Build Systems

Pat Pannuto / Marcus Darden

Q: What is a Build System?

Q: What is a Build System?

Q: Who has used a one?

Make, the Grandbinary of them all

- Automate compiling and linking
- Easily repeat long/complex commands
- Eliminate unnecessary work

Make syntax

(in 2 lines or less)

Make syntax

(in 2 lines or less)

```
target: dependencies
        rules
```

3 Makefiles

1. EECS 280 WN15
2. Build a sentence
3. EECS 281 (time permitting)

Let's Makefile!

- Build a sentence
  - Subject
  - Verb
  - Object
- "Source" files (.sh) create "Output" files (.txt)
  - Need a source file for each word
  - Need a source file for the whole sentence
- NEED A MAKEFILE

Other build systems

Other build systems

make-based

- GNU make
- nmake
- ...

Other build systems

make-based

- GNU make
- nmake
- ...

Non-make/Proprietary

- Bazel (Google's tool)
- Buck (Facebook's tool)
- MSBuild (Microsoft's tool)
- xcodebuild (Apple's tool)
- ...

Other build systems

make-based

- GNU make
- nmake
- ...

Non-make/Proprietary

- Bazel (Google's tool)
- Buck (Facebook's tool)
- MSBuild (Microsoft's tool)
- xcodebuild (Apple's tool)
- ...

Other languages for scripting and building

- Rake (ruby)
- Apache Ant (Java + XML)
- A-A-P (python)
- sbt (Scala)
- ...

Why stop at compilation?

What else could we automate?

Why stop at compilation?

What else could we automate?

- Build script generation
- Installation
- Testing
- Continuous Integration (CI)

Closing remarks

- **New section!!**
- Max points from Advanced Exercises in 3 or more sections
- Reminder: You must submit to staff at OH