

# Fundamentos de los Sistemas Telemáticos

# Laboratorio 1

# Comandos básicos para el manejo de ficheros

Entrega obligatoria al final de la sesión presencial de laboratorio

# **Objetivos:**

- Familiarizarse con la línea de comandos (órdenes) de Unix.
- Conocer los comandos básicos para trabajar con ficheros.
- Saber utilizar un editor de texto.
- Aprender a consultar las páginas del manual.

### **Entorno:**

Ordenadores del laboratorio. Pero las actividades que se describen pueden realizarse en cualquier equipo con un sistema de tipo Unix.

# **Actividades previas:**

Antes de la sesión presencial de laboratorio que corresponda a su grupo deberá usted leer este documento y realizar las actividades guiadas que en él se describen (excepto el apartado 10 de "Entrega de resultados"), bien en un ordenador del mismo laboratorio (en horario libre) o en cualquier otro ordenador. De no hacerlo, es posible que no le dé tiempo a realizar la entrega. De este modo, la sesión de laboratorio tiene la función de una clase de tutoría para resolver las dudas que se le hayan podido presentar.

**Resultados:** Dos ficheros de texto que subirá al Moodle al final de la sesión de laboratorio. No se podrán subir dichos ficheros después de la hora de terminación del laboratorio de su grupo:

- a) Uno de los ficheros de texto contendrá todas las órdenes que ha utilizado en las actividades guiadas descritas en los siguientes apartados, como se indica al final de este documento (apartado 10). El fichero de texto debería llevarlo al laboratorio ya generado si ha realizado todas las actividades previamente.
- b) El profesor proporcionará al comienzo del laboratorio un enunciado con la práctica a realizar durante la sesión presencial. El resultado será el segundo fichero de texto que subirá al Moodle.

#### 1. Inicio

Como ya ha comprobado al realizar la actividad A0, los ordenadores del laboratorio están configurados para que al entrar en su cuenta le ofrezcan un entorno gráfico ("gnome") desde el cual se puede abrir en una ventana un terminal de texto. Todas las actividades de esta práctica se pueden hacer en ese terminal de texto, pero se recomienda que realice estas actividades desde un terminal de texto "puro". Para ello:

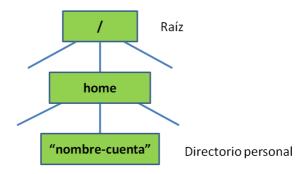
1. Pulse *simultáneamente* tres teclas: Ctrl, Alt y F1<sup>1</sup>. La pantalla se convertirá en una consola de texto, con letras blancas sobre fondo negro, pidiéndole de nuevo su nombre de

<sup>&</sup>lt;sup>1</sup> Puede ser F1, F2...F6: se pueden utilizar hasta seis consolas, y se puede cambiar de una a otra con Alt+Fi. En algunas distribuciones de Linux la consola "F1" no está disponible, porque queda bloqueada por los programas que lanzan el entorno gráfico: en lugar de una sola línea pidiendo el *login*, la pantalla está llena de mensajes de texto. En este caso, pulse Alt y F2 para cambiar a la siguiente consola.

login y contraseña. En cualquier momento puede volver al entorno gráfico pulsando simultáneamente Alt y F7, y retornar de éste a la consola con Ctrl+Alt+F1.

2. Una vez identificado le saldrá la invitación (prompt) de Unix: nombre-cuenta\$

El directorio de trabajo en donde entra es su directorio personal y ahí se almacenarán todos sus ficheros y directorios.



En lo sucesivo, el símbolo \$ significa que debe usted escribir la orden indicada.

Antes de empezar a practicar con las órdenes básicas, escriba:

### \$ history -c

Hay que dejar al menos un espacio de separación entre el nombre de la orden, history, y la opción, -c, y pulsar la tecla "Intro" (también conocida como "Ret" o "Retorno"). Observará que no hay ninguna respuesta (vuelve a salir la invitación), pero esta orden borra (c = clear) el historial de órdenes anteriores, de modo que al final de esta sesión de laboratorio se habrá generado un historial de las órdenes que usted ha tecleado.

Escriba su nombre y apellidos:

### \$ Nombre Apell1 Apell2

El intérprete de órdenes ("Shell") le dirá que no conoce ninguna que se llame como su nombre (salvo que usted tenga un nombre extraño que coincida con el de alguna orden). Pero la (falsa) orden ha quedado registrada. Escriba a continuación:

### \$ history

y fijese en el resultado.

