make

Lecture 22
Class 24 of 28 | April 13th 2023 | COMP 211-002 | Joshua Bakita

Welcome!

Today:

- ➔ More on `make`

Logistics:

- ➔ Final exam exceptions: https://eef.oasis.unc.edu/
- ➔ For regrade rqs., prefer Gradescope or Pizza
- ➔ Research opportunity if you get an A/A-

Fun fact…

vim recognizes `make` *as a command, so you can just type* `:make` *from normal mode to rebuild your program (assuming you have a Makefile)*

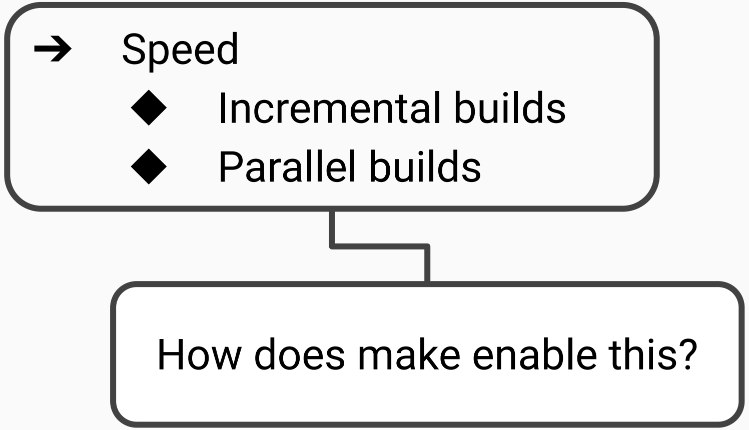
The `make` command

Picking up from last time…

The `make` command

Why another layer over `gcc`?

- ➔ Reliability
  - ◆ Never delete your source with a mistyped `gcc` command
  - ◆ Never forget to rebuild some part of your program
- ➔ Ease-of-use
  - ◆ Makes compilation instructions as simple as "Download and run `make`"



- ➔ Speed
  - ◆ Incremental builds
  - ◆ Parallel builds

How does make enable this?

Terminology Recap

Target
What you're building

Prerequisite
What you need to build it

Commands/Recipe
How to build it

`Makefile`
The file in which you specify all the above

The make command

Speed via a Dependency Gra

- ➔ A target can be a prerequisite for another target
- ➔ The combination of targets and prerequisites defines a tree, alternately known as a dependency graph

Legend

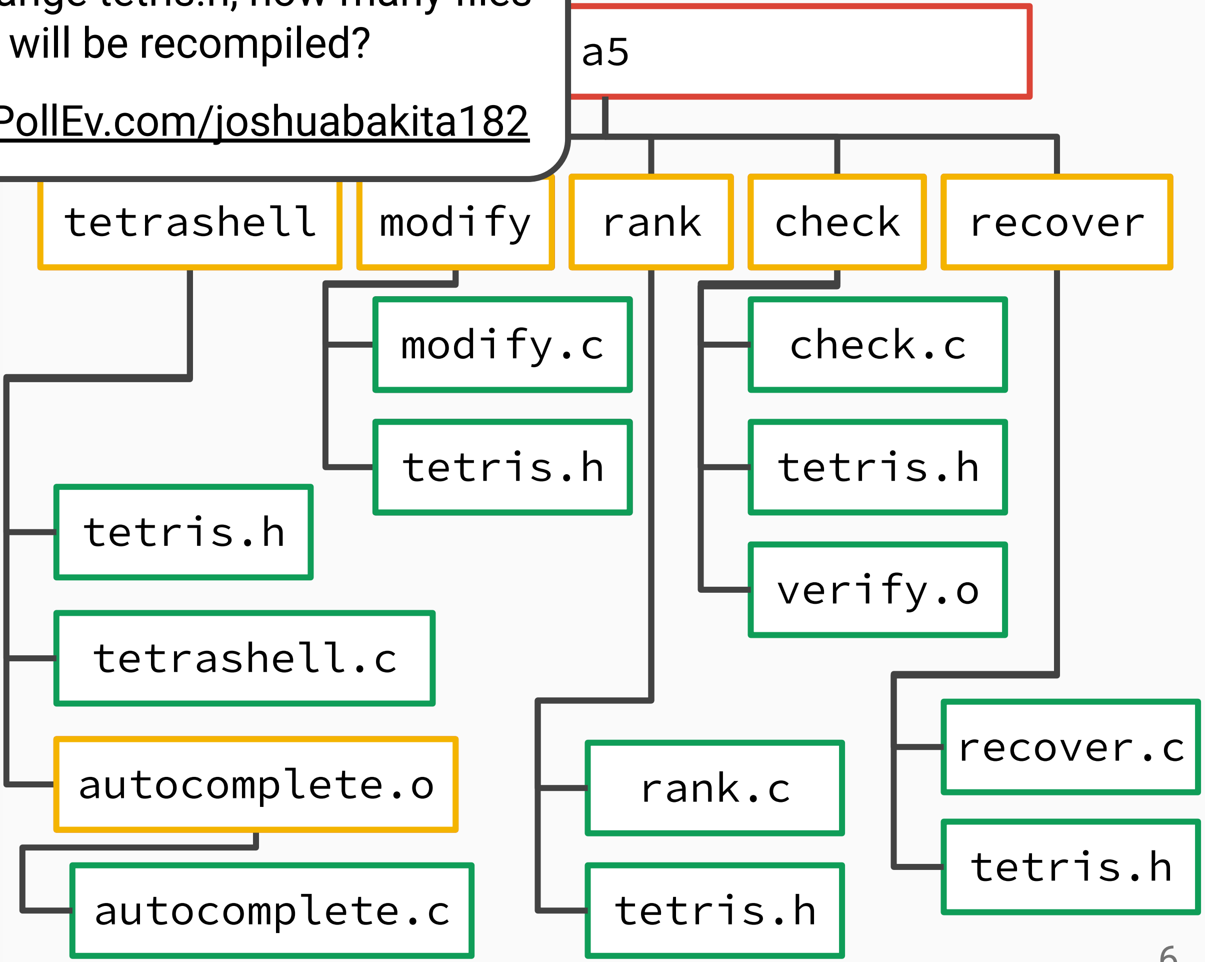
target

both

prerequisite

If we change tetris.h, how many files will be recompiled?

https://PollEv.com/joshuabakita182



The `make` command

Let's write a `Makefile`!

The `make` command

Advanced Features: Automatic Variables

Repeating filenames, like in this example:

```
rank: rank.c tetris.h
    gcc rank.c -o rank
```

can become annoying and error-prone.

See Sec. 2.2.1 "Automatic Variables" (pg. 16-17) in *Managing Projects with GNU Make* for more details.

Automatic variables are created as aliases for the target or prerequisite names. Two useful ones:

➔ `$@` for the filename of the target

➔ `$^` for the list of all prerequisites

Integrating this into the example:

```
rank: rank.c tetris.h
    gcc $^ -o $@
```

Questions?

Contact:

Email: hacker@unc.edu

Twitter: @JJBakita

Web: https://cs.unc.edu/~jbakita

Old Well, University of North Carolina at Chapel Hill, Winter 2017