Introduction to Linux Shell

Dennis Ruiz (darvein)

10/15/2021

Trivial question

Why does **sudo id** works and **sudo cd** does not?

Why does **sudo id** works and **sudo cd** does not?

Because **cd** is a builtin command not a program (executable).

```
$ compgen -b
jobs
exit
bg
pwd
source
eval
alias
cd
export
echo
wait
kill
fg
...
```

What is a Linux shell?

It is a command-line interpreter software that can be interactive or scripted.

- Some popular shells: sh, bash, zsh, csh, ksh.
- Some popular terminals: xterm, iterm2, kitty, gnome-terminal

sh / bash:

- `sh` was created by Ken Thompson
- `sh` was created by Stephen Bourne
- Bash (Bourne Again Shell) is based on Bourne Shell created by Brian Fox

Where are the programs or executables?

Programs are located in $PATH

We can also see where exactly a program is located at:

```
~sh» whereis vim
vim: /usr/bin/vim /usr/share/vim
```

How to run an executable?

```
$ man id
$ id --help
```

Executables always comes with a help menu or a manual page.

```
$ $COMMAND help | --help | -h
```

Standard input, output & error

How data is being streamed:

- Kernel reads input from: /dev/stdin
- Kernel outputs a result: /dev/stdout
- Kernel outputs an error: /dev/stderr

Background and Foreground commands

Shell jobs can be managed in a way that background commands can be hidden from terminal and foreground occupies the terminal screen.

```
# List jobs
$ jobs -l

# Switch job to background or foreground
$ bg %n
$ fg %n

# Job stopped
$ CTRL-z or kill -tstp $PID
```

Completetition

A shell can autocomplete commands, options, files and directories by using .

Bash example of completion:

```
_foo()
{
    local cur=${COMP_WORDS[COMP_CWORD]}
    COMPREPLY=( $(compgen -W "bar baz" -- $cur) )
}
complete -F _foo foo
```

Wildcards and expansiosn

If we don't know the exact name of a file or directory we just use $*$:

```
$ ls a/**/*z*
a/b/c/z.txt
```

Brace expansions allow us to combine options:

```
$ echo {a,b,c}{d,e,f}
ad ae af bd be bf cd ce cf
```

```
~» echo {1..10}
1 2 3 4 5 6 7 8 9 10
```

History

Random command:

```
~sh» ls a/b
a.txt  b.txt  c  c.txt  foo
```

Simple access to a previous command:

```
~sh» ls !$/c
ls a/b/c
z.txt
```

Accessing to a specific command in history:

```
~sh» ls -ltra !10096:$
ls -ltra a/b/c
```

Splitting outputs with cut & awk

Cut example:

```
$ echo "abc def" | cut -f 2 -d ' '
def
```

AWK example:

```
$ echo "abc def" | awk '{print $2}'
def
```

Environment variables

Environment variables help programs to be dynamic allowing support for custom configurations or values:

```
~sh» TZ=Japan date
vie 15 oct 2021 14:27:24 JST
```

Conditionals and loops

Simple `if` conditional to create a file:

```
$ [ -f file.txt ] || touch file.txt
```

Simple loop conditional to iterate over a list of files & dirs:

```
$ for i in *; do echo $i; done
foo-completion-bash
intro-linux-shell.md
intro-linux-shell.pdf
```

More information

- Webpage: https://www.sudo.ws/sudo/
- Writing your own Bash Completion Function: https://fahdshariff.blogspot.com/2011/04/writing-your-own-bash-completion.html
- Bash manual: https://www.gnu.org/software/bash/manual/bash.html
- ZSH manual: https://zsh.sourceforge.io/Doc/Release/zsh_toc.html