

Introducción a Linux

Módulo 2

Variables

Variables

Las variables son parámetros con nombre que contienen valores. Se pueden crear, modificar, referenciar y hacerlas parte del entorno de ejecución.

Por ejemplo, la personalización del prompt **bash** se puede realizar mediante la modificación de la variable **\$PS1**.

Ejemplo

```
$ PS1="[soy \u en \h dir actual \W]\$"  
[soy educacionit en centos dir actual etc]$
```

Secuencias de escape con ‘\’ y sus significados

- \u** Nombre de usuario.
- \h** Nombre de host.
- \W** Directorio de trabajo.
- \A** Hora actual.

Nota: el listado completo de opciones se puede tener en el manual de ayuda de **man** (**man bash**).

El sistema cuenta con algunas variables predefinidas que se pueden verificar con el comando **env**.

```
[soy educacionit en centos dir actual etc]$ env
TERM=xterm
SHELL=/bin/bash
OLDPWD=/etc
USER=educacionit
MAIL=/var/mail/educacionit
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
PWD=/etc
LANG=en_US.UTF-8
PS1=[soy \u en \H dir actual \W \A]$
SHLVL=1
HOME=/home/educacionit
LANGUAGE=en_US:en
LOGNAME=educacionit
```



La variable **USER** indica el usuario que está logueado en el shell.

La variable **HOME** muestra el directorio de inicio del usuario logueado.

La variable **PATH** le indica al shell actual dónde buscar los comandos que son ejecutados cuando no se le especifica una ruta. Si se tiene un comando en un directorio **/opt/prueba** que no está en la variable **PATH**, para ejecutarlo se debe escribir la ruta completa. Ejemplo:

```
/opt/prueba/comando
```

Para definir una nueva ruta en la variable **PATH** podemos hacerlo de la siguiente manera:

```
$ PATH=$PATH:/opt/prueba
```

Luego, con el comando **export** se puede exportar la variable como variable del entorno, para que sea utilizada por cualquier proceso de la shell actual y de sus subshells.

```
$ export PATH
```

Una variable muy importante es **EDITOR**, que es empleada por varias aplicaciones. Sencillamente se define con la ruta al comando:

```
$ export editor=/usr/bin/vim
```

Para que el valor de una variable sobreviva a la salida de sesión en una shell, es necesario exportar la variable pero además guardar esa configuración (como en el ejemplo anterior) en archivos como `~/ .bashrc` o `~/ .profile`.



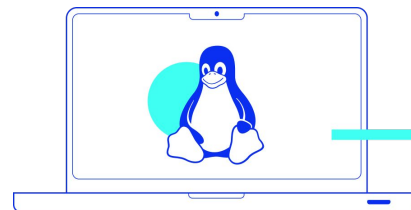
Otros comandos

Comando history

Este comando se utiliza para ver los **comandos ejecutados con anterioridad** en el *shell*. Veamos un ejemplo:

```
[educacionit@centos /etc]$ history
1 ls -ltr
2 cd /tmp/
3 ltr
4 cat onlycomm
5 vi onlycomm
```

Además, los archivos que fueron ejecutados antes se guardan en otro archivo que comienza por los antiguos hasta los más recientes.



La variable **HISTSIZE** define el número de comandos que recordará el *shell* durante la sesión actual. Esta variable puede estar definida tanto en **/etc/profile** como en **~/.profile** (~ equivale al directorio home del usuario) y su valor por defecto es de 1000 entradas. El comando `history` muestra todas las entradas del archivo del historial que se guarda en **~/.bash_history**.

La variable **HISTCMD** proporciona el índice dentro del historial comando que se está ejecutando. La variable **HISTFILE** especifica el nombre del fichero que contendrá el historial (**~/.bash_history** por defecto).

La variable **HISTFILESIZE** especifica el máximo número de líneas que contendrá el archivo especificado en **HISTFILE** y, aunque suele tener el mismo valor que **HISTSIZE**, podría ser diferente ya que esta última se refiere solo al histórico de la sesión actual y no al tamaño total del archivo histórico.



Operadores de expansión del historial de comandos

Operador	Descripción
!!	También conocido como bang-bang, este operador hace referencia al comando más reciente del historial.
!n	Hace referencia al comando número n del historial. Se puede utilizar el comando history para conocer estos números.
!-n	Hace referencia al comando actual menos n en el historial.
!cadena	Hace referencia al comando más reciente que comience por cadena.
!?cadena	Hace referencia al comando más reciente que contenga cadena.
^cadena1^cadena2	Sustitución rápida. Repite el último comando reemplazando la primera aparición de cadena1 por cadena2 .

El comando **fc**

La utilidad **fc** proporciona otra opción para editar los comandos del fichero histórico antes de ser ejecutados: abre el editor de textos por defecto con el comando especificado. De esta manera, es posible disponer de toda la potencia de un editor de textos, para editar y salvar un comando antes de ejecutarlo.

Se puede llamar a **fc** utilizando como parámetro el número del comando que se desea editar o, también, indicar un rango de comandos para editar y ejecutar en conjunto. También, se puede especificar el editor de textos que se va a utilizar.

Luego de llamar a **fc**, se puede editar el archivo de comandos como cualquier otro archivo de texto. Los comandos editados se ejecutarán al salir del editor.

La opción **-1** se utiliza para mostrar una lista de los valores especificados a continuación, se puede escribir, por ejemplo, **fc -1 50 60** para obtener la lista de los comandos del 50 al 60 en el historial.



**¡Sigamos
trabajando!**