IS893: Advanced Software Security

1. Introduction

Kihong Heo

KAIST

About Me

- Instructor: Kihong Heo (허기홍, kihong.heo@kaist.ac.kr)
- Research area: program analysis, SW security, programming language
- Homepage: https://kihongheo.kaist.ac.kr
- Office: N5 2321
- Office Hours: by appointment

Course Information (1)

- Course Website: https://github.com/prosyslab-classroom/is893-2020-fall
- TAs:
  - Changhoon Song (송창훈, songch@kaist.ac.kr)
  - ???
- Textbook:
  - Lecture slides will be provided
  - See the course website

Course Information (2)

- Announcement & QnA: the GitHub issue board
- Grading:
  - Homework: 40%
  - Project: 30%
  - Paper presentation & participation: 30%

A long time ago
in a galaxy far, far away….

SOFTWARE BUGS

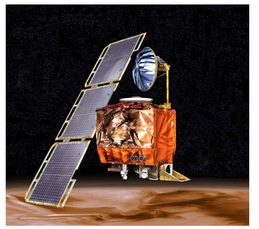
Ariane 5 Explosion

Software Bugs: A Persistent Problem

- A long time ago, far far away

The Patriot Missile (1991)
Floating-point roundoff
28 soldiers died

The Ariane-5 Rocket (1996)
Integer Overflow
$100M



NASA's Mars Climate Orbiter (1999)
Meters-Inches Miscalculation
$125M

- Unfortunately, it becomes your own problem now

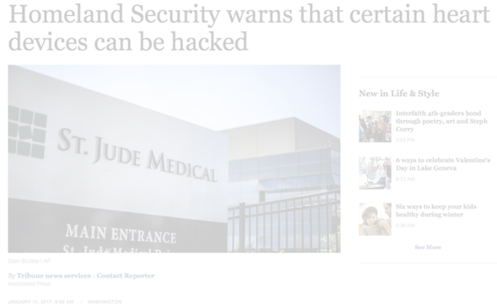
Software Bugs: A Persistent Problem

- A long time ago, far far away
- Unfortunately, it becomes your own problem now

COST OF A SOFTWARE BUG

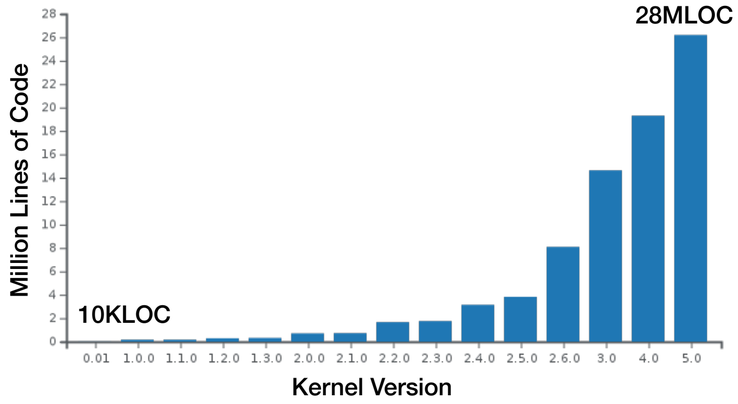
$100	$1,500	$10,000
If found in Gathering Requirements phase	If found in QA testing phase	If found in Production

- IBM Systems Sciences Institute, 2015

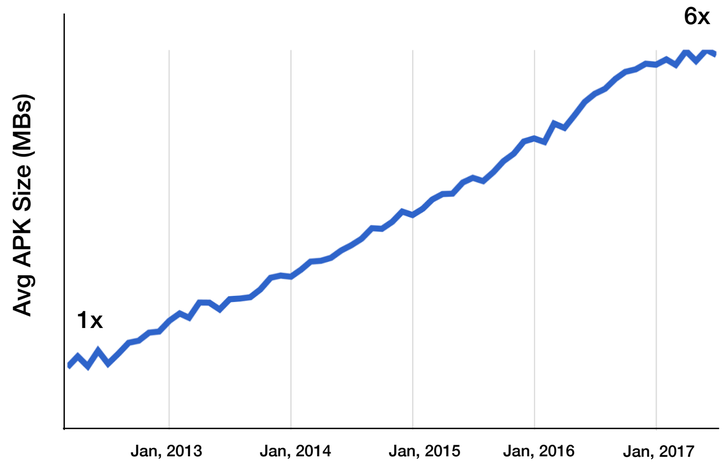


Why Software Still Fails?

