale a todos los ficheros excepto los ocultos; `.*` equivale a todos los ficheros ocultos; `*.*` equivale sólo a aquellos ficheros que tienen un punto en medio del nombre, seguido de otros caracteres; `p*r` equivaldría tanto a `peor` como a `por`; `*c*` equivaldría tanto a `pecado` como a `peca`.
- Cuando utilice `more`, pulse espacio para ir leyendo a través del fichero, `q` o Ctrl-C para salir. `less` es más intuitivo y permite utilizar las teclas del cursor;
- No hay `UNDELETE`, así que *piénselo dos veces* antes de borrar cualquier cosa;
- Además de los $<$ $>$ $>>$ del DOS, Linux tiene el operador `2>` para redirigir los mensajes de error (stderr); más aún, el operador `2>&1` redirige stderr a stdout (la salida estándar), mientras que `1>&2` redirige stdout a stderr;

- Linux tiene otro comodín: los corchetes `[]`. Usar `[abc]*` equivale a los ficheros que empiezan por `a`, por `b` o por `c`; `*[I-N,1,2,3]` equivale a los ficheros que acaban por `I, J, K, L, M, N, 1, 2, 3`;
- No hay un `RENAME` como en DOS; esto es, `mv *.xxx *.yyy` no funciona. Un orden similar a REN está disponible en *ftp://sunsite.unc.edu/pub/Linux/utils/file*;
- Use `cp -i` y `mv -i` para ser avisado cuando un fichero vaya a ser sobreescrito.

4.5 Ejecución de programas: Multitarea y Sesiones

Para ejecutar un programa, escriba su nombre tal y como lo haría bajo DOS. Si el directorio —vea la sección 5 (Directorios)— donde el programa está guardado está incluido en la variable de entorno `PATH` —sección 8.1 (Inicialización)— el programa comenzará a ejecutarse. Excepción: al contrario que bajo DOS, en Linux un programa localizado en el directorio actual no se ejecutará a manos que el directorio actual (simbolizado por «`.`») esté incluido en el PATH. Para evitarlo, suponiendo que el programa se llame `prog`, teclee `./prog`.

Este es el aspecto típico de una línea de órdenes:

```
$ orden [-o1 [-o2] ... [-on]] [par1 [par2] ... [parn]]  [< input] [> output]
```

donde `-o1`, ..., `-on` son las opciones del programa, `par1`, ..., `parn` son los parámetros del programa. Puede encadenar varias órdenes en la línea de órdenes:

```
$ orden1 ; orden2 ; ... ; ordenN
```

Esto es todo acerca de ejecutar órdenes, pero es fácil ir un paso más allá. Una de las principales razones para usar Linux es que es un sistema operativo multitarea —puede ejecutar varios programas (a partir de ahora, procesos) a la vez—. Puede lanzar procesos en segundo plano (*background*) y seguir trabajando inmediatamente. Más aún, Linux permite tener varias sesiones abiertas simultáneamente: es como tener muchos ordenadores en los que trabajar a la vez.

- Para cambiar a la sesión 1..6 de las consolas virtuales, presione <ALT-F1> ... <ALT-F6>
- Para comenzar una nueva sesión en la misma consola virtual sin dejar la actual, teclee `su - <nombre_de_usuario>`. Ejemplo: `su - root`. Esto es útil, por ejemplo, cuando se necesita realizar una tarea que sólo el root puede hacer.
- Para acabar una sesión, teclee `exit`. Si hay trabajos parados (ver más abajo), será avisado.
- Para lanzar un proceso en segundo plano, añada un '`&`', al final de la línea de órdenes:

```
$ nomprog [-opciones] [parametros] [< input] [> output] &
[1] 123
```

- el intérprete de órdenes identifica el proceso con un número de trabajo (p.e. `[1]`; ver más abajo), y con un PID (123 en nuestro ejemplo).
- Para ver cuántos procesos hay, teclee `ps ax`. Esto generará una lista de procesos actualmente en ejecución.
- Para matar (finalizar) un proceso, teclee `kill <PID>`. Puede que necesite matar un proceso cuando no sepa cómo cerrarlo de la manera correcta... ;-). A veces, sólo podrá matar un proceso con `kill -SIGKILL <PID>`.

  Además, el intérprete de órdenes permite suspender temporalmente (parar) un proceso, mandar un proceso al segundo plano, y traer un proceso del segundo plano al primer plano. En este contexto, los procesos son denominados «trabajos» (jobs).

- Para ver cuántos trabajos hay, teclee `jobs`. Aquí los trabajos son identificados por su número de trabajo, no por su PID.
- Para parar un proceso ejecutándose en primer plano pulse <CTRL-C> (no siempre funciona).
- Para suspender un proceso ejecutándose en primer plano, pulse <CTRL-Z> (ídem al anterior).
- Para mandar un proceso suspendido al segundo plano (convirtiéndolo en trabajo), teclee `bg <trabajo>`.
- Para traer un trabajo al primer plano, teclee `fg <%trabajo>`. Para traer al primer plano el último trabajo enviado al segundo plano, simplemente teclee `fg`.
- Para matar un trabajo, teclee `kill <%trabajo>` donde `trabajo` puede ser 1, 2, 3...

Usando estas órdenes puede formatear un disco, comprimir un puñado de ficheros, compilar un programa, y descomprimir un fichero simultáneamente, y todavía tener la línea de órdenes a su disposición. Inténtelo con Windows, sólo para ver la diferencia de prestaciones (siempre que no se le cuelgue, claro).

4.6 Ejecución de programas en sistemas remotos

Para ejecutar un programa en una máquina remota cuyo nombre es `maquina.remota.edu`, teclee:

```
$ telnet maquina.remota.edu
```

Tras introducir su nombre de usuario y contraseña, arranque su programa favorito. Obviamente, debe tener una cuenta en la máquina remota.

