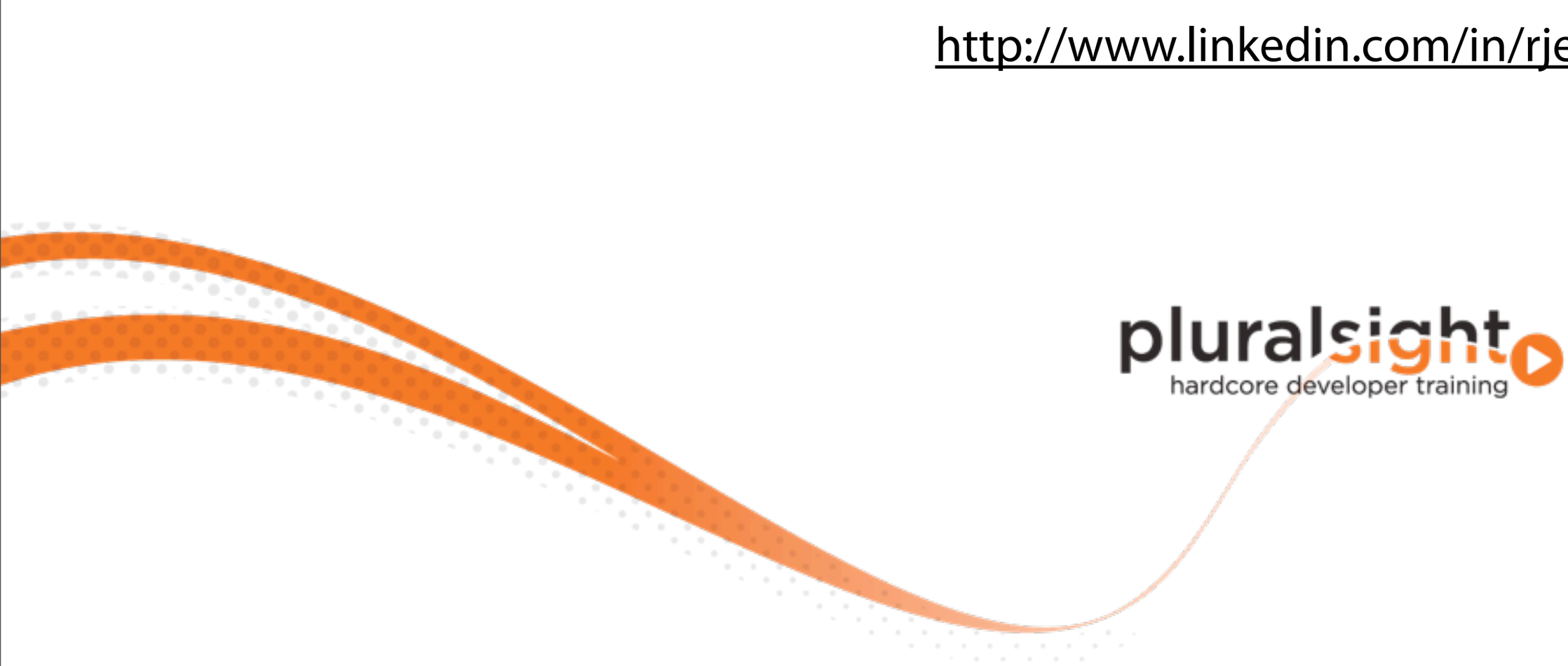
Introduction to the Bash shell on Linux and Mac OS

Basic File Management

Reindert-Jan Ekker

@rjekker

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



Overview

- **Displaying files**
- **Filenames**
- **Paths**
- **File and directory operations**
  - Create
  - Copy
  - Move
  - Delete

- **Cheat sheet available in extra course materials.**

Creating files and directories

- **Creating a directory: mkdir**

mkdir somefolder	Create directory “somefolder”
mkdir a b c d	Creates directories a, b, c, d

- **Creating an empty file: touch**

touch x	Creates empty file x

  - On existing file it updates modification and access dates

Listing text files

- **cat**
  - Best for small files - no paging
  - May mess up your terminal
  - Use “reset” to fix terminal
- **less**
  - Pager with lots of features
  - See “man less”
  - Use space to move down a page
  - Move back a page with “b”
  - Search with “/”
  - Exit with “q”