Size of Linux Kernel



Avg. Size of Android Apps



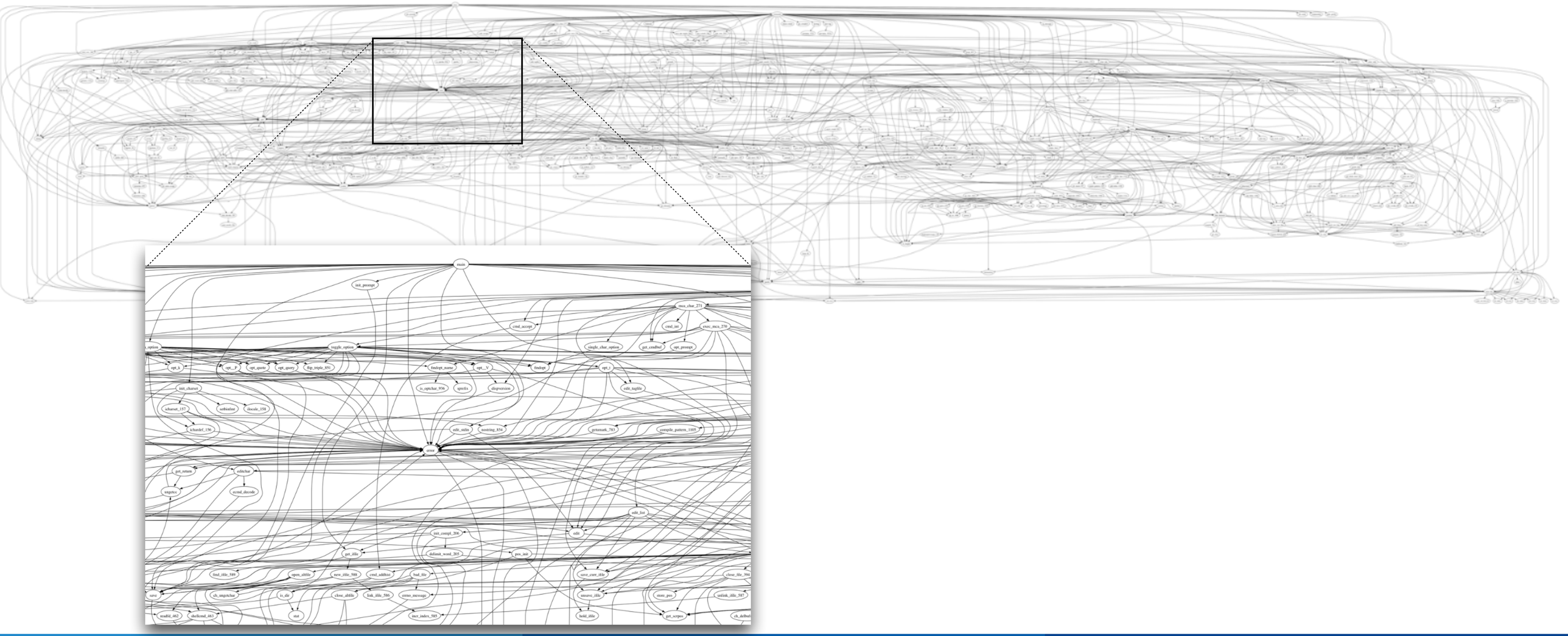
X

GitHub

10M+ New Developers
44M+ New Repositories
87M+ New Pull Requests
in 2019

Software Complexity

less-382 (23,822 LOC)