Si tiene X11, puede incluso ejecutar una aplicación X en un ordenador remoto, mostrándolo en su pantalla de X. Supongamos `maquina.remota.edu` la máquina X remota y `sistema.linux.yo` su máquina Linux. Para ejecutar desde `sistema.linux.yo` un programa X que reside en `maquina.remota.edu`, haga lo siguiente:

- arranque X, arranque un xterm o un emulador de terminal equivalente, y después teclee:

  ```
  $ xhost +maquina.remota.edu
  $ telnet maquina.remota.edu
  ```

- tras introducir su nombre y contraseña, teclee:

  ```
  remote:$ DISPLAY=sistema.linux.yo:0.0
  remote:$ programa &
  ```

  (en vez de `DISPLAY...`, puede que tenga que escribir `setenv DISPLAY sistema.linux.yo:0.0`. Depende del intérprete de órdenes remoto).

Y.. ¡voila! Ahora `programa` se iniciará en `maquina.remota.edu` y se mostrará en su máquina. Aunque mejor no intente esto con un módem, porque sería demasiado lento para poder utilizarse. Además: es un método muy inseguro: por favor, lea el «mini-COMO de Aplicaciones X Remotas» en *`http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps`*.

5 Gestión de Directorios

5.1 Directorios: Nociones preliminares

Hemos visto las diferencias entre los ficheros de DOSWin y Linux. Entre directorios, bajo DOSWin el directorio raíz es \ y bajo Linux es `/` . De manera similar, los directorios anidados se separan mediante \ en DOSWin y mediante `/` en Linux. Ejemplo de rutas de fichero:

```
DOS:    C:\PAPERS\GEOLOGY\MID_EOC.TEX
Linux:  /home/guido/papers/geology/middle_eocene.tex
```

Como es normal, `..` es el directorio padre y `.` es el directorio actual. Recuerde que el sistema no le dejará hacer `cd`, `rd` o `md` donde usted quiera. Cada usuario tiene sus propias cosas en un directorio llamado 'home', directorio personal o de usuario, cedido por el administrador del sistema; por ejemplo, en mi PC mi directorio personal es `/home/guido`.

5.2 Permisos de directorios.

Los directorios también tienen permisos. Lo que hemos visto en la sección 4.3 (Permisos) también rige para los directorios (usuario, grupo, y otros). Para un directorio, `rx` significa que puede cambiar a ese directorio, y `w` significa que puede crear o borrar ficheros en él (según los permisos de los ficheros, por supuesto), o el directorio mismo.

Por ejemplo, para prevenir que otros usuarios husmeen en `/home/guido/text`:

```
$ chmod o-rwx /home/guido/text
```

5.3 Equivalencia de órdenes

```
DIR:        ls, find, du
CD:         cd, pwd
MD:         mkdir
RD:         rmdir
DELTREE:    rm -R
MOVE:       mv
```

EJEMPLOS

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>dir                            $ ls
C:\GUIDO>dir file.txt                   $ ls file.txt
C:\GUIDO>dir *.h *.c                    $ ls *.h *.c
C:\GUIDO>dir/p                          $ ls | more
C:\GUIDO>dir/a                          $ ls -l
C:\GUIDO>dir *.tmp /s                   $ find / -name "*.tmp"
C:\GUIDO>cd                             $ pwd
        n/a - ver nota                  $ cd
        idem                            $ cd ~
        idem                            $ cd ~/temp
C:\GUIDO>cd \otros                      $ cd /otros
```

```
C:\GUIDO>cd ..\temp\trash          $ cd ../temp/trash
C:\GUIDO>md newprogs               $ mkdir newprogs
C:\GUIDO>move prog ..              $ mv prog ..
C:\GUIDO>md \progs\turbo           $ mkdir /progs/turbo
C:\GUIDO>deltree temp\trash        $ rm -R temp/trash
C:\GUIDO>rd newprogs               $ rmdir newprogs
C:\GUIDO>rd \progs\turbo           $ rmdir /progs/turbo
```

Notas:

- Cuando se use `rmdir`, el directorio a borrar debe estar vacío. Para borrar un directorio y todos sus contenidos, use `rm -rf` (bajo su propia responsabilidad).
- El carácter `~` es un atajo para el nombre de su directorio personal. Las órdenes `cd` o `cd ~` le llevarán a su directorio personal desde dondequiera que esté; la orden `cd ~/tmp` le llevará a `/home/su_directorio_de_usuario/tmp`.
- `cd -` «deshace» el último `cd`.

6 Disquetes, discos duros y similares.

Hay dos maneras de adminsitrar dispositivos bajo Linux: a la manera de DOS y a la manera de UNIX. Elija.

6.1 Administración de dispositivos en «modo DOS»

La mayor parte de las distribuciones de Linux incluyen las Mtools, un conjunto de órdenes que son perfectamente equivalentes a sus correspondientes órdenes en DOS, pero que empiezan con una 'm': p.e., `mformat`, `mdir`, `mmd`, `mdel`, y así. Pueden incluso conservar los nombres de ficheros largos, pero no los permisos de los ficheros. Si usted configura las Mtools editando un fichero llamado `/etc/mtools.conf` (se proporciona un ejemplo en la distribución), podrá también acceder a la partición DOS/Win, el CD-ROM, y la unidad Zip. Sin embargo, para formatear un disco sin formato, la orden `mformat` no funcionará. Como root, tendrá que ejecutar esta orden con anterioridad: `fdformat /dev/fd0H1440`

Nota: no podrá acceder a los ficheros del disquete con una orden como, por ejemplo, `less a:fich.txt`. Esta es la desventaja de la manera DOS de montar discos.

6.2 Administración de dispositivos en «modo UNIX»

UNIX maneja los dispositivos de una manera distinta. No hay unidades separadas como A: o C:; un disco, sea un disquete o lo que sea, pasa a ser parte del sistema de ficheros local mediante una operación denominada «montaje» (mounting). Cuando se acaba de utilizar el disco, antes de extraerlo, debe «desmontarlo» (unmount).

Físicamente, dar formato a un disco es una cosa, hacer un sistema de ficheros en él es otra. La orden de DOS `FORMAT A:` hace ambas cosas, pero bajo Linux son acciones separadas. Para dar formato a un disco, mire más arriba. Para crear un sistema de ficheros:

```
# mkfs -t ext2 -c /dev/fd0H1440
```

Puede utilizar `dos`, `vfat` (recomendado) u otros formatos en lugar de `ext2`. Una vez que el disco esté preparado, móntelo con la orden

