

WIP = Work in Progress

Navegación eficiente de la CLI

La CLI (Interfaz de línea de comandos) es un entorno muy potente, flexible y programable. Si utiliza la interfaz de línea de comandos de forma regular, sabrá lo importante que es personalizar su entorno de trabajo y desarrollar accesos directos para garantizar un flujo de trabajo eficiente. A continuación hay algunos consejos sobre cómo navegar a través de directorios de una manera rápida y fácil.

Volver al Home

El comando `cd` cambia el directorio de trabajo actual a su `/` `home` :

```
$ pwd
/home/user/data/documents/work
$ cd
$ pwd
/home/user
```

Regresar al directorio anterior

Para volver al directorio anterior, puede usar `cd -` :

```
$ pwd
/home/user/.i3/config
$ cd ~/data/projects/dotfiles/i3
$ pwd
/home/user/data/projects/dotfiles/i3
$ cd -
/home/user/.i3/config
$ pwd
/home/user/.i3/config
```

Use el último argumento del comando anterior

La variable `$_` devuelve el último argumento del comando anterior. Esto puede ser útil en una variedad de escenarios:

```
$ pwd
/home/user/downloads/
$ cp i3status.tar.gz ~/data/builds/i3/i3status
$ cd $_
$ pwd
```

```
/home/user/data/builds/i3/i3status
```

```
$ chmod +x /path/to/my/script/script.sh  
$ $_  
(This will execute script.sh)
```

Lo mismo se puede lograr utilizando la combinación `Alt + .` o `Esc + .`.

Bash Completion

Bash offers `TAB` completion, a very useful feature that reduces the number of keystrokes you are required to type to navigate to a directory.

Autocompletado en Bash

Bash ofrece autocompletar al presionar `TAB`, una característica muy útil que reduce la cantidad de pulsaciones de teclas que debe escribir para navegar a un directorio.

```
$ cd d[TAB]
```

Expandirá la ruta con un directorio que comienza con `d`. Tenga en cuenta que Linux distingue entre mayúsculas y minúsculas, por lo que se ignorará `~/Desktop` o `~/Downloads`. Si hay más de 1 directorio que comienza con `d`, debe presionar la tecla `TAB` dos veces para obtener las finalizaciones posibles que se enumeran a continuación:

```
$ cd d[TAB][TAB]  
data/      downloads/
```

Ahora presione `o` y `TAB` para expandir la ruta a descargas.

Puede reducir en gran medida el número de pulsaciones de teclas al acceder a los directorios. Por ejemplo:

```
$ cd data/projects/python-dir/euler/completed/
```

Las pulsaciones de teclas que utilicé:

```
cd da[TAB]p[TAB]p[TAB]e[TAB]c[TAB]
```

Si cada directorio contuviera solo un subdirectorio, podría haber hecho lo siguiente:

```
cd [TAB][TAB][TAB][TAB][TAB]
```

El autocompletado de bash no solo reduce la cantidad de pulsaciones de teclas, sino que también resulta útil si no recuerda los nombres de directorio.

Tenga en cuenta que el autocompletado de bash también funciona en archivos:

```
$ cp data/projects/scripts/sync-script.sh .
```

Pulsaciones de teclas utilizadas:

```
$ cp d[TAB]p[TAB]s[TAB]s[TAB] .
```

El punto (.) representa el directorio actual, por lo que el comando copiará `sync-script.sh` al directorio actual.

Autocompletado de Bash Programmable

Para aprovechar todas las funciones de autocompletado de Bash, debe instalar un paquete adicional desde `/ extra` :

```
# slackpkg install bash-completion
```



CDPATH

Si trabaja en ciertos directorios de manera regular, es posible que desee incluirlos en la variable `$ CDPATH`. Suponga que a menudo trabaja en el directorio `slackbuilds` que contiene algunas compilaciones:

```
$ cd ~/data/projects/slackbuilds/  
$ ls  
i3 i3status yajl dmenu libev
```

Add it to the `$CDPATH` variable by modifying `~/ .bashrc`:

```
CDPATH=$CDPATH:~/data/projects/slackbuilds/
```

Tenga en cuenta que la ruta incluida en la variable `CDPATH` es la ruta al **directorio padre** donde se encuentran dichos directorios.

Después de obtener `.bashrc` (`source ~/ .bashrc`), puede `cd` a cualquiera de esos directorios desde cualquier lugar:

```
$ pwd  
/home/user/.config/xfce4  
$ cd yajl  
/home/user/data/projects/slackbuilds/yajl
```

Si desea habilitar el autocompletado de TAB dentro de los directorios agregados a través de la variable `CDPATH`, debe instalar `bash-complete` desde el directorio `/ extra` de Slackware.