Course Objectives: Principles

- Why do SW bugs cause security problems?
  - E.g., How does a buffer overrun lead to privilege escalation?
- How to detect SW bugs?
  - Dynamic approaches (e.g., fuzzing)
  - Static approaches (e.g., type system, constraint solving)
- Three parts: basic concept, dynamic approaches, and static approaches

Course Objectives: Communication

- “Science is communication”
  - Reading: understand idea and concepts by reading research papers
  - Presenting: express what you learned
  - Listening and reacting: ask questions
- Each student will be
  - a presenter for 1 paper / part (i.e., basic concept, dynamic, static)
  - a commentator for 2 papers / part

Course Objectives: Practice & Challenge

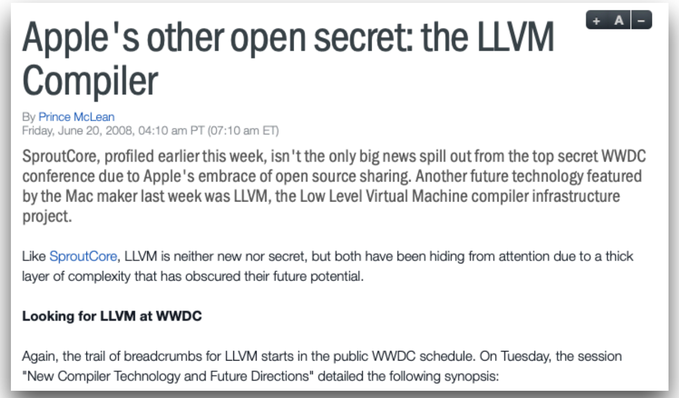
- Homework: **implement** conventional techniques
  - 6 - 8 (main) + 1 (prerequisite) programming assignments
- Project: **design and implement** your own tool
  - e.g., your own research domain, advanced techniques, new idea

Programming Assignments

- Programming assignments in OCaml using LLVM and Z3
  - You will write your tool in OCaml using Z3
  - Your analyzer will analyze LLVM IR code
- Why LLVM? (https://llvm.org)
- Why OCaml? (https://ocaml.org)
- Why Z3? (https://github.com/Z3Prover/z3)

The LLVM Compiler Infrastructure

- The de-facto standard & well-structured compiler toolchain
  - parser, code optimizer, linker, loader, debugger, etc
- A wide variety of frontends: C/C++, Obj-C, Swift, Fortran, etc
  - translated to the LLVM IR (intermediate representation)



Apple's other open secret: the LLVM Compiler

By Prince McLean
Friday, June 20, 2008, 04:10 am PT (07:10 am ET)

SproutCore, profiled earlier this week, isn't the only big news spill out from the top secret WWDC conference due to Apple's embrace of open source sharing. Another future technology featured by the Mac maker last week was LLVM, the Low Level Virtual Machine compiler infrastructure project.

Like SproutCore, LLVM is neither new nor secret, but both have been hiding from attention due to a thick layer of complexity that has obscured their future potential.

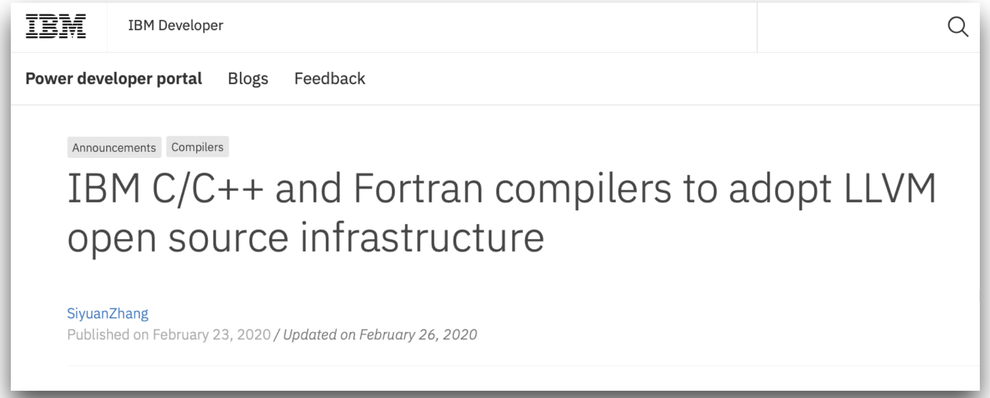
Looking for LLVM at WWDC

Again, the trail of breadcrumbs for LLVM starts in the public WWDC schedule. On Tuesday, the session "New Compiler Technology and Future Directions" detailed the following synopsis:

Google Chrome is replacing Microsoft's C++ compiler with Clang

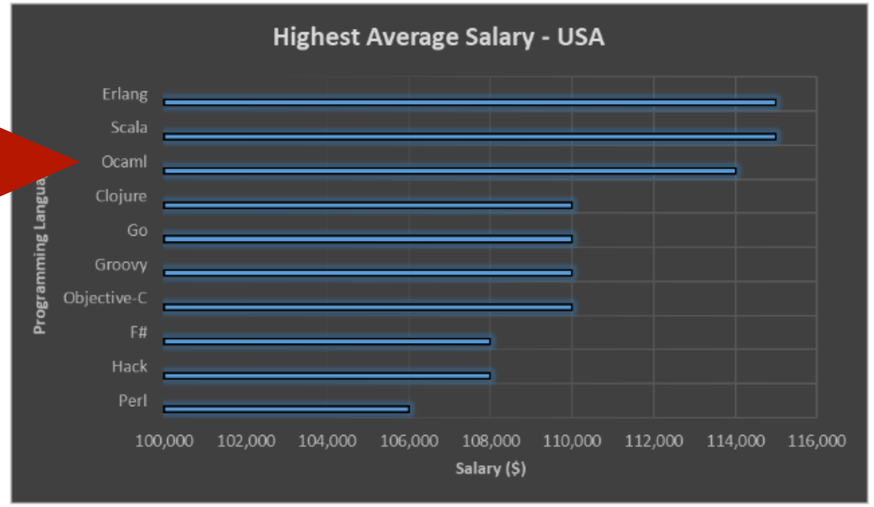
By Muhammad Jarir Kanji · Mar 6, 2018 14:06 EST · HOT!

Alongside bringing better touch support and automatic ad-blocking for 'intrusive' ads to the desktop version of Chrome, Google is also making some changes to its browser under the hood. The company is now starting to build Chrome for Windows using the Clang compiler which it already uses for other platforms like macOS and Linux.



The OCaml Language

- Simple, safe, realistic and high-level programming language
- Official OCaml bindings to LLVM API supported
- A lot of growing demands from academia and industry



StackOverflow, 2018

The Z3 Theorem Prover

- State-of-the-art automated theorem prover by Microsoft Research
- Solving satisfiability modulo theory (SMT) problems
  - first-order logic with background theories
    (e.g., arithmetic, bit-vectors, arrays, datatypes, uninterpreted functions, etc)

Boolean Satisfiability Problem (SAT)

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge (A \vee \neg C \vee B)$$

Satisfiable when
A = false
B = ture
C = true

Satisfiability modulo theory (SMT)