```
# mount -t ext2 /dev/fd0 /mnt
```

especificando el sistema de ficheros correcto si no utiliza `ext2`. Ahora usted puede acceder a los ficheros del disquete. Todo lo que solía hacer con A: o B: se hace ahora utilizando en su lugar `/mnt`. Ejemplos:

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>DIR A:                         $ ls /mnt
C:\GUIDO>COPY A:*.*                     $ cp /mnt/* .
C:\GUIDO>COPY *.ZIP A:                  $ cp *.zip /mnt
C:\GUIDO>EDIT A:FILE.TXT                $ jstar /mnt/file.txt
C:\GUIDO>A:                             $ cd /mnt
A:> _                                   /mnt/$ _
```

Cuando haya acabado, antes de sacar el disco *deberá* desmontarlo con la orden

```
# umount /mnt
```

Obviamente, debe hacer un `fdformat` y un `mkfs` sólo a los discos no formateados, que no han sido usados nunca. Si quiere utilizar la unidad B:, ponga `fd1H1440` y `fd1` en lugar de `fd0H1440` y `fd0` en los ejemplos anteriores.

No hace falta decir que la manera de proceder con los disquetes también funciona con otros dispositivos, como por ejemplo, otro disco duro o una unidad CD-ROM. Esto es para montar el CD-ROM:

```
# mount -t iso9660 /dev/cdrom /mnt
```

Esta era la manera «oficial» de montar discos, pero hay un truco. Como es algo incómodo tener que ser root para montar un disquete o un CD-ROM, puede darse permisos a cada usuario de esta manera:

- como root, hacer lo siguiente:

```
# mkdir /mnt/a: ; mkdir /mnt/floppy ; mkdir /mnt/cdrom
# chmod 777 /mnt/floppy* /mnt/cd*
# # asegúrese de que el dispositivo CD-ROM es el adecuado
# chmod 666 /dev/hdb ; chmod 666 /dev/fd*
```

- añadir en `/etc/fstab` las siguientes líneas:

```
/dev/cdrom        /mnt/cdrom    iso9660   ro,user,noauto          0       0
/dev/fd0          /mnt/floppy   vfat      user,noauto             0       0
```

Ahora, para montar un disquete DOS y un CD-ROM:

```
$ mount /mnt/floppy
$ mount /mnt/cdrom
```

Cualquier usuario puede acceder a `/mnt/floppy`, `/mnt/a:`, y `/mnt/cdrom`. Recuerde que si considera importante la seguridad, dejar que todo el mundo pueda montar discos de esta manera constituye un gran agujero.

Dos órdenes útiles son `df`, que da información acerca de los sistemas de ficheros montados, y `du nombredir` que informa del espacio de disco consumido por el directorio.

6.3 Copias de seguridad

Hay muchos paquetes que pueden servirle, pero lo más simple para hacer una copia de seguridad multivolumen (como root) es:

```
# tar -M -cvf /dev/fd0H1440 /directorio_a_guardar
```

Asegúrese de tener un disquete formateado en la unidad, y otros ya preparados. Para restaurar sus ficheros, inserte el primer disquete en la unidad y utilice:

```
# tar -M -xpvf /dev/fd0H1440
```

7 ¿Qué pasa con Windows?

El «equivalente» a Windows es el sistema gráfico X Window, o X11, para abreviar. Al contrario que Windows o el Mac, X11 no fue diseñado para facilitar el uso o para lograr una mejor apariencia, sino para proporcionar utilidades gráficas a las estaciones de trabajo UNIX. Estas son las principales diferencias:

- Mientras que la apariencia y comportamiento de Windows es similar en todo el mundo, en X11 esto no es así: es mucho más configurable. La apariencia general de X11 está encomendada a un programa clave llamado «gestor de ventanas» (window manager), del cual hay una amplia oferta entre la que escoger: `fvwm`, basico pero agradable y eficiente con el consumo de memoria, `fvwm2-95`, `Afterstep`, `WindowMaker`, `Enlightenment`, y muchos más. El gestor de ventanas es normalmente llamado por un fichero denominado `.xinitrc`;
- Su gestor de ventanas puede ser configurado de manera que las ventanas actúen como en Windows: si usted hace click en ella, viene al primer plano. Otra posibilidad es que venga al primer plano si el puntero del ratón está sobre ella («focus»). También, el lugar de emplazamiento de las ventanas en la pantalla puede ser automático o interactivo: si un marco extraño aparece en lugar de su programa, haga click con el botón izquierdo del ratón donde quiera que aparezca;
- La mayor parte de las acciones pueden configurarse editando uno o más ficheros de configuración. Lea la documentación de su gestor de ventanas; el archivo de configuración puede ser `.fvwmrc`, `.fvwm2rc95`, `.steprc`, etc. Un fichero ejemplo de configuración suele encontrarse en `/etc/X11/nombre-de-gestor-de-ventanas/system.nombre-gestor-ventanas`;
- Las aplicaciones X11 se escriben empleando unas bibliotecas especiales (*widget sets*); como hay muchas disponibles, las aplicaciones tienen aspectos distintos. Las más básicas son aquellas que usan los controles (widgets) Athena (con aspecto 2D; `xdvi`, `xman`, `xcalc`); otras usan Motif (`netscape`), otras aún usan Tcl/Tk, Xforms, Qt, Gtk, y otra más. Casi todas esas bibliotecas dan el mismo aspecto y comportamiento que Windows;
- La manera de comportarse, desafortunadamente, puede no ser coherente. Por ejemplo, si selecciona una línea de texto utilizando el ratón y pulsa <RETROCEDER>, esperará que la línea desaparezca, ¿verdad? Esto no sucede con aplicaciones basadas en Athena, pero sucede con aplicaciones basadas en otros controles;
- El cómo funcionen las barras de desplazamiento y el cambio de tamaño dependerá del gestor de ventanas y el juego de widgets. Pista: si se encuentra con que las barras de desplazamiento no se comportan como usted esperaba, utilice el botón central o los dos botones juntos para moverla;

- Las aplicaciones no tienen un icono por defecto, pero pueden tener muchos. La mayor parte de los gestores de ventanas proporcionan un menu al que se tiene acceso haciendo click en el escritorio («ventana raíz»); no hará falta decir que el menú puede ser configurado. Para cambiar la apariencia de la ventana raíz, utilice `xsetroot` o `xloadimage`;
- El portapapeles sólo puede contener texto, y se comporta de manera extraña. Cada vez que selecciona texto, se copia automáticamente al portapapeles: vaya a cualquier sitio y pulse en botón central del ratón para pegarlo. Hay una aplicación, `xclipboard`, que proporciona varios portapapeles;
- Arrastrar y soltar es una opción, y está disponible solamente si usted utiliza aplicaciones X11 que la permitan.