Para que tenga clasificados todos los documentos que va a generar en cada práctica, debe crear un directorio para cada práctica. En el segundo laboratorio aprenderá a crear y gestionar directorios. Para la realización de esta primera práctica teclee:

```
$ mkdir lab1
$ cd lab1
```

# **Observaciones generales**

 A continuación, deberá ir escribiendo una serie de órdenes que quedarán registradas en el historial. Si se equivoca al teclear y el intérprete de órdenes le dice que no existe esa orden no se preocupe y escríbala correctamente. No es necesario que borre todo el historial anterior y repita lo que ya ha hecho.

• En algún momento puede ser que el intérprete se quede "bloqueado" y sin volver a dar la invitación (esto ocurre si, por ejemplo, ha olvidado completar la orden con algún parámetro). Para conseguir que vuelva a la invitación (y así anular dicha orden) pulse ^C (pulsación de la tecla C mientras se mantiene pulsada la "Ctrl").

### 2. Listar ficheros

Compruebe los resultados de estas órdenes (tenga cuidado de dejar al menos un espacio de separación entre la orden, **ls** y la opción que le sigue, que empieza por "—", y pulsar la tecla "Intro" tras terminar de escribir la última letra de la orden (todas las órdenes se terminan con esta tecla)):

Orden	Resultado
ls	Lista los nombres de los ficheros y directorios que hay en el directorio actual.
ls -l	Listado largo en el que, además de los nombres, aparecen datos asociados a los ficheros y directorios: permisos, número de enlaces, nombres del propietario y del grupo, tamaño en bytes y fecha y hora de la última modificación.
ls -a	Incluye ficheros y directorios ocultos: aquellos cuyo nombre empieza por un punto.
ls -l -a o bien: ls -la	Combina las dos opciones anteriores.
ls -R	Lista los nombres de los directorios y ficheros que hay en el directorio actual de forma recursiva

De momento, y si no ha creado aún ningún fichero en su cuenta, ls y ls -l no dan ningún resultado. (En los ordenadores del laboratorio aparece un directorio llamado "Escritorio" que contiene ficheros y/o enlaces relacionados con el entorno gráfico).

# 3. Crear y editar ficheros de texto

Vamos a utilizar un editor de texto sencillo que se llama nano:

# \$ nano prueba.txt

(En este caso, prueba.txt no es una opción, sino un parámetro, o argumento, para nano. Las opciones se distinguen porque empiezan con "-").

Si el fichero prueba.txt ya existía, se muestra su contenido. Supuesto que no, casi toda la ventana estará en blanco, y el cursor en la parte superior, esperando alguna entrada de teclado. Las dos últimas líneas resumen las acciones más comunes. El símbolo "^" representa a la tecla Ctrl. Por ejemplo, ^G significa "pulsar simultáneamente las teclas "Ctrl y G" (lo que lleva a mostrar una ayuda más extensa).

Escriba varias líneas con nombres de asignaturas en las que esté matriculado, por ejemplo. Guarde el fichero (^O, pide confirmación, que se le da con "retorno") y salga del editor (^X). Compruebe el número de bytes que tiene prueba.txt con la orden:

#### \$ 1s -1

Vuelva a abrir nano para crear y editar otro fichero de texto:

# \$ nano prueba2.txt

Inserte en él, con ^R (read), el fichero prueba.txt. Vamos a "engordar" prueba2.txt de modo que contenga al menos cien líneas. Compruebe que puede mover el cursor a cualquier parte con las teclas de flecha, y que puede insertar texto, así como modificarlo con las teclas "\(
el carácter anterior al cursor el cursor el carácter que está bajo el cursor el cursor el carácter que está bajo el cursor el cursor el carácter que está bajo el cursor el carácter el car

Como se trata de obtener un fichero con texto arbitrario, para no perder el tiempo escribiendo cien líneas, haga uso de otra facilidad del editor: Sitúese en cualquier punto y pulse ^K. Esto "corta" la línea actual, pero la conserva en memoria. Pulse luego ^U: esto "pega" esa línea delante de aquella en la que se encuentre el cursor. Por tanto, si mantiene dos o tres segundos pulsadas las teclas ^ y U, rápidamente se insertarán más de cien líneas.

Salve el fichero (pulse ^O y confirme que desea guardarlo con el nombre prueba2.txt), salga de nano (^X), y con ls -l compruebe el número de bytes de prueba2.txt.

