Patrick O’Keefe with Matt Michaud

Google’s GO compiler

1. Identify which files/functions are responsible for each phase in the compiler source (scan/lex/tokenize, parse, AST, optimization, code generation).h

Scanner

Errors.go, example\_test.go, scanner.go, scanner\_test.go

Token

Position.go, position\_test.go, serialize.go, serialize\_test.go, token.go

Parser and Lex’

Error\_test.go, example\_test.go, interface.go, parser.go, parser\_test.go, performance\_test.go, short\_test.go

AST

Ast.go, ast\_test.go, commentmap.go, commentmap\_test.go, example\_test.go, filter.go, filter\_test.go, import.go, print.go, print\_test.go, resolve.go, scope.go, walk.go

Optimization and Code Generation

In the build files: <https://github.com/golang/go/tree/master/src/go/build>

1. What was the most ridiculous thing you found? (funny comments? awful code?)

Many uses of foo and bar I guess?

1. Take notes along the way (if you find something that’s unrelated to a compiler phase, try to infer what it’s doing).
2. Write up your findings in a short document and post to your repository (no more than two pages, please). For example:
3. git add findings.txt
4. git commit -m "Lab 1 findings."

git push origin master