Para ahorrar memoria, uno debería utilizar aplicaciones que usaran las mismas bibliotecas, pero esto es difícil de hacer en la práctica.

Hay proyectos que pretenden hacer que X11 tenga una apariencia y comportamiento tan coherente como Windows. Gnome, *http://www.gnome.org* y KDE, *www.kde.org* son un ejemplo. Pruébelos: no echará de menos su escritorio Windows nunca más.

8 Personalización del sistema

8.1 Ficheros de inicialización del sistema

Dos ficheros importantes bajo DOS son el `AUTOEXEC.BAT` y el `CONFIG.SYS`, los cuales se utilizan al rearrancar el sistema para inicializarlo, dar valores a algunas variables de entorno como `PATH` y `FILES`, y posiblemente lanzar un programa o fichero de procesamiento por lotes. Bajo Linux hay varios ficheros de inicialización, algunos de los cuales no deberían ser modificados hasta que usted supiese con seguridad lo que está haciendo. De todos modos, estos son los más importantes:

```
FICHEROS            NOTAS
/etc/inittab        ¡no tocar por ahora!
/etc/rc.d/*         idem
```

Si todo lo que necesita es establecer el PATH y otras variables de entorno, o desea cambiar los mensajes del login o ejecutar automáticamente un programa tras iniciar una sesión, eche un vistazo a los siguientes ficheros:

```
FICHEROS                        NOTAS
/etc/issue                      establece el mensaje de antes del login
/etc/motd                       establece el mensaje de despues del login
/etc/profile                    establece $PATH y otras variables, etc.
/etc/bashrc                     define alias y funciones, etc. (ver mas abajo)
/home/su_home/.bashrc           define sus alias y sus funciones
/home/su_home/.bash_profile   o
/home/su_home/.profile          establece el entorno y ejecuta sus programas
```

Si el último fichero existe (fíjese en que es un fichero oculto), se leerá tras el inicio de sesión y se ejecutarán las órdenes en él almacenados.

Ejemplo; mire este `.bash_profile`:

```
# Soy un comentario
echo Entorno:
printenv | less   # equivalente de la orden SET bajo DOS
alias d='ls -l'   # es facil comprender lo que es un alias
alias up='cd ..'
echo "Recuerde que su path es "$PATH
echo "Hoy es `date`"  # usa la salida de la orden 'date'
echo "Que tenga un buen dia, "$LOGNAME
# Lo siguiente es una funcion del shell
ctgz() # Lista los contenidos de un fichero .tar.gz
{
  for file in $*
  do
    gzip -dc ${file} | tar tf -
  done
}
# fin de .profile
```

`$PATH` y `$LOGNAME`, lo adivinó, son variables de entorno. Hay muchas otras con las que jugar; para buscar ejemplos, LPM de aplicaciones como `less`.

Poner esta línea en su `/etc/profile` le proporcionará el equivalente de `PROMPT $P$G`;

```
export PS1="\w\\$ "
```

9 Ficheros de inicialización de programas

Bajo Linux, casi todo puede ser configurado de acuerdo con sus necesidades. La mayoría de los programas tienen uno o más ficheros de inicialización con los que puede trastear, a menudo llamados `.nombreprogramarc`, situados en su directorio `home`. Los primeros que querrá modificar son:

- `.inputrc`: usado por `bash` para definir significados de teclas;
- `.xinitrc`: usado por `startx` para inicializar el sistema X Window;
- `.fvwmrc`: usado por el gestor de ventanas `fvwm`.
- `.joerc`: usado por el editor `joe`;
- `.jedrc`: usado por el editor `jed`;
- `.pinerc`: usado por el lector de correo `pine`;
- `.Xdefault`: usado por muchos programas X.

Para el resto de programas con el que se encontrará tarde o temprano, LPM. Quizá podría ser interesante para usted el documento «Cómo configurar Linux» *http://sunsite.unc.edu/mdw/HOWTO/Config-HOWTO.html*.

10 Trabajo en red: conceptos

No solo está disponible bajo Linux un «Acceso telefónico a redes», sino que también es más estable y más rápido. El nombre clave es «PPP», el protocolo empleado para conectarse a Internet utilizando módems. Necesita un núcleo que de soporte a PPP y una herramienta que marque y realice la conexión.

Para recoger el correo de su proovedor de servicios Internet necesita una herramienta llamada «recolector de correo» (email fetcher) que utiliza el protocolo POP; cuando recoja el correo, parecerá como si hubiera sido enviado directamente a su sistema Linux. Entonces deberá utilizar un gestor de correo de usuario como `pine`, `mutt`, `elm` u otros muchos para administrarlo.

Mientras que bajo Windows, siempre que lanza una aplicación de Internet se ejecuta el programa de conexión, bajo Linux el proceso es al revés. Primero se conecta, y después se lanza la aplicación. Un programa llamado `diald` proporciona el acceso usual. Instalar y configurar una red solía ser una de las cosas más difíciles de hacer bajo Linux, pero ya no lo es: por favor, consulte el documento COMO configurar Linux.

Finalmente, acerca del «Entorno de red»: puede hacer que su máquina Linux aparezca como un Windows NT/9x en una red local de máquinas Windows. La palabra mágica es Samba: no el baile brasileño, sino una implementación del protocolo SMB para Linux. Visite *http://samba.anu.edu.au/samba*.

11 Un poco de programación

11.1 Rutinas del Intérprete de Comandos: Ficheros .BAT con esteroides

Si ha utilizado ficheros `.BAT` para crear atajos de largas líneas de orden (yo solía hacerlo), el objetivo puede ser obtenido insertando las líneas de alias convenientes (ver ejemplo de más arriba) en el `profile` o en el `.bash_profile`. Pero si sus ficheros `.BAT` son más complicados, le encantará el lenguaje de Script (guionado) que el shell pone a su disposición: es tan potente como el viejo Qbasic (o más). Tiene variables, estructuras como `while`, `for`, `case`, `if-then-else`, y montones de nuevas características: puede ser una buena alternativa a un lenguaje de programación «de verdad».