NO se recomienda dar a sus directorios nombres similares a los del sistema, por ejemplo. `usr` , etc , que puede dar lugar a un comportamiento impredecible.

Enlaces simbólicos

En algunas situaciones, puede considerar usar [enlaces simbólicos](#) para crear *accesos directos* a directorios visitados regularmente:

```
$ ln -s /home/user/data/projects/scripts/slackbuilds ./slackbuilds
```

Alias Bash

Puede facilitarle la vida creando alias (= accesos directos) para los comandos que utiliza con frecuencia. La sintaxis es muy simple:

```
name_of_the_alias='value'
```

Puede colocar sus alias en `~/.bashrc` . Es posible que deba crear este archivo. Cada vez que edite este archivo, debe recargar `source` después para que los cambios surtan efecto:

```
source ~/.bashrc
```

o

```
. ~/.bashrc
```

Cuando se trata de navegación, uno podría, por ejemplo, crear algunos alias para acelerar la navegación en el árbol de directorios:

```
alias 1.='cd .. ; pwd'  
alias 2.='cd ../.. ; pwd'  
alias 3.='cd ../../.. ; pwd'  
alias 4.='cd ../../../../.. ; pwd'
```

El valor de un alias puede ser bastante complejo. Como puede ver, `4.` primero cambiará los directorios `cd ../../../../..` y luego imprimirá el directorio de trabajo actual - `pwd` . Tenga en cuenta un punto y coma (`;`) que separa los comandos.

```
$ cd data/projects/python-dir/euler/  
$ 4.  
/home/user  
$ cd -  
/home/user/data/projects/python-dir/euler  
$ 3.  
/home/user/data
```

Los alias se pueden usar de diferentes maneras. Algunos ejemplos más:

```
alias epyt='emacs -nw /home/user/data/projects/python-dir/euler/32-  
problem.py'  
alias slacktop='ssh user@slacktop'
```

Stack (pila) de directorios en bash

BASH presenta algunos complementos útiles de la pila de directorios que lo ayudan a navegar por los directorios visitados recientemente.

- `pushd`: guarda el directorio actual al principio del stack de directorios. Si le pasamos como parámetro un directorio guardará el directorio actual y el especificado (este último primero).
- `popd`: elimina el primer directorio del stack de directorios y hace un «`cd`» contra ese directorio.
- `dirs`: muestra el listado de directorios almacenados en el stack.

¿Cómo funciona en la práctica?

En primer lugar, agregue un directorio a la pila. Tenga en cuenta que también cambia automáticamente al directorio (el indicador `-n` suprime este comportamiento).

```
user@darkstar:~$ pushd data/projects/programming/  
~/data/projects/programming ~  
user@darkstar:~/data/projects/programming$
```

Alternativamente, puede `cd` a un directorio dado y emitir:

```
pushd .
```

Después de agregar algunos directorios, puede mostrar el contenido de la pila:

```
user@darkstar:~$ dirs -v  
0 ~/projects/web-develop/project-eden/includes  
1 ~/projects/web-develop/project-eden/includes  
2 ~/projects/web-develop/project-eden/pages/en  
3 ~/projects/web-develop/project-eden/css  
4 ~/projects/designs  
5 ~/projects/notes  
6 ~/public_html/project_eden
```

Tenga en cuenta que la primera entrada siempre muestra el directorio de trabajo actual, por lo que si también se encuentra en la parte superior de la pila, verá lo que parecen líneas duplicadas. El indicador `-v` es responsable de una salida bien indexada.

Para cambiar a uno de los directorios en la pila, puede emitir:

```
user@darkstar:~$ cd $(dirs +2 -l)  
user@darkstar:~/projects/web-develop/project-eden/pages/en$
```

Es cierto que esta no es la forma más concisa de cambiar directorios. Para acortarlo, podemos agregar un alias y una función al archivo `~ / .bashrc` .

```
alias dv='dirs -v'
```

Enumere la pila actual simplemente escribiendo `dv` .

```
cdd()  
{  
    position=$1  
    if [ -z $position ]; then  
        echo "You need to specify the position of the directory in the  
stack"  
    else  
        cd $(dirs +$1 -l)  
    fi  
}
```

La función `cdd` ¹⁾ hace posible `cd` a un directorio dado de la pila escribiendo:

```
cdd 3
```

Fuentes

- Fuente original: [Blog Post](#) escrito por [sycamorex](#)
- Traducido por [V́ctor](#) 2019/08/28 18:12 (UTC)

[howtos](#), [cli](#), [cd](#), [bashrc](#), [completion](#), [work in progress](#), [author sycamorex](#)

¹⁾

Basado en [esta función](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/es:howtos:software:efficient_cli_navigation

Last update: **2019/08/29 12:46 (UTC)**

