

Make and Makefiles

Ben Winjum

Office of Advanced Research Computing

February 9, 2023

Outline

- Set the stage for Make
- Write a simple Makefile
- Make progressive revisions to this Makefile to gain familiarity with the basic functionality and syntax
 - This will ideally serve you in a large percentage of your Makefile needs

To start

- We will be accessing materials I've made available at:
 - <https://github.com/benjum/oarc-make-intro.git>
- Grateful acknowledgement is given to Software Carpentry and their materials on “Automation and Make” [Copyright © Software Carpentry]
- I have substituted different texts, with grateful acknowledgement for those going to Project Gutenberg.

Make was created by Stuart Feldman in 1976

*Make originated with a visit from Steve Johnson (author of yacc, etc.), storming into my office, cursing the Fates that had caused him to waste a morning debugging a correct program (bug had been fixed, file hadn't been compiled, cc *.o was therefore unaffected). As I had spent a part of the previous evening coping with the same disaster on a project I was working on, the idea of a tool to solve it came up. It began with an elaborate idea of a dependency analyzer, boiled down to something much simpler, and turned into Make that weekend. Use of tools that were still wet was part of the culture. Makefiles were text files, not magically encoded binaries, because that was the Unix ethos: printable, debuggable, understandable stuff.*

— Stuart Feldman, The Art of Unix Programming, Eric S. Raymond 2003

Very basic Makefile

```
# comment
```

```
target : dependencies  
    action
```

Exercise

- 1) Write a new rule for `frank.dat` created from `books/frank.txt`.
- 2) Update the rule for `data` with this target
- 3) Write a new rule for `results.txt`. The rule needs to:
 - a) Depend on each of the three `.dat` files
 - b) Invoke the action:

```
python testzipf.py alice.dat carol.dat frank.dat > results.txt
```
- 4) Put this rule at the top of the Makefile so that it's the default target.
- 5) Update `clean` to remove `results.txt`.

Exercise

Rewrite each `dat` rule to use `$@` and `$<` (the first dependency of the current rule)

Exercise

Add `testzipf.py` as a dependency `results.txt`. What happens and why?

Exercise

- 1) Update the `% .dat` rule to reference `COUNT_SRC` and `COUNT_EXE`
- 2) Do the same for `testzipf.py` and the `results.txt` rule using the variables `ZIPF_SRC` and `ZIPF_EXE`.

End-of-workshop survey

Please fill out a [short survey](#)

to help us improve our workshop offerings!