Writing simple tools: Spotter

http://github.com/borntyping/spotter

Sam Clements

May 3, 2013

About Spotter

- Spotter is a command line tool
- It takes a set of filename patterns paired with commands
- It waits for those files to change and then runs the command

Existing tools

inotify

```
while inotifywait -e close_write <path>; do
    <command>
done
```

```
python -m pyinotify <path> --command <command>
```

Existing tools

watchr uses a ruby file containing filename patterns and functions:

```
watchr watch.rb
```

```
watch(<path>) {system(<command>)}

watch("test/(.*)\.rb") do |md|
    system("ruby test/#{md[1]}.rb")
end
```

Why make Spotter?

All of the existing tools are either short bash one liners or use a full programming language.

With Spotter:

- It's very easy to understand and to read
- It doesn't require the use of a programming language
- The patterns and commands can be kept in the same place as a project, without being a full script

Spotter files

Watch a file for changes

```
watch: *.txt -> echo "File changed"
```

Spotter files

Watch a file for changes

```
watch: *.txt -> echo "File changed"
```

Use the filename in the command

```
watch: *.txt -> cat "{filename}"
```

Spotter files

Running commands at the start and finish

```
start: echo "Started watching files"

watch: *.txt -> echo "File changed"

stop: echo "Finished watching files"
```

Spotter files

Defining constants

```
define: <name> -> <value>
```

Spotter files

Defining constants

```
define: <name> -> <value>
```

Run a command when spotter starts and when a file changes

```
define: build -> python build.py

start: {build}

watch: *.html -> {build} --filename "{filename}"
```
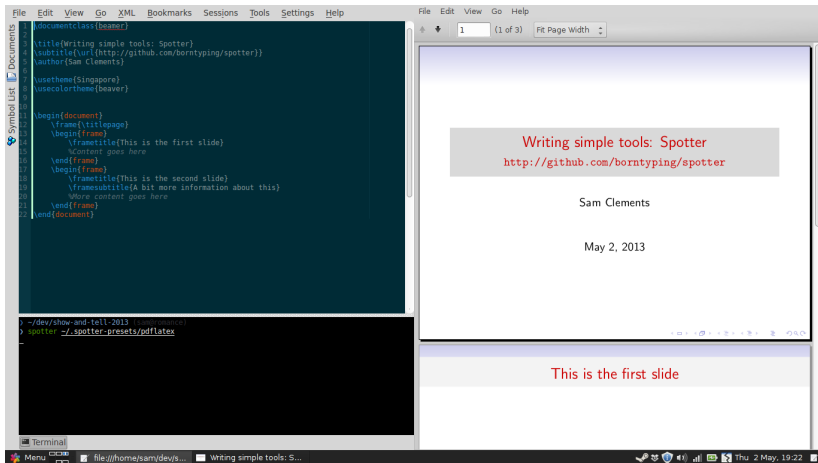
```
# Watch the LaTeX document
watch: *.tex -> pdflatex -draftmode "{filename}"
watch: *.tex -> pdflatex "{filename}"
watch: *.tex -> texcount -brief "{filename}"

# Clean unneeded files on exit
stop: rm -f *.aux
stop: rm -f *.log
stop: rm -f *.out
stop: rm -f *.toc
```

Using spotter to compile latex
Useful for non-dedicated-LaTeX editors

Using spotter to build a static website

```
# Watch the site for changes
define: build -> ./vdg.py --build-path ../build

# Build the entire site at start
start: {build}

# Rebuild the whole site when templates changes
watch: templates/* -> {build}

# Build individual files when they change
watch: * -> {build} {filename}
```

Writing simple tools: Spotter

Sam Clements

May 3, 2013