

Introducción a Linux

Módulo 2



Comportamiento de bash

Comportamiento de bash

Bash usa un conjunto de funciones llamado **readline** al ingresar comandos.

Existen dos modos de operaciones de **readline**: **Vi** y **Emacs**. El nombre Emacs se debe al nombre de un editor que en algún momento fue muy popular en Linux. A continuación se verán algunas de sus características.

Algunos atajos de teclado que provee este modo:

Ctrl + t: intercambia la posición del carácter.

Alt + f: mueve el cursor a la siguiente palabra.

Alt + b: mueve el cursor a la anterior palabra.

Ctrl + u: borra la línea actual.

Completando Comandos

El shell bash incluye una característica llamada “completado de comandos”, que permite teclear sólo las primeras letras de un comando y pulsar la tecla tabulador para que el sistema lo complete. Si se desea ejecutar el comando **dmesg** para mostrar el buffer del kernel, se puede teclear:

```
$ dm
```

y pulsar la tecla **Tabulador**.

El sistema completará de esta manera:

```
$ dmesg
```

Si existe más de una coincidencia con la cadena tipeada, antes de pulsar la tecla Tabulador, el sistema hará sonar un *beep*. Al pulsar otra vez la tecla Tabulador se mostrarán todas las posibilidades que coincidan con lo tecleado. Pulsar la tecla **Esc** dos veces produce el mismo efecto que pulsar la tecla **Tabulador**.

Búsqueda incremental inversa

Bash tiene muchos atajos de teclado en modo Emacs. Uno de ellos es la búsqueda incremental inversa. Es muy útil para repetir comandos complejos y/o largos que fueron ejecutados con anterioridad. Por ejemplo, si ejecutamos con anterioridad el comando:

```
$ vi  
.local/share/kwin/scripts/Shimmer/contents/code/  
main.js
```

En lugar de escribirlo otra vez, se utiliza el atajo de teclado Ctrl + r y al tipear **scri** aparecerá algo así:

```
(reverse-i-search)`scri': vi  
.local/share/kwin/scripts/Shimmer/contents/code/  
main.js
```

Se llama *incremental* porque de haber más coincidencias se puede seguir buscando presionando nuevamente el atajo.

Alias y substitución de comandos

Alias de comandos

Aunque el sistema operativo y el shell nos ofrecen multitud de comandos y utilidades, podemos crear alias con nombres que tengan más sentido para nosotros o que sean más pequeños y así teclear menos caracteres.

Por ejemplo, si estamos familiarizados con la línea de comandos de Windows o del DOS estamos acostumbrados a escribir **“dir”** para obtener una lista de archivos de un directorio, para obtener lo mismo en Linux se escribiría:

```
“ls -l”
```

Podrías crear un alias de tal forma que cada vez que escribieras **dir** se ejecutase **ls -l**, utilizando la siguiente expresión:

```
$ alias dir="ls -l"
```

La sintaxis será siempre el alias seguido del comando que queremos que se ejecute cuando se teclee el alias, separados por el signo igual. En raras ocasiones podemos hacerlo sin encerrar el comando entre comillas, ejemplo **alias vi=vim**, pero como norma general deberíamos incluirlas.

Con el comando `alias` listamos todos los que están activados:

```
$ alias
alias cp='cp -i'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
$ unalias cp      # Sacar un alias
```


Substitución de comandos

Aquí se hace que los comandos se ejecuten de manera diferente, dándoles otro comportamiento.

```
$ touch ejemplo`echo -rino`.`date -I`  
$ ls -l ejemplo*  
-rw-rw-r-- 1 crond1 crond1 0 Oct 27 16:48  
ejemplo-rino.2011-02-03
```

En este ejemplo, se utilizan las comillas invertidas para que se ejecute el comando. Como se ve, bash interpreta que tiene que ejecutar un comando y lo que devuelve lo pone en el nombre del archivo que crea.

Otro ejemplo sería con **\$(comando)**

```
$ touch otro$(echo -rino).$(date)  
$ ls -l otr*  
-rw-rw-r-- 1 crond1 crond1 0 Oct 27 16:56  
otro-rino.Thu
```

Esas son las dos formas de substitución de comandos que se ampliarán en shell *scripting*.

**¡Sigamos
trabajando!**