Para escribir una rutina (script) —el equivalente a un fichero `.BAT` bajo DOS— todo lo que tiene que hacer es escribir un fichero ASCII estándar que contenga las instrucciones, guardarlo, y entonces hacerlo ejecutable con la orden `chmod +x fichero`. Para ejecutarlo, teclee su nombre.

Aviso: el editor del sistema se llama `vi`, y es un hecho probado que la mayor parte de los nuevos usuarios lo encuentran muy difícil de usar. No voy a explicar cómo usarlo; por favor, consulte el libro de Matt Welsh o busque un cursillo en la red. Suficiente decir por ahora que:

- para insertar texto, pulse 'i' y después el texto;
- para borrar caracteres, teclee `ESC` y después 'x';
- para salir de `vi` sin guardar, pulse `ESC` y después `:q!`
- para guardar y salir, pulse `ESC` y luego `:wq`

`joe` es un buen editor para novatos: llámelo tecleando `jstar` y tendrá los mismos atajos de teclas que en el editor de DOSWin. `jed` en modo WordStar o IDE es incluso mejor. Por favor, consulte la sección 12.4 (Dónde encontrar aplicaciones) para ver dónde conseguir estos editores.

Escribir rutinas bajo bash es una materia tan extensa que requeriría un libro para abarcarla toda, y no voy a profundizar más allá en este tema. Sólo daré un ejemplo de script, del cual se pueden extraer las reglas básicas:

```
#!/bin/sh
# ejemplo.sh
# Soy un comentario
# no cambie la primera linea: debe estar ahi
echo "Este sistema es: `uname -a`" # usa la salida de la orden
```

```
echo "Me llamo $0" # variables intrinsecas
echo "Usted me dio $# parametros: "$*
echo "El primer parametro es: "$1
echo -n "¿Como se llama? " ; read su_nombre
echo fijese en la diferencia: "Hola, $su_nombre" # entrecomillando con "
echo fijese en la diferencia: 'Hola, $su_nombre' # entrecomillando con '
DIRS=0 ; FILES=0
for fichero in `ls .` ; do
  if [ -d ${fichero} ] ; then # si el fichero esta en el directorio
    DIRS=`expr $DIRS + 1`  # DIRS = DIRS + 1
  else if [ -f ${fichero} ] ; then
   FILES=`expr $FILES + 1`
  fi
  case ${fichero} in
    gif|*jpg) echo "${fichero}: fichero grafico" ;;
    *.txt|*.tex) echo "${fichero}: fichero de texto" ;;
    *.c|*.f|*.for) echo "${fichero}: fichero de codigo fuente" ;;
    *) echo "${fichero}: fichero generico" ;;
  esac
done
echo "Hay ${DIRS} directorios y ${FILES} ficheros"
ls | grep "ZxY--!!!WKW"
if [ $? != 0 ] ; then # Sacar el codigo del ultimo orden
  echo "ZxY--!!!WKW no encontrado"
fi
echo "ya es suficiente... Para mas informacion teclee 'man bash'."
```

11.2 Sus programas en C

Bajo Unix, el lenguaje del sistema es C, le guste o no. Pero también están a su disposición otros lenguajes como FORTRAN, Pascal, Lisp, Basic, Perl, awk...

Suponiendo que usted sepa C, hay un par de guías para aquellos que han utilizado Turbo C++ o uno de sus hermanos bajo DOS. El compilador de C se denomina gcc y carece de todas las florituras que normalmente acompañan a sus análogos bajo DOS: no tiene IDE, ni ayuda en línea, ni debugger integrado, etc... Es sólo un rústico compilador de línea de órdenes, muy potente y eficiente. Esto es lo que debe teclear para compilar el típico programa `hello.c`:

```
$ gcc hello.c
```

lo que creará un fichero ejecutable llamado `a.out`. Para cambiar el nombre del ejecutable a otro nombre:

```
$ gcc -o hola hello.c
```

Para enlazar una librería al programa, añada la opción `-lnombre_de_librería`. Por ejemplo, para enlazar la librería `math`:

```
$ gcc -o mathprog mathprog.c -lm
```

(`-lnombre_libreria` fuerza a gcc a enlazar la librería `/usr/lib/libnombre_librería.a`; por lo tanto `-lm` enlaza `/usr/lib/libm.a`).

Bien. Pero cuando su programa está compuesto por varios ficheros de código fuente, necesitará usar la utilidad `make`. Supongamos que ha escrito un evaluador de expresiones: su código fuente se denomina `parser.c` e tiene un `#include` de dos ficheros en su cabecera: `parser.h` y `xy.h`. Entonces, desea utilizar las rutinas de `parser.c` en otro programa, `calc.c`, que hace un `#include` de `parser.h`. ¡Vaya lío! ¿Cómo se puede compilar `calc.c`?

Debe escribir un fichero llamado `Makefile`, el cual muestra al compilador las dependencias entre los ficheros de código fuente y los ficheros de código objeto. En nuestro ejemplo:

```
# Este es el Makefile utilizado para compilar calc.c
# ¡Pulse la tecla <TAB> en las posiciones marcadas!

calc: calc.o parser.o
<TAB>gcc -o calc calc.o parser.o -lm
# calc depende de dos ficheros de codigo objeto: calc.o and parser.o

calc.o: calc.c parser.h
<TAB>gcc -c calc.c
# calc.o depende de dos ficheros de codigo fuente

parser.o:  parser.c parser.h xy.h
<TAB>gcc -c parser.c
# parser.o depende de tres ficheros de codigo fuente

# fin de Makefile.
```

Guarde esto como `Makefile` y teclee

```
$ make
```

para compilar su programa. Guárdelo como `calc.mak` de forma alternativa y teclee

```
$ make -f calc.mak
```

Y, por supuesto, LPM.