$$x + 2 = y \implies f(read(write(a, x, 3), y - 2)) = f(y - x + 1)$$

Arithmetic | Array | Uninterpreted Functions

Homework

- All submissions will be managed using Github / Github Classroom

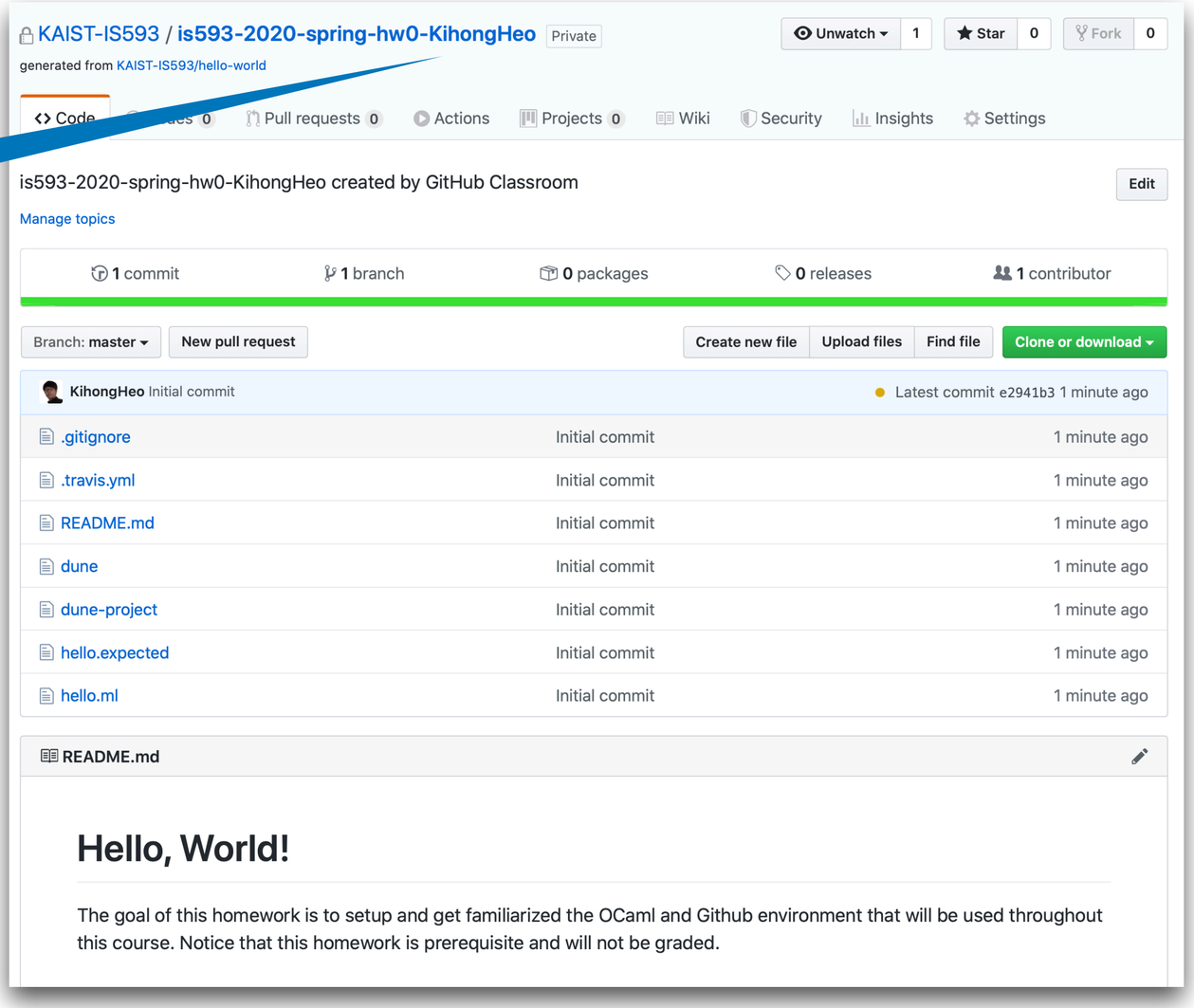
1. For each HW, a unique invitation URL will be posted at KLMS
2. Once you accept, a private repo for your assignment will be created
3. You can push as many commits as you want before the deadline
4. The final commit of your master branch will be graded
    - 80% credit for 1-day late, 50% credit for 2-days late, NO credit otherwise

Homework 0: Hello, World!

- Goal: setting up and getting familiarized with OCaml and Git environments
  - Implement your “hello-world” program in OCaml
  - Push to your Github repository
  - See the result in travis-ci.com (sign in with your Github account)
- The invitation URL is posted at KLMS
- No due date

