

IS893: Advanced Software Security

1. Introduction

Kihong Heo



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

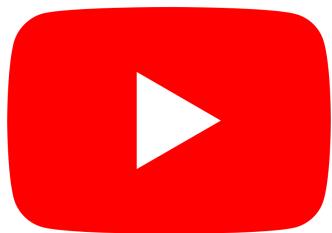
Course Information (2)

- 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

CNN U.S. | World | Politics | Money | Opinion | Health | Entertainment | Tech | Style | Travel | Sports | Video | Live TV | **US**

The 'Heartbleed' security flaw that affects most of the Internet

By Heather Kelly, CNN
Updated 5:11 PM ET, Wed April 9, 2014

A large red heart icon with a single drop of blood falling from its bottom, symbolizing the 'Heartbleed' bug.

This dangerous Android security bug could let anyone hack your phone camera

By Anthony Spadafra November 23, 2019

Camera app vulnerabilities allow attackers to remotely take photos, record video and spy on users

A smartphone lying on a keyboard, with binary code visible on its screen, illustrating a software vulnerability.

AERIAN MARSHALL / TRANSPORTATION 06:30 2019 07:00 AM

What Boeing's 737 MAX Has to Do With Cars: Software

Investigators believe faulty software contributed to two fatal crashes. A newly discovered fault will likely keep the 737 MAX grounded until the fall.

A Boeing 737 MAX airplane captured from a low angle, flying through a cloudy sky.

Homeland Security warns that certain heart devices can be hacked



New in Life & Style

Interfaith 4th-graders bond through poetry, art and Steph Curry 2:03 PM

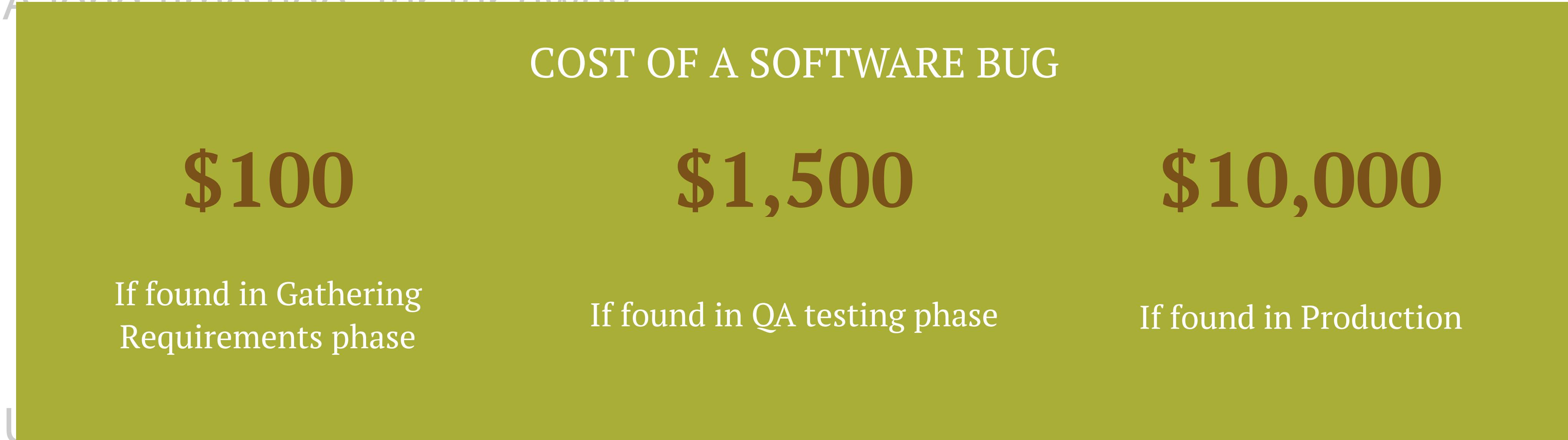
6 ways to celebrate Valentine's Day in Lake Geneva 8:55 AM

Six ways to keep your kids healthy during winter 8:56 AM

See More

