

PERSONALIZANDO LA SHELL

Muchos somos los usuarios que nos tiramos delante de una consola demasiado tiempo. Dependiendo de como estemos aparecerán 2 modos de prompt, esa pequeña línea en la que inmediatamente después podemos escribir los comandos. Si estamos como usuario aparecerá algo así:

```
bash-2.04$
```

Si estamos como root la cosa cambia levemente (fijaos en el último carácter):

```
bash-2.04#
```

Estos prompts no es que sean bonitos pero son los que vienen por defecto. ¿Sabías que se pueden personalizar? Pues sí, les puedes poner colores, frases de bienvenida de todo tipo, que aparezca el nombre de tu máquina....

Prompt básicos.

Bajo Bash podemos cambiar el prompt cambiando el valor de la variable de entorno `PS1` de la siguiente manera:

```
export PS1="> "  
>
```

Los cambios toman efecto inmediatamente pero se pierden al reiniciar. Para que esto no suceda se puede especificar esa variable dentro del archivo `~/.bashrc`. `PS1` puede contener palabras:

```
export PS1="Este es mi super prompt > "  
Este es mi super prompt >
```

Esto es interesante, pero de momento es algo no del todo útil. Un caso algo más útil es que aparezca tu nombre de usuario y tu host (por si tienes problemas de identidad ;-)) sobre todo en gente que nos metemos en infinidad de usuarios y máquinas:

```
export PS1="\u@\H > "  
zx80@linux >
```

Fijaos que se han usado algunos caracteres especiales como `"\u"` para username y `"\H"` para la primera parte del hostname.

Que lío, ¿y como se yo los caracteres especiales que se deben poner para cada caso? Pues aquí los tienes:

Secuencia Descripción

<code>\a</code>	El famoso "beep" al entrar en consola.
<code>\d</code>	Fecha en modo "Sab Sep 06".
<code>\e</code>	El carácter ASCII escape.
<code>\h</code>	Primera parte del hostname.
<code>\H</code>	Todo el hostname ("pepito.griyo.com")
<code>\j</code>	Número de procesos suspendidos presionando CTRL+Z .
<code>\l</code>	El nombre en crudo del terminal ("tty4")
<code>\n</code>	Línea en blanco.
<code>\r</code>	Retorno de carro.
<code>\s</code>	El nombre del shell ejecutado (bash, cshell...)
<code>\t</code>	Tiempo en modo 24 horas ("23:01:01").
<code>\T</code>	Tiempo en modo 12 horas ("11:01:01").

```

\@      Tiempo en modo 12 horas am/pm.
\u      Nombre de usuario.
\v      Versión de bash.
\V      Versión de bash, incluido "patchlevel".
\w      Directorio actual.
\W      Directorio de trabajo.
\!      Posición en el historial del buffer del comando actual.
\#      Numero de comando.
\$$     Usuario.
\##     Root.
\xxx    Modo de tres número por el que se conocen los caracteres ASCII (007)
\\      Una contrabarra.
\[      Esta secuencia aparecerá antes de una secuencia de caracteres que no
        muevan el cursor.
\]      Esta secuencia aparecerá después de una secuencia de caracteres que
        no muevan el cursor.

```

Coloreando el shell.

Añadir un color al bash es muy fácil. Los colores se seleccionan añadiendo una secuencia especial a la conocida variable PS1.

Esto se hace emparejando un valor numérico entre un "\e[" y "m". Si especificamos más de un código numérico lo deberemos hacer separados por punto y coma (;)

Un ejemplo:

```
"\e[0m"
```

```
;; Vaya !! Un.. ¿bonito? prompt invisible....
```

Cuando se especifica cero como código numérico se interpreta como un reset del fondo y texto.

Para poner los colores que queremos los elegiremos de la siguiente tabla:

	40	41	42	43	44	45	46	47
30	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal
31	Bold	Bold	Bold	Bold	Bold	Bold	Bold	Bold
32	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal
33	Bold	Bold	Bold	Bold	Bold	Bold	Bold	Bold
34	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal
35	Bold	Bold	Bold	Bold	Bold	Bold	Bold	Bold
36	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal
37	Bold	Bold	Bold	Bold	Bold	Bold	Bold	Bold

Para usarla busca el color que quieres para el texto y su fondo, por ejemplo, un texto verde con fondo negro sería 32 y 40. Ahora mételo en el la línea de la siguiente manera:

```
export PS1="\w> "
```

seguido de:

```
export PS1="\e[32;40m\w> "
```

Muy bonito, mu chulo, pero no es perfecto aun. Para volver a dejarlo como antes ponemos la sintaxis:

```
export PS1="\e[32;40m\w> \e[0m"
```

Esto nos dejará un bonito prompt verde. La siguiente sintaxis es usada por los comandos o teclas que no hacen avanzar el cursor o que mostrarían signos raros:

```
export PS1="\[\e[32;1m\]\w> \[\e[0m\]"
```

No os preocupéis por usar muchos colores en un mismo prompt. Se puede hacer:

```
export PS1="\[\e[36;1m\]\u@\[\e[32;1m\]\H> \[\e[0m\]"
```

Mejoras en Xterm.

Los terminales tienen algunas opciones especiales por pertenecer al sistema gráfico X. Por ejemplo, podemos ponerle el nombre que queramos a la ventana de un xterm de la siguiente manera:

```
export PS1="\e]2;Jelouuuu\a"
```

Donde aparecerá la palabra Jelouuuu en el marco superior de la ventana del terminal.

No es necesario usar un texto estático, sino que se puede poner un grupo de caracteres en la barra de título. Por ejemplo, en esta sintaxis aparece el nombre de usuario, de hostname y el directorio activo en un texto verde:

```
export PS1="\[\e]2;\u@\H \w\a\e[32;1m\]>\[\e[0m\] "
```

Este prompt es muy útil por la cantidad de información que nos da en la barra de título. El problema de este truco es que si usamos un terminal no gráfico, como la consola del sistema, no aparecerá este título. Para solucionar esto se puede añadir estas líneas al .bashrc:

```
if [ "$TERM" = "linux" ]
then
  #Estamos en la consola del sistema o en un telnet.
  export PS1="\[\e[32;1m\]\u@\H > \[\e[0m\]"
else
  #No estamos en consola, sino en xterm
  export PS1="\[\e]2;\u@\H \w\a\e[32;1m\]>\[\e[0m\] "
fi
```

Este script configura el prompt dependiendo si nos encontramos en un terminal de sistema o en un xterm y procede.

NOTA: Aseguraos de que en el ~/.bash_profile o /etc/profile aparezca la siguiente línea: source ~/.bashrc

Si queremos abrir otro terminal éstas especificaciones también estarán activas. Un ejemplo algo completo para mostrar un terminal bien equipado:

```
export PS1="\n\[\033[35m\]\$ (/bin/date) \n\[\033[32m\]\w\n\[\033[1;31m\]\u@\h: \
[\033[1;34m\]\$ (/usr/bin/tty | /bin/sed -e 's:/dev/::'): \[\033[1;36m\]\$
(/bin/ls -l | /usr/bin/wc -l | /bin/sed 's: ::g') files \[\033[1;33m\]\$ (/bin/ls
-lah | /bin/grep -m 1 total | /bin/sed 's/total //' )b\[\033[0m\] -> \[\033[0m\]"
```