Puede pedir ayuda acerca de las funciones de C, que se encuentra en las páginas `man`, sección 3; por ejemplo:

```
$ man 3 printf
```

Hay muchas librerías disponibles por ahí; entre las primeras que deseará usar están las `ncurses` (efectos en modo texto) y `svgalib`, (gráficos para consola). Si se siente suficientemente valiente para atacar la programación bajo X (no es tan difícil), hay muchas bibliotecas que harán mucho más fácil escribir programas para X11. Eche un vistazo a *http://www.xnet.com/~blatura/linapp6.html*, teniendo en cuenta el hecho de que Gtk se está volviendo un estándar.

Muchos editores pueden actuar como IDE. `emacs` y `jed`, por ejemplo, proporcionan marcado de sintaxis, indentación automática, y cosas similares. De manera alternativa, consiga el paquete `rhide` de *ftp://sunsite.unc.edu:/pub/Linux/devel/debuggers/*. Es un clon del IDE de Borland, y cabe la posibilidad de que le guste.

12 El restante 1%

Mucho más que un 1%, en realidad...

12.1 Utilización de tar y gzip

Bajo Unix hay algunas aplicaciones ampliamente utilizadas para archivar y comprimir ficheros. `tar` se utiliza para archivar varios ficheros en uno —es como `PKZIP` o `Winzip`, pero no comprime, sólo archiva. Para archivar varios ficheros en uno (que llamaremos archivo de ahora en adelante):

```
$ tar -cvf <nombre_fichero_final.tar> <fichero1> [fichero2...]
```

Para extraer ficheros de un archivo:

```
$ tar -xpvf <nombre_fichero.tar> [fichero]
```

Para listar los ficheros contenidos de un archivo:

```
$ tar -tf <nombre_fichero.tar> | less
```

Puede comprimir los ficheros usando `compress`, el cual es obsoleto y no debería ser utilizado nunca más, o usando `gzip`:

```
$ compress <fichero>
$ gzip <fichero>
```

eso crea un fichero comprimido con la extensión `.Z` (compress) o `.gz` (gzip). Estos programas sólo pueden comprimir un fichero cada vez. Para descomprimir, use:

```
$ compress -d <fichero.Z>
$ gzip -d <fichero.gz>
```

LPM.

Hay también utilidades `unarj`, `zip` y `unzip` (compatibles con ARJ y PK??ZIP). Los ficheros con la extensión `.tar.gz` o `tgz` (archivados con `tar`, posteriormente comprimidos con `gzip`) son tan comunes en el mundo Unix como los ficheros `.ZIP` bajo DOS. Para listar los contenidos de un fichero `.tar.gz` utilice:

```
$ gzip -dc <fichero.tar.gz> | tar tf - | less
```

o también:

```
$ tar -cvzf <fichero.tar.gz>
```

12.2 Instalación de aplicaciones

Antes de nada: instalar paquetes es trabajo del root. La mayoría de las aplicaciones Linux se distribuyen como ficheros `.tar.gz`, que contienen, típicamente, un directorio bien denominado conteniendo ficheros y/o subdirectorios. Una buena norma es instalar esos paquetes en el directorio `/usr/local` con la orden

```
# tar -zxf <archive.tar.gz>
```

leyendo después el fichero `README` o `INSTALL`. En la mayoría de los casos, la aplicación se distribuye en código fuente, por lo que deberá compilarla; a menudo, será suficiente teclear `make` y después `make install`. Si el archivo contiene una rutina `configure` ejecútela primero. Obviamente, necesitará el compilador `gcc` o `g++`.

Algunos archivos deben ser desempaquetados desde el directorio raíz; éste es el caso de los archivos `.tgz` de Slackware. Otros archivos contienen los ficheros pero no un subdirectorio - cuidado con liar las cosas. Antes de instalar un archivo, liste su contenido.

Debian y Red Hat tienen su propio formato de archivo; respectivamente, `.deb` y `.rpm`. El último está ganando una amplia aceptación; para instalar un paquete `rpm`, teclee

```
# rpm -i package.rpm
```

12.3 Trucos imprescindibles

Visión de pantallas anteriores: presionar <MAYÚS + RE PÁG> (la tecla gris) le permite volver atrás unas cuantas páginas, dependiendo de la memoria de vídeo que posea.

Resetear la pantalla: si por un casual hace un `cat` o un `more` de un fichero binario, su pantalla puede acabar llena de basura. Para arreglar las cosas, teclee `reset` a ciegas o pulse esta secuencia de caracteres: `echo CTRL-V ESC c RETURN`.

Pegar texto: en consola, ver más abajo; en las X, haga click y arrastre para seleccionar el texto en una ventana xterm, después haga click en el botón central (o con los dos a la vez si tiene un ratón de dos botones) para pegar. También hay un `xclipboard` (portapapeles de X) (de momento, sólo para texto); no se confunda por su muy baja velocidad de respuesta.

Usar el ratón: instale `gpm`, el controlador de ratón para la consola. Haga click y arrastre para seleccionar texto, entonces haga un click con el botón derecho para pegar el texto seleccionado. Funciona a través de diferentes consolas virtuales.

Mensajes del núcleo: échele un vistazo a `/var/adm/messages` o `/var/log/messages` como root para ver lo que el núcleo le dice, incluyendo los mensajes de inicio. La orden `dmesg` también es útil.

12.4 ¿Dónde encontrar aplicaciones?

Si se está preguntando si puede reemplazar su vieja y fiable aplicación DOS/Win con una aplicación de Linux, le sugiero que navegue por los mayores directorios de software para Linux: *ftp://sunsite.unc.edu/pub/Linux*, *ftp://tsx-11.mit.edu/pub/linux*, y *ftp://ftp.funet.fi/pub/Linux*. Otros buenos lugares para empezar son la «Página de Utilidades y Aplicaciones para Linux» *http://www.xnet.com/~blatura/linapps.shtml*, y la página «oficial» de Linux *http://www.linux.org*.

12.5 Unas cuantas cosas que Usted no podía hacer

Linux puede hacer muchas cosas que eran difíciles o imposibles de hacer con DOS/Windows. Esta es una pequeña lista que puede abrir su apetito:

- `at` le permite ejecutar programas a una hora determinada;
- `awk` es un lenguaje simple pero potente para manipular ficheros de datos (y no sólo de datos). Por ejemplo, siendo `data.dat` su fichero de datos con varios campos,

  ```
  $ awk '$2 ~ "abc" {print $1, "\t", $4}' data.dat
  ```

imprime los campos 1 y 4 de cada línea de `data.dat` cuyo segundo campo contenga la cadena «abc».