Opening other files on Mac OS

- **How to open non-text files?**

- **Mac OS: "open"**
  - Opens the program associated with this file
  - Like double-clicking in Finder
  - Opens in Aqua graphical user interface

open index.html	Opens html file in browser
open .	Opens current directory in Finder
open -a Preview picture.jpg	Opens picture.jpg with Preview

Opening other files on Linux

- **Linux does not offer a default "open" command**
  - Some window managers have a utility for this
  - Gnome: gnome-open
  - Kde: kde-open
  - Xfce: exo-open

- **Call the correct program directly**
  - "firefox index.html"
  - "opera index.html"
  - You have to know what is installed on your system

About filenames

- **Filenames can contain just about anything**
  - Except /
  - Hidden files start with a dot

- **Case sensitivity**
  - “hello” vs “Hello”
  - Linux filenames are case sensitive
  - Mac OS filenames are not
  - But these are defaults: it depends on filesystem

- **Extensions (.exe, .zip) are optional**
  - “file” command shows type

Filename dos and don’ts

- **Use letters, numbers, - and _**
  - If you want to be really safe, don’t use uppercase letters
- **Be careful with spaces**
  - Especially trailing spaces
- **Things to avoid:**

` ‘ ”	quotes	@	at	< >	less/greater than
~	tilde	&	ampersand	^	caret
*	asterisk	|	pipe		tab
#	pound	{}	brackets		space
!	exclamation mark	()	parens		delete
$	dollar	: ;	(semi)colon		backspace
?	question mark	\	backslash		

Quoting, Escaping, Completion

- **Backslash escapes a single character**
- **Single quotes**
  - Escape all characters between them
- **Double quotes**
  - Don’t use for now

Quoting examples

- **Example: folder "My Documents"**

cd My Documents	Error: My: No such file or directory
cd 'My Documents'	Ok
cd My\ Documents	Ok

- **Filename "$?*\"**

rm \$\?*\\	
rm '$?*\'	

- **Tip: use tab completion!**

Absolute paths

- A path represents a location on the filesystem

- Absolute paths
  - Start with /
  - Relative to the root

Examples	
/	
/bin/bash	
/var/log	
/home/reindert	

Relative paths

- **Relative paths**
  - Don’t start with a /
  - Resolved relative to current working directory

Path	Working directory	Location
Library	~	/Users/reindert/Library
alice	~/demos	/Users/reindert/demos/alice
demos/alice	~	/Users/reindert/demos/alice
../demos/alice	~/Library	/Users/reindert/demos/alice

Copying files

- **cp**
  - Warning: will silently overwrite existing files!
  - Arguments: a source filename and a target filename

Command	New file
cp a b	./b
cp a dir/b	dir/b

  - Other form: one or more source filenames and a target directory

cp a dir	dir/a	
cp a b c dir	dir/a, dir/b, dir/c	
cp dir/a .	./a	
cp * dir	dir/...	Copies all files in current directory

Copying directories

- **Use cp with the -R switch**
  - Copies everything in the directory recursively
- **cp -R source_dir target_dir**
- **cp -R dir1 dir2 dir3 target_dir**
- **cp -R dir1 file1 dir2 file2 target_dir**
- **On Mac OS: adding a slash copies only contents**
  - cp -R source/ target
- **Watch out when copying directory into itself**

Moving files

- **mv**
  - Warning: will silently overwrite existing files!
  - Arguments: a source filename and a target filename

Command	New file	
mv a b	./b	“rename“
mv a dir/b	dir/b	

  - Other form: one or more source filenames and a target directory
  - Use it to move directories as well

mv a b c dir	dir/a, dir/b, dir/c	
mv dir/a .	./a	
mv * dir	dir/...	Moves all files in current directory

Deleting files

- **rm**
  - Warning: will delete files permanently.
  - rm a
  - rm a b c
  - rm dir/a dir/b
  - rm *

- **rmdir**
  - Invocation like rm
  - Will remove empty directories only

- **rm -r**
  - Will recursively remove a directory and everything in it

Safety first: the -i switch

- **Prompt before overwriting or deleting files**
  - Use the -i switch
  - “cp -i” and “mv -i” will ask you before overwriting files
  - “rm -i” will ask before deletion

- **Combine it with other options**
  - cp -Ri
  - rm -ri

Summary

- **Displaying files:**
  - cat, less, open
  - “file” shows type
- **File and directory operations**
  - Create: touch, mkdir
  - Copy: cp
  - Move: mv
  - Delete: rm, rmdir
- **Filenames and quoting**
- **Absolute and relative paths**

- **Cheat sheet available in extra course materials.**