Otra orden útil para conocer el tamaño de los ficheros de texto es **wc** (word count), que da el número de líneas, de palabras y de bytes. Compruébelo con estas dos órdenes:

```
$ wc prueba.txt
$ wc prueba2.txt
```

# 4. Ver el contenido de un fichero de texto

Abriendo el fichero con un editor, como nano, se puede ver su contenido, pero si sólo se trata de leerlo (no editarlo) se puede utilizarla orden cat (concatenate):

### \$ cat prueba.txt

le mostrará el contenido del fichero prueba.txt. Sin embargo, observe que con

# \$ cat prueba2.txt

únicamente se pueden visualizar las últimas líneas que caben en la pantalla. Para ver todo, desde el principio, se puede utilizar un paginador como **more**:

# \$ more prueba2.txt

muestra las primeras líneas y pulsando la barra espaciadora salta a las siguientes, y con la tecla b (back) a las anteriores. Termina cuando se avanza hasta el final o con la tecla q (quit). También se puede, con ciertas teclas, avanzar o retroceder línea a línea o un determinado número de líneas, pero las teclas de flechas no funcionan. Generalmente es preferible otro paginador, **less**, en el que sí lo hacen, y es más versátil:

# \$ less prueba2.txt

Para avanzar (y retroceder) línea a línea utilice las teclas de flecha abajo (y arriba); para avanzar una página, la barra espaciadora, y para retroceder, la tecla b. Para salir, la tecla q.

Salvo que se trate de un fichero pequeño, cat no tiene mucha utilidad para ver su contenido. Pero su función principal es otra: dándole como parámetros varios nombres de fichero, los concatena uno tras otro antes de enviarlos a la salida. En el Laboratorio 3 veremos cómo esa salida puede redirigirse a otro fichero en lugar de la pantalla.

# 5. Copiar, mover y borrar ficheros

La orden **cp** (**copy**) se utiliza con dos parámetros, un origen y un destino:

```
$ cp prueba2.txt prueba3.txt
$ cp prueba2.txt prueba4.txt
```

hacen una copia del fichero origen prueba2.txt en el fichero destino prueba3.txt, y del fichero prueba2.txt en el fichero destino prueba4.txt. El fichero origen debe existir. Si el fichero destino no existe, lo crea, y si existe, lo sobreescribe.

La orden diff (differences) también se utiliza con dos nombres de fichero, y sirve para mostrar las diferencias, línea a línea, entre los contenidos de los ficheros. Edite (con nano) prueba 4. txt, modificando dos o tres líneas cualesquiera, y analice los resultados de:

```
$ diff prueba2.txt prueba3.txt
$ diff prueba2.txt prueba4.txt
```

La orden **mv** (**move**), con dos nombres de ficheros como parámetros, renombra el primer fichero sin cambiar su contenido:

```
$ mv prueba.txt pp.txt
```

simplemente cambia el nombre de prueba. txt para que pase a llamarse pp. txt

Finalmente, **rm** (remove) borra uno o varios ficheros. Compruébelo con:

```
$ 1s
$ rm pp.txt prueba3.txt
$ 1s
```

Observe que el mismo resultado de la orden anterior para cambiar el nombre de prueba.txt (mv prueba.txt pp.txt) se podría haber conseguido con cp prueba.txt pp.txt seguida de rm prueba.txt.

Esta orden (rm) debe utilizarse con cuidado, porque los ficheros borrados no pueden recuperarse y, además, no pide confirmación: para que lo haga está la opción "-i". En algunas configuraciones está redefinida para que por defecto se utilice con esta opción. En otras se define una orden del, equivalente a rm -i.

### 6. Buscar en el contenido de ficheros de texto

Hay varias posibilidades para buscar una cadena de caracteres en un fichero de texto:

• Con un editor:

```
$ nano prueba2.txt
```

Vaya al final del fichero (con la flecha, o con ^V, que avanza página a página) e inserte una palabra nueva. Luego retroceda al principio (con la flecha, o con ^Y, que retrocede página a página). Para buscar la palabra pulse ^W. Salve el fichero (^O) y salga (^X).

Dentro de less:

```
$ less prueba2.txt
```

Escriba "/" seguido de la palabra que insertó antes y un retorno y observe el resultado. Salga (q).

