

Magic of Makefiles

Guy Nankivell

1 History of Makefiles

- Born in 1976 (43 years ago)

2 Requirements of a Build System

- Portable
- Efficient (Stop unnecessary recompilations)
- Language Agnostic (nested projects)
- Fast
- Modular
- Flexible

3 A Little Note...

What I henceforth refer to as ‘make’ is the GNU implementation. This is due to being the most common reference for `make` in the wild.

4 Cornerstone Features of Make

- Parallel

5 Okay, but can it work for us?

It is used as the build system for the Linux kernel so it must be doing something right...

6 Drawbacks

- Arcane Syntax

7 Compiling this presentation...

```
CC =          pdflatex
SRC =         talk.tex
OUT_DIR =     tmp
CFLAGS =      -output-directory $(OUT_DIR)
FILE_REDIR =  /dev/null
FNAME =       out.pdf

all: $(OUT_DIR)
@ $(CC) $(CFLAGS) $(SRC) > $(FILE_REDIR)
@ mv $(OUT_DIR)/*.pdf ./$(FNAME)

clean:
@ $(RM) -r $(OUT_DIR)/

$(OUT_DIR):
@ mkdir -p $(OUT_DIR)/
```

8 Plays nicely with vim

From command mode in vim all one has to do to action the Makefile in the same directory to which your session is running is;

`:make`

Which keeps your editing session alive and allows you to view the output of the command and then you need only press enter to drop back into editing mode

9 *Finis.*

Questions?