CONFIGURANDO EL ARRANQUE

profile

Resulta que ya tienes tu Linux instalado y quieres ponerlo un poco a tu gusto para sentirte un poco más cómodo. En cada Unix/Línux la línea de comando aparece de una manera diferente, aunque parecida. La manera más común sea quizá

usuario@maquina /path $

Esto se llama ***prompt***. A mí personalmente no me gusta el prompt anterior que a veces viene por defecto porque hace que la línea de comando sea demasiado larga y no me deja sitio para escribir mis comandos. A mí me gusta más algo del estilo de:

/directorio-actual $

Lo normal es que sepa en qué directorio estoy, pero si no lo sé, en lugar de mirar el prompt, tecleo ***pwd*** y me dice en qué directorio estoy. Para saber con qué usuario he entrado (a veces estarás con tu usuario y otras como root) lanzamos el comando ***"who am i"*** y nos dará la información del usuario con el que estamos registrados.

En Linux cada usuario tiene su directorio donde guardar sus ficheros particulares. El directorio de usuarios siempre estará en /home , de tal manera que si somos el user *alumno*, nuestra carpeta de usuario será /home/alumno . Sencillo ¿no? Para referirnos a nuestro directorio home, hay una forma muy corta y sencilla de invocarla. Es con ~ . De tal manera que estos 2 comandos son equivalentes y nos llevarían al mismo sitio:

cd /home/alumno *cd ~*

La variable donde se guarda el prompt en Linux es PS1. Para ver qué prompt tenemos podemos teclear lo siguiente:

$ echo $PS1

Para dejar un prompt como el que a mí me gusta (recordemos, último directorio únicamente), tendríamos que teclear lo siguiente:

$ PS1="\w $ "

El caso es que una vez que tenemos configurado nuestro prompt también vamos a querer tener configurado nuestro PATH. En MS-DOS, y en Windows, hay una variable de entorno llamada "PATH" que nos permite invocar ficheros ejecutables aunque no nos encontremos en el directorio donde reside el ejecutable. Esto es especialmente útil para no tener que estar tecleando la ruta del comando completa cuando lo lanzamos.

Por ejemplo, en vez de lanzar el comando de forma normal, tal que:

$ /directorio/subdirectorio/mi-comando

teclearíamos únicamente

$ mi-comando

y funcionaría igual. En caso de rutas largas se ahorra bastante tiempo. Para conseguir esto, como decía tenemos que configurar la variable PATH. Para ver lo que tiene esta variable tecleamos:

$ echo $PATH

Si queremos fijar nuestro PATH particular, tendremos que teclear algo como esto:

$ export PATH=$PATH/usr/bin:/etc:/mi-path-particular:

Como teclear todo esto cada vez que abrimos sesión es una locura además de un rollo, en Linux también tenemos unos ficheros para guardar nuestras preferencias que de alguna manera son el equivalente al autoexec.bat del viejo MS-DOS (¿os acordáis cuando hacíais vuestros ficheros .bat para comprimir con el arj sin tener que aprenderos todo el chorro "arj a –j –r –v1440 fichero.arj"? Pues esto es lo mismo). Este equivalente en Linux se llama .bashrc y .profile y podemos encontrarnos en nuestro directorio HOME.

~/.bashrc ~/.profile

La diferencia principal de .bashrc y .profile es que .profile se ejecuta cuando arrancamos la sesión y .bashrc cada vez que se invoca una consola de bash.

El arranque de Linux

Cuando arrancas tu máquina Linux puedes hacerlo de varias maneras. Puedes hacer que arranque en modo monousuario, en

modo multiusuario, en modo texto, en modo gráfico,… Lo habitual durante el curso es que arranquemos en modo gráfico multiusuario pero cuando estemos en un entorno de producción con un Linux configurado como servidor lo más seguro es que el modo gráfico no lo invoquemos para no desperdiciar recursos fundamentalmente. Además, desde modo de texto se puede arrancar el modo gráfico posteriormente (igual que antiguamente con MS-DOS, que se invocaba Windows con win.com ¿recordáis?).

Para controlar esto tenemos el fichero ***/etc/inittab*** En él vamos a especificar en qué modo vamos a arrancar.

```
# Default runlevel. The runlevels are:
# 0 (halt the system)
#  1 (single-user / minimal mode),
#  2 through 5 (multiuser modes), and
#  6 (reboot the system).
id:2:initdefault:
```

En Debian por defecto se arranca el Runlevel 2 (como podemos ver en la última línea del script anterior). Del Runlevel 2 al 5 en Debian no hay diferencia (en otros sistemas como Red Hat sí la hay). El significado de cada Runlevel lo establecemos en ***/etc/init.d*** En cualquier caso, lo que el ordenador va a hacer en el arranque y lo que no va a hacer se lo vamos a decir en los directorios de cada Runlevel. Hay 7 Runlevels en total, aunque para el caso en cuestión sólo vamos a alterar el 2. Por tanto nos vamos al directorio ***/etc/rc2.d*** y allí nos vamos a encontrar un montón de ficheros que empiezan por S o por K, un número y un nombre. En realidad lo que estamos viendo aquí no son más que servicios. Un fichero que empieza por S invoca el arranque de un servicio y uno que empieza por K, invoca su final (para cerrarlo ordenadamente antes de apagar la máquina). El número que viene a continuación es para establecer la prioridad del arranque, y el nombre, una identificación para sepamos lo que se va a arrancar. De tal manera que ***S01Apache*** establecería que el primer servicio que va a ejecutar la máquina una vez arranque será el servidor web Apache.

Los servicios podemos invocarlos a nuestro capricho desde la consola, como todo en Linux. Los accesos para llegar a los servicios los tenemos en el directorio /etc/init.d . Arrancar o parar un servicio es muy sencillo. Tan sólo tenemos que invocar el nombre del servicio y como argumento pondremos start/stop/restart según queramos arrancarlo, pararlo o reiniciarlo. Por ejemplo, queremos arrancar la base de datos mysql. Entonces teclearíamos

$ /etc/init.d/mysql start

Si todo va bien, nos saldrá un mensaje por pantalla diciendo que MySQL se ha iniciado y está listo para recibir peticiones. Si no queremos usarlo más, entonces:

$ /etc/init.d/mysql stop

Como ves, es tan sencillo, lógico e intuitivo como el resto de Linux.