• **grep** es un programa que tiene muchas opciones para buscar cadenas de caracteres en ficheros de texto. Por ejemplo, grep -n "cadena" "fichero" (donde "cadena" es una cadena de caracteres y "fichero" es un nombre de fichero) escribe las líneas del fichero que contienen esa cadena precedidas de sus números de línea (opción -n). Si la palabra que insertó antes es, por ejemplo, Pepe, compruébelo con

```
$ grep -n Pepe prueba2.txt
```

#### 7. Uso de comodines

Los caracteres "\*" y "?" tienen un significado especial para el intérprete de órdenes, y se llaman comodines (wildcards).

"\*" casa con cualquier secuencia de caracteres. Véalo con:

```
$ 1s p*
```

"?" se empareja con un solo carácter:

```
$ 1s p?
```

Si no hay ningún fichero que empiece por p y tenga exactamente dos caracteres no se obtiene nada. Haga algunos:

```
$ cp prueba2.txt pp
$ cp prueba2.txt pa
$ cp prueba2.txt pab
$ cp prueba2.txt pac
```

y observe el resultado de:

```
$ 1s p?
```

Finalmente, pruebe esto sin llegar a borrar los ficheros:

```
$ rm -i p?
$ ls p*
```

(la opción "-i" hace que pida confirmación antes de borrar) y luego esto otro:

```
$ rm -i p*
$ ls p*
```

# 8. Moverse por la historia de órdenes

En este momento usted ha tecleado unas 40 órdenes que han quedado guardadas en una zona de memoria reservada por el intérprete de órdenes. Puede listarlas con

```
$ history
```

Pero observe que sólo puede ver las últimas órdenes, las que caben en la pantalla. En los siguientes laboratorios aprenderá cómo verlas todas<sup>2</sup>.

Puede repetir la orden utilizando las teclas de flechas arriba y abajo ("↑" y "↓"): con pulsaciones sucesivas de "↑" se le irán mostrando la última orden realizada, la anterior, etc. (y con "↓"se moverá en sentido inverso). Puede confirmar la repetición de una orden con "retorno", y si es necesario, editar antes sus opciones o parámetros (con las teclas "←", "→" y "Supr").

# 9. Las páginas del manual

Casi todas las órdenes de Unix tienen un nombre corto y admiten muchas opciones. Una orden muy útil para obtener ayuda sobre cualquier orden es **man**. Así, para obtener toda la información sobre ls escriba:

### \$ man 1s

man utiliza normalmente less para paginar, lo que resulta útil para localizar informaciones en las páginas del manual, a veces muy extensas: escriba "/size" y "retorno" y verá que le lleva a la primera aparición de size<sup>3</sup>. Pulsando n irá a la siguiente, y con N a la anterior.

Igual que con less, la tecla q hace salir de man y devuelve a la invitación del intérprete de órdenes.

# 10. Entrega de resultados

Escriba lo siguiente:

# \$ history -w Grupo-Apell1-Apell2-L1-1.txt

Ello hará que la historia se grabe en el fichero **Grupo-Apell1-Apell2-L1-1.txt** (opción w=write). Compruébelo con:

### \$ less Grupo-Apell1-Apell2-L1-1.txt

(Recuerde que, por una limitación del Moodle, no se pueden utilizar caracteres con tilde en los nombres de los ficheros a subir.)

Naturalmente, Grupo, Apell1 y Apell2 deben ser los que le correspondan, por ejemplo:

```
G12-Perez-Gomez-L1-1.txt.
```

Suba este fichero a Moodle durante la sesión presencial de laboratorio. Hágalo desde el navegador.

Aquellos ficheros que se suban a Moodle y no respeten estrictamente el formato del nombre, se ignorarán y por tanto no se evaluará la entrega correspondiente.

Si tuviera abierta una sesión en el terminal de texto, ciérrela con:

#### \$ logout

(De lo contrario, y aunque luego cierre la sesión desde el entorno gráfico, el terminal de texto queda abierto y cualquiera podría acceder a su cuenta con Ctrl+Alt+F1, o Ctrl+Alt+F2, etc., como se indica en la nota de la primera página).

 $<sup>^2</sup>$  Si quiere hacerlo ya, escriba history | less. El símbolo "|" se obtiene pulsando AltGr y 1.

<sup>&</sup>lt;sup>3</sup> Esto, suponiendo que su sistema tiene las páginas de manual en inglés. Si las tiene en español, escriba tamaño.

