

Introduction to the Bash shell on Linux and Mac OS

Using Bash More Effectively

Reindert-Jan Ekker

@rjekker

<http://www.linkedin.com/in/rjekker>



Overview

- Wildcards
 - Brace expansion
 - IO Redirection
 - Pipes
 - Command Substitution
-
- Keys
 - Searching History

Wildcards

- *****
 - Matches anything
 - Including nothing at all
 - *jpg, *str*

- **?**
 - Matches exactly 1 character

- **[acd7_]**
 - Matches one of the characters in the list
 - Above would match a, c, d, 7 or _
 - [^ax2] matches anything but a,x,2
 - ranges: [a-z], [0-9], [A-C3-5]

Brace Expansion

- **Generates strings**
 - Does not have to match existing filenames
- **Syntax: pre{list,of,strings}post**

Input	Expansion
<code>touch {a,b,c}.txt</code>	<code>touch a.txt b.txt c.txt</code>
<code>mv file.{txt,jpg} dir/</code>	<code>mv file.txt file.jpg dir</code>
<code>touch {a..c}{1..3}.txt</code>	<code>touch a1.txt a2.txt ... c2.txt c3.txt</code>

- **Brace expansion comes before wildcard expansion**

<code>mv *{txt,jpg} Documents</code>	<code>mv *txt *jpg Documents</code>
<code>mv filea?.{jpg,txt} a</code>	<code>mv filea?.jpg filea?.txt a</code>

Output Redirection

- **Redirecting standard output stream**
- **>**
 - Saves the output of a command to a file
 - `ls > listing.txt`
 - `cat > story.txt`
 - This will overwrite existing files!
- **>>**
 - Appends the output to the end of a file
 - `echo "buy milk" >> shopping.txt`

The Terminal and the Command Line

- **Bash: Text in, text out**
- **Terminal**
 - Handles keys pressed
 - Draws text
 - Fonts
 - Colors
 - Scrolling
 - Copy/Paste
- **On Mac**
 - “use option as meta” to use alt
 - Or install other terminal like iTerm 2
- **On linux**
 - Use middle-click paste

Command Substitution

- **Replace a command with its output**
 - Output will become a part of the command line
 - Put command between `$()`
 - `echo "hello, $(whoami)"`
 - `echo "Buy milk" > "notes$(date).txt"`
- **Note the use of double quotes**
 - Keep command substitution intact
- **Older form uses backticks**
 - `echo "you are currently on `hostname`"`

Movement keys

Key	Meaning	Alternative
Ctrl-a	Start of line	
Ctrl-e	End of line	
Ctrl-f	Forward 1 char	Right Arrow
Ctrl-b	Back 1 character	Left Arrow
Alt-f	Forward 1 word	Command-Left
Alt-b	Back 1 word	Command-right

Deletion keys

Key	Meaning	Alternative
Ctrl-D	Delete a char	Del
Ctrl-H	Delete a char backward	Backspace
Alt-D	Delete a word	
Ctrl-W	Delete a word backward	Alt-Backspace
Ctrl-K	Delete rest of line	
Ctrl-U	Delete from start of line	

Miscellaneous keys

Key	Meaning	Use for
Ctrl-C	Break	End a running program
Ctrl-D	End of transmission	Exit bash, End "cat > x"
up	previous line in history	Ctrl-p
down	next line in history	Ctrl-n
Ctrl-R	Search back in history	

Summary

- Wildcards
 - Brace expansion
 - IO Redirection
 - Pipes
 - Command Substitution
-
- Keys
 - Searching history