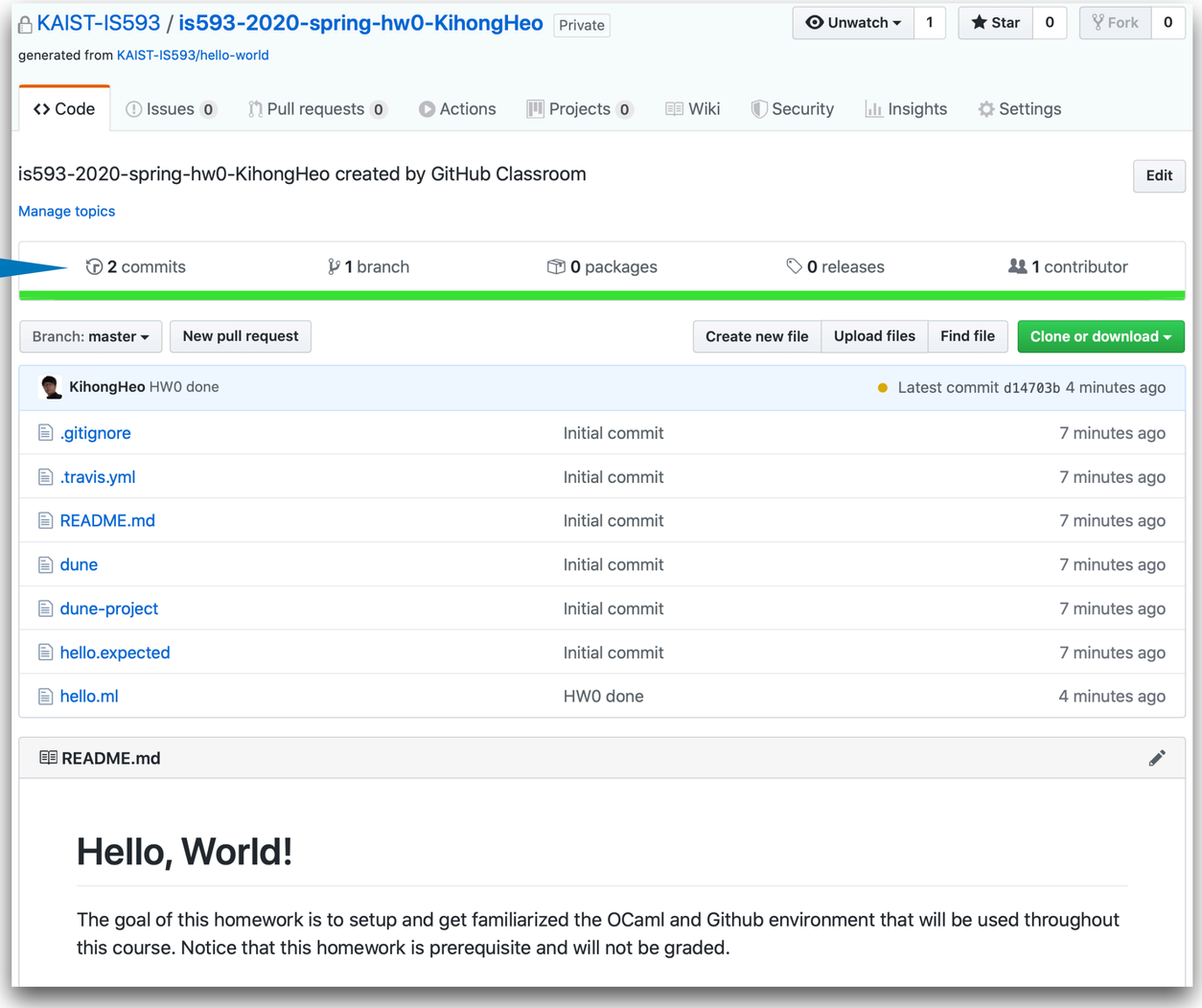
Homework 0: Hello, World!