# 11. Resumen

Orden	Uso
ls	Lista nombre de ficheros y directorios. Algunas opciones: - l: listado largo - a: incluye ficheros ocultos - R: listado recursivo
nano [ <fichero>]</fichero>	Editor de texto
wc <fichero></fichero>	Cuenta de líneas, de palabras y de bytes Algunas opciones: - w: Cuenta palabras; - l: Cuenta líneas - c: Cuenta bytes; - m: Cuenta caracteres
cat <f1> <f2></f2></f1>	Concatena ficheros y envía el resultado a la salida
more <fichero></fichero>	Paginador para ver el contenido de ficheros de texto
less <fichero></fichero>	Otro paginador con más facilidades que more
cp <f1> <f2></f2></f1>	Copia el fichero <f1> en el <f2>, creando éste si no existe Algunas opciones: - i: si <f2> existe, pregunta antes de sobreescribirlo - b: si <f2> existe, hace una copia (backup) previa con nombre <f2>~</f2></f2></f2></f2></f1>
diff <f1> <f2></f2></f1>	Lista las diferencias en los contenidos de <f1> y <f2></f2></f1>
mv <f1> <f2></f2></f1>	<f1> pasa a llamarse <f2></f2></f1>
rm <f1> <f2></f2></f1>	Borra los ficheros - i: pide confirmación antes de borrar
grep <cad> <f1></f1></cad>	Escribe las líneas de <fl> que contienen la cadena de caracteres <cad> Algunas opciones:  - i: Ignora la caja (mayúsculas = minúsculas)  - v: Escribe las líneas que no contienen la cadena  - n: Escribe también los números de las líneas p</cad></fl>
man <orden></orden>	Página de manual de la orden. Algunas opciones: - a: Si hay varias páginas para la orden las presenta secuencialmente - k: Seguida de una palabra (y sin nombre de orden) responde con todas las órdenes que mencionan esa palabra

# **Notas:**

- Los símbolos < y > en los nombres de parámetros no deben escribirse: <orden> significa <un nombre cualquiera de orden>. [<fichero>] significa que el nombre de fichero es opcional.
- Las opciones deben ir antes que los parámetros.
- No olvide separar con espacios el nombre de la orden, las opciones y los parámetros.

### ANEXO: Sobre la línea de comandos

El texto que sigue está tomado de <a href="http://en.flossmanuals.net/command-line">http://en.flossmanuals.net/command-line</a>, y es traducción de parte de un documento publicado en 2009 por la Free Software Foundation con el título "Command Line".

# Ventajas de utilizar comandos

Muchas personas que le dan una oportunidad a la línea de comandos quedan tan sorprendidas de sus posibilidades que no desean regresar a la interfaz gráfica de usuario. ¿Porqué? Bien, en resumen, las principales ventajas que ofrece la línea de comandos sobre los programas gráficos comunes son:

- Flexibilidad. Con los programas gráficos algunas veces se alcanza un límite, al encontrar que no puede hacer lo que desea, o tendrá que encontrar formas incómodas de trabajar considerando los límites del programa. No obstante, trabajando en la línea de comandos, usted puede combinar instrucciones para producir un intervalo virtualmente infinito de funciones nuevas e interesantes. Al combinar instrucciones de manera creativa, usted puede lograr que la línea de comandos realice exactamente lo que usted desea, lo pone al control de su computadora.
- Confiabilidad. Los programas gráficos a menudo son inmaduros o incluso inestables. En contraste, la mayoría de las herramientas que ofrece la línea de comandos son altamente confiables. Una de las razones de esto es su madurez: los programas de línea de comandos más antiguos han estado ahí desde finales de los años 1970, lo que significa que han sido evaluados durante tres décadas. También tienden a trabajar de la misma manera a través de diferentes sistemas operativos, a diferencia de la mayoría de las interfaces gráficas. Si usted desea una navaja suiza en la que pueda confiar, la línea de comandos es para usted.
- Rapidez. Las imágenes de las interfaces gráficas consumen gran cantidad de los recursos de hardware, a menudo resultando en lentitud o inestabilidad. La línea de comandos, por otra parte, utiliza los recursos de cómputo mucho más escasamente, liberando a la memoria y potencia de procesamiento para las tareas que usted desea llevar a cabo. La línea de comandos es también intrínsecamente más rápida: en lugar de dar clicks a través de largas cadenas de menús gráficos, usted puede escribir una docena o algo similar de golpes de teclado, y a menudo aplicarlos a múltiples archivos o otros objetos. Si usted es un mecanógrafo rápido, esto le habilitará a incrementar su productividad drásticamente.
- Experiencia. El utilizar la línea de comandos es una grandiosa experiencia de aprendizaje. Cuando usted utiliza la línea de comandos, se comunica con su computadora más directamente que con los programas gráficos, aprende así mucho sobre la forma como trabaja internamente la computadora: el utilizar la línea de comandos de manera natural es el camino para convertirse en un gurú de GNU/Linux.
- Diversión. ¿Alguna vez ha deseado ser como aquellos hackers de la computadora que pueden lograr que las máquinas GNU/Linux hagan cosas que usted ni siquiera ha soñado? La mayoría de ellos lo hacen utilizando la línea de comandos. Una vez que haya aprendido cómo utilizar esta poderosa herramienta, usted se verá a sí mismo haciendo todas esas cosas tan divertidas e interesantes que ni siquiera ha pensado que sean posibles.