- `cron` es útil para realizar tareas periódicamente, en una fecha y hora determinadas. Escriba `man 5 crontab`.
- `find` (vea también sección 5.3 (Directorios: equivalencia de órdenes)) es una de las órdenes más potentes y útiles. Se utiliza para encontrar ficheros que tengan unas determinadas características y para realizar acciones en ellos. Un uso general de `find` es:

  ```
  $ find <directorio> <expresión>
  ```

  donde <expresión> incluye los criterios de búsqueda y las acciones. Ejemplos:

  ```
  $ find . -type l -exec ls -l {} \;
  ```

  encuentra todos los ficheros que son enlaces simbólicos y muestra a quién apuntan.

  ```
  $ find / -name "*.old" -ok rm {} \;
  ```

  encuentra todos los ficheros que ajusten con el patrón y los borra, pidiendo su permiso primero.

  ```
  $ find . -perm +111
  ```

  encuentra todos los ficheros cuyos permisos ajusten con 111 (ejecutables).

  ```
  $ find . -user root
  ```

  encuentra todos los ficheros que pertenecen al root. Hay cientos de posibilidades. LPM.
- `grep` busca patrones de texto en ficheros. Por ejemplo,

  ```
  $ grep -l "geología" *.txt
  ```

  lista todos los ficheros `*.txt` que contengan la palabra «geología». La variante `zgrep` trabaja con ficheros comprimidos con gzip. LPM;
- **Las expresiones regulares** son un método complejo pero muy potente de efectuar operaciones de búsqueda en un texto. Por ejemplo, `^a[^a-m]X{4,}txt$` se ajusta a una línea que comienza con 'a', seguida de cualquier carácter excepto los del intervalo a-m, seguido de 4 o más 'X', y terminado en 'txt'. Puede utilizar expresiones regulares con editores avanzados, `less`, y muchas otras aplicaciones. `man grep` para una introducción.
- `script` <`fichero_script`> vuelca los contenidos de la pantalla en `fichero_script` hasta que teclee la orden `exit`. Es útil para depuración;
- `sudo` permite a usuarios efectuar algunas tareas propias del root, como por ejemplo, formatear y montar discos; LPM);
- `uname -a` le da información acerca de su sistema;
- `zcat` y `zless` son útiles para visualizar y trabajar con ficheros comprimidos mediante gzip sin necesidad de descomprimirlos. Por ejemplo:

  ```
  $ zless textfile.gz
  $ zcat textfile.gz | lpr
  ```

- Los siguientes órdenes son útiles a menudo: `bc, cal, chsh, cmp, cut, fmt, head, hexdump, nl, passwd, printf, sort, split, strings, tac, tail, tee, touch, uniq, w, wall, wc, whereis, write, xargs, znew.` LPM.

12.6 Practicando UNIX bajo DOS/Windows

Tanto como si me cree como si no, hay buenas utilidades que proporcionan un entorno similar a UNIX bajo DOS/Windows. Uno es la suite Djgpp (*http://www.delorie.com/djgpp*) para DOS, mientras que Cygnus (*http://www.cygnus.com*) es un puerto más completo para Win32. Ambos incluyen las mismas herramientas y utilidades de desarrollo GNU que Linux; sin embargo, no conseguirá la misma estabilidad.

Si desea tener el sabor de Linux, pruebe Djgpp. Descargue e instale los siguientes ficheros (en el momento de escribir, la última versión era la 2.02): `djdev202.zip`, `bnu281b.zip`, `bsh1147b.zip`, `fil316b.zip`, `find41b.zip`, `grep22b.zip`, `gwk303b.zip`, `lss332b.zip`, `shl112b.zip`.. Se proporcionan instrucciones de instalación, y puede encontrar ayuda en *news:comp.os.msdos.djgpp*.

En particular, utilizar `bash` bajo DOSWin es una bocanada de aire fresco. Para configurarlo adecuadamente, edite el fichero `BOOT.BAT` que viene en la distribución para que refleje su instalación, y ponga estos ficheros en su directorio de usuario (de la partición Windows) en lugar de los que se proporcionan:

```
# este es _bashrc

LS_OPTIONS="-F -s --color=yes"
alias cp='cp -i'
alias d='ls -l'
alias l=less
alias ls="ls $LS_OPTIONS"
alias mv='mv -i'
alias rm='rm -i'
alias u='cd ..'

# este es _bprof
if [ -f ~/_bashrc ]; then
  . ~/_bashrc
fi
PS1='\w\$ '
PS2='> '
CDPATH="$CDPATH:~"
# cosas para less(1)
LESS="-M-Q"
LESSEDIT="%E ?lt+%lt. %f"
VISUAL="jed"                    # editor
LESSCHARSET=latin1              # visualizar letras acentuadas
export PS1 PS2 CDPATH LS_OPTIONS LESS LESSEDIT LESSOPEN VISUAL LESSCHARSET
```

12.7 Extensiones comunes y programas relacionados

Se podrá encontrar con una gran variedad de extensiones de ficheros. Excluyendo las más exóticas (como las de fuentes, etc.), aquí tenemos una lista de las extensiones más usuales:

- `1 ...` 8: páginas de manual. Léalas con `groff -Tlatin1 -man` <`file.1`>.
- `arj`: archivo hecho con `arj`.
- `dvi`: fichero de salida producido por TeX (ver más abajo). Use `xdvi` para visualizarlo; Use `dvips` para transformarlo en un fichero PostScript (`.ps`).
- `gz`: archivo comprimido con `gzip`.