1. Accept the invitation and have your repository



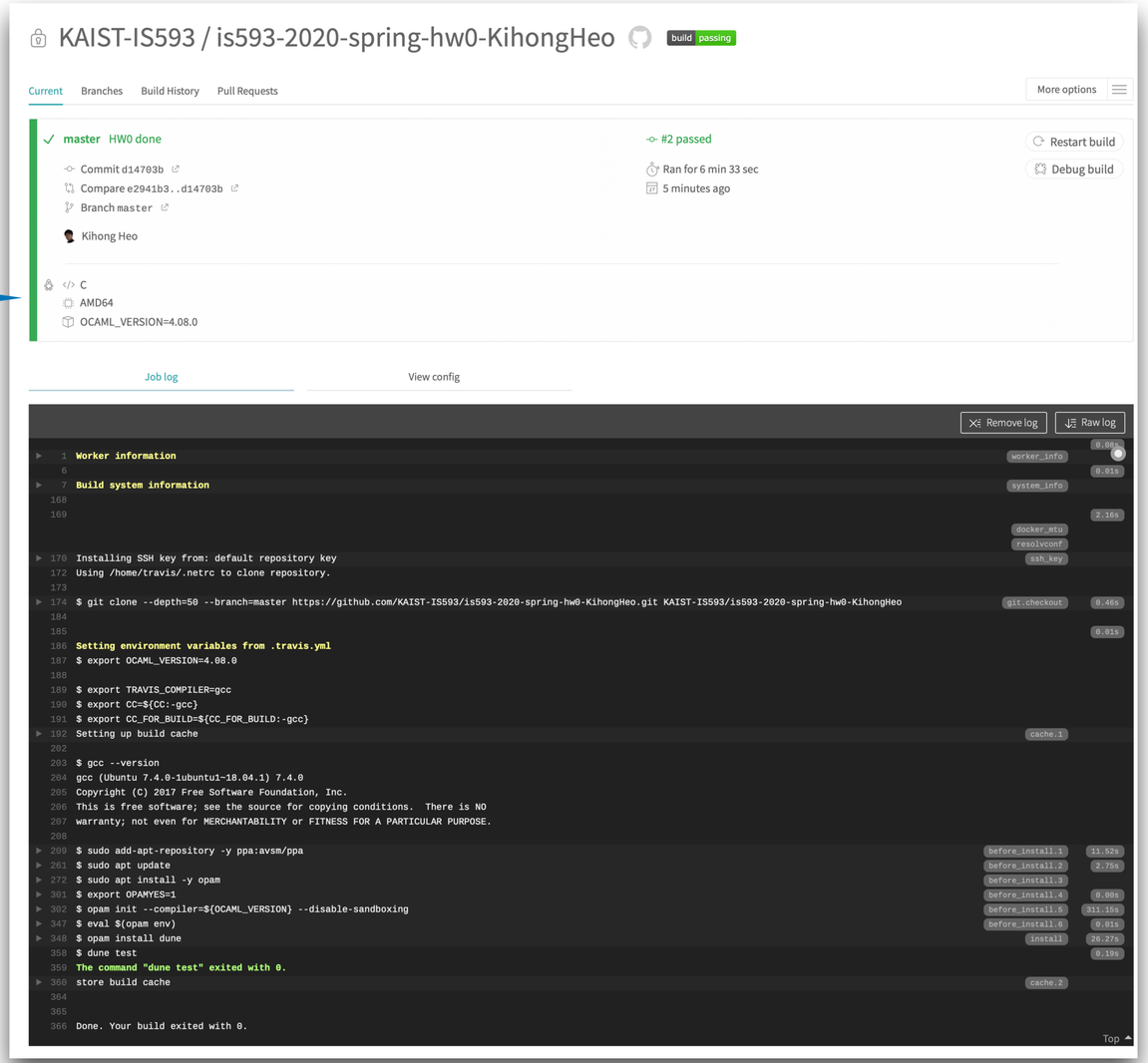
Homework 0: Hello, World!

2. Commit your code



Homework 0: Hello, World!

3. See your result



Misc

- Zoom: turn on your camera and mute by default
- Online office hours by appointment
- Any integrity violation (cheating, plagiarism, etc) : F
- Rules for programming assignments
  - Preserve the structures (directories, files, types, etc)
  - Don’t install further Github App
- Project: each student will give two 10-minute presentations (proposal & final)

Action Items

- See the issue board and do the followings: (TAs will provide links)
  - Submit your Github account
  - Pick 3 papers (one for each part) and sign up at the schedule (TA will randomly assign commentators for each paper)