### El valor de las instrucciones de comandos

Pero espere, ¡aún hay más!. Usted puede incluso almacenar instrucciones en archivos de texto. Estos archivos de texto son llamados scripts y pueden utilizarse en lugar de teclear una larga serie de instrucciones cada vez. Por ejemplo, si usted almacena instrucciones en un archivo llamado mycommand.sh, no tiene que volverá a escribir las instrucciones, sino simplemente teclear:

\$ mycommand.sh

Adicionalmente se pueden combinar instrucciones de manera simple o sofisticada. Más aún, usted puede programar scripts para que ocurran en un tiempo o fecha específicos o con un evento específico en su computadora.

También puede escribir scripts para que acepten información adicional del usuario. Por ejemplo, un script de redimensionamiento de imagen puede preguntarle al inicio el tamaño al que hay que redimensionar la imagen.

¿Alguna vez intentó hacer algo remotamente parecido a eso utilizando una interfaz gráfica de usuario?

Tal vez ahora pueda ver cómo el trabajo con la interfaz de línea de comandos comienza a abrir todo un mundo nuevo de utilidad a su computadora.

# ¿Está enferma mi computadora?

Otro uso de la línea de comandos consiste en revisar el buen estado de su computadora. Existen muchas instrucciones que usted puede usar para revisar cada aspecto de la salud de su computadora, desde la cantidad de espacio disponible en el disco duro hasta la temperatura de la unidad central de proceso (CPU). Si el desempeño de su computadora es pobre y usted no sabe qué es lo que está pasando, unas cuantas instrucciones lo ayudarán a determinar rápidamente si se trata de un asunto de Hardware o Software, y le ayudará a efectuar una reparación rápida.

### Atravesando la red

Existe otra característica importante de la línea de comandos que las interfaces gráficas no pueden igualar: la interacción sobre la red. Imagine que tiene una computadora en otra habitación y desea apagarla. ¿Cómo puede hacer eso? Fácil ¿No? Levántese, camine hacia la computadora y haga click sobre el botón apagar.

Bien, aquellos que tienen conocimiento de la materia pueden conectarse a la computadora de la habitación contigua por medio de la línea de comandos e introducir halt.

Esto puede parecer trivial. Tal vez sea mejor para usted levantarse de esa silla confortable y gastar 5 calorías caminando a la siguiente habitación. Sin embargo, ¿qué tal que la computadora que desea apagar está en otro suburbio? ¿En otra ciudad? ¿Otro país? El control remoto de esa computadora podrá ser entonces muy útil.

Apagar una computadora remota es sólo un comienzo. Cualquier cosa que usted pueda hacer en la línea de comandos la puede hacer en la computadora remota. Eso significa que puede correr scripts, ejecutar instrucciones, editar archivos de texto, revisar los diagnósticos, y llevar a cabo muchas otras tareas. El mundo de la línea de comandos ha crecido bastante.

### Incluso los programas gráficos son comandos

Cuando usted hace click sobre un icono o menú de opciones para abrir un programa, usted está ejecutando una instrucción. Encontrará ocasiones en las que es útil saber cuál es la instrucción o comando que se está ejecutando. Por ejemplo, si sospecha que un programa está ejecutándose de manera invisible en segundo plano y está disminuyendo la velocidad de su computadora, usted puede encontrar la instrucción de dicho programa y terminar el mismo. Los programas de interfaz de usuario gráfica conocidos como GUI, a menudo, envían más mensajes de error a la interfaz de línea de comandos (CLI), los cuales son mostrados en los cuadros de diálogo. Siendo esto de gran utilidad para diagnosticar problemas.