- `info`: archivo *info* (Algo así como una alternativa a las páginas de manual). Consiga `info`.
- `lsm`: Fichero *Linux Software Map*. Es un fichero de texto ASCII plano que contiene la descripción de un paquete.
- `ps`: Fichero PostScript. Para visualizarlo o imprimirlo consiga `gs` y, opcionalmente, `ghostview`.
- `rpm`: Paquete de Red Hat. Puede instalarlo en cualquier sistema utilizando el gestor de paquetes `rpm`.
- `taz, tar.Z`: archivo hecho con `tar` y posteriormente comprimido con `compress`.
- `tgz, tar.gz`: archivo hecho con `tar` y posteriormente comprimido con `gzip`.
- `tex`: fichero de texto para utilizar con TeX, un poderoso formateador de textos. Consiga el paquete `tex`, disponible en muchas distribuciones.
- `texi`: fichero texinfo, a partir del cual se pueden producir tanto archivos TeX como info. Consiga `texinfo`.
- `xbm, xpm, xwd`: fichero gráfico.
- `Z`: archivo hecho con `compress`.
- `zip`: archivo hecho con `zip`. Consiga `zip` y `unzip`.

12.8 Conversión de ficheros

Si necesita intercambiar ficheros de texto entre DOS/Win y Linux, tenga en cuenta del problema de «fin de línea». Bajo DOS, cada línea de texto finaliza con CR/LF (esto es, ASCII 13 + ASCII 10). Bajo Linux, con LF. Si edita un fichero de texto DOS bajo Linux, cada línea acabará con un carácter 'M' de extraña apariencia. un fichero de texto Linux bajo DOS aparecerá como una sóla línea kilométrica sin párrafos. Hay un par de utilidades, `dos2unix` y `unix2dos`, que convierten estos ficheros.

Nota del revisor: Puede utilizar `vim` para convertir este tipo de ficheros:

- Para convertir un fichero DOS a formato Linux: (Las líneas con # delante son comentarios)

```
vi fichero_dos.txt
# pulse ESC
:set fileformat=unix
# pulse Enter
!wq fichero_linux.txt
# pulse Enter
```

- Para convertir un fichero Linux para su uso en DOS:

```
vi fichero_linux.txt
# pulse ESC
:set fileformat=dos
# pulse Enter
!wq fichero_dos.txt
# pulse Enter
```

Si sus ficheros de sólo-texto contienen caracteres acentuados, asegúrese de que están hechos bajo Windows (con NotePad) y no bajo DOS; si no, todas las letras acentuadas aparecerán como caracteres extraños.

Para convertir ficheros del Word o WordPerfect a texto plano, la cuestión es más dificil, pero aún posible. Necesita una de las herramientas que pueden encontrarse en los sitios CTAN; uno de ellos es *ftp://ftp.tex.ac.uk*. Consiga el programa `word2x` del directorio `/pub/tex/tools/`. En concreto, `word2x` convierte ficheros de Word 6, mientras que para ver ficheros realizados con Word 97 necesitará `mswordview` (*http://www.csn.ul.ie/~caolan/docs/MSWordView.html*) que los transforma en HTML.

12.9 Entornos ofimáticos gratuitos

Si convertir ficheros no le parece suficiente, tiene la posibilidad de probar un paquete (¡gratuito!) similar a Microsoft Office.

La suite StarOffice es gratuita para uso privado. Es grande, algo lenta, pero muy buena: ofrece muchas funcionalidades no encontradas en Microsoft Office. Puede también leer y escribir ficheros de Word y Excel, aunque la conversión no es siempre perfecta. Visite *http://www.stardivision.com*.

Otro buen paquete es Corel WordPerfect, del cual está disponible para descarga una edición gratuita. ¿Debería decir más? Vaya a por él: *http://www.corel.com*.

13 Fin, por ahora

¡Felicidades! Se ha iniciado un poco en el UNIX y está preparado para comenzar a trabajar. Recuerde que su conocimiento del sistema es aún limitado, y que se espera que practique más con Linux para usarlo cómodamente. Pero si todo lo que quería hacer era conseguir un puñado de aplicaciones y empezar a trabajar con ellas, apuesto a que lo que se incluyó aquí fue suficiente.

Estoy seguro de que habrá disfrutado de usar Linux y seguirá aprendiendo más acerca de él (todo el mundo lo hace). ¡Apuesto lo que sea, también, a que nunca volverá a DOSWin! Espero haberme hecho entender y haber realizado un buen servicio a mis 3 o 4 lectores.

13.1 Copyright

13.2 Agradecimientos

"CÓMO pasar de DOS a Linux"fue escrito por Guido Gonzato, *BORREMEguido@ibogeo.df.unibo.it* (Borre el «BORREME»). Muchas gracias a Matt Welsh, el autor de *Linux: Instalación y Primeros Pasos*, a Ian

Jackson, el autor del *Linux frequently asked questions with answers*, a Giuseppe Zanetti, el autor de *Linux*, a todos los amigos que me enviaron sugerencias, y especialmente a Linus Torvalds y GNU que nos trajeron Linux.

Este documento se distribuye «tal cual». He puesto un gran esfuerzo en escribirlo tan correctamente como he podido. Pese a ello, la información contenida en el mismo debe ser utilizada bajo su propia responsabilidad. En ningún caso el autor será responsable de cualquier daño resultante del uso de este documento.

Los comentarios son bienvenidos. Para cualquier duda, sugerencia, crítica, etc., siéntase libre de contactar conmigo.

Disfrute de Linux y de la vida,

Guido `=8-)`

13.3 Traducción

«CÓMO pasar de DOS/Win a LINUX» fue traducido por David Marín Carreño, *davefx@bigfoot.com*, como un pequeño grano de arena más dentro del Proyecto INSFLUG. Para más información, vea la sección 14 ().

He intentado una traducción fidedigna del documento original de Guido, aunque en algunos lugares haya actualizado información o cambiado algunas expresiones por otras más adecuadas a la jerga en nuestro idioma.

Si desea plantear alguna duda, sugerencia o crítica, pues tampoco dude en contactar conmigo.

```
  |   |   |  _|   David Marín Carreño (aka DaveFX)
   \  |  /  /
    |/____\|      Estudiante de Ingeniería en Informática.
   /_|_/\_|_\     Traductor de documentos Línux.
   \| (__) |/     Madrid. España.
   / \____/ \
  (____  ____)    ICQ#: 34866516
  _____||_____    E-mail: <davefx@bigfoot.com>
 /  _      _  \   WEb SiTe: http://www.bigfoot.com/~davefx
```

14 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: *www.insflug.org*. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Francisco José Montilla, *pacopepe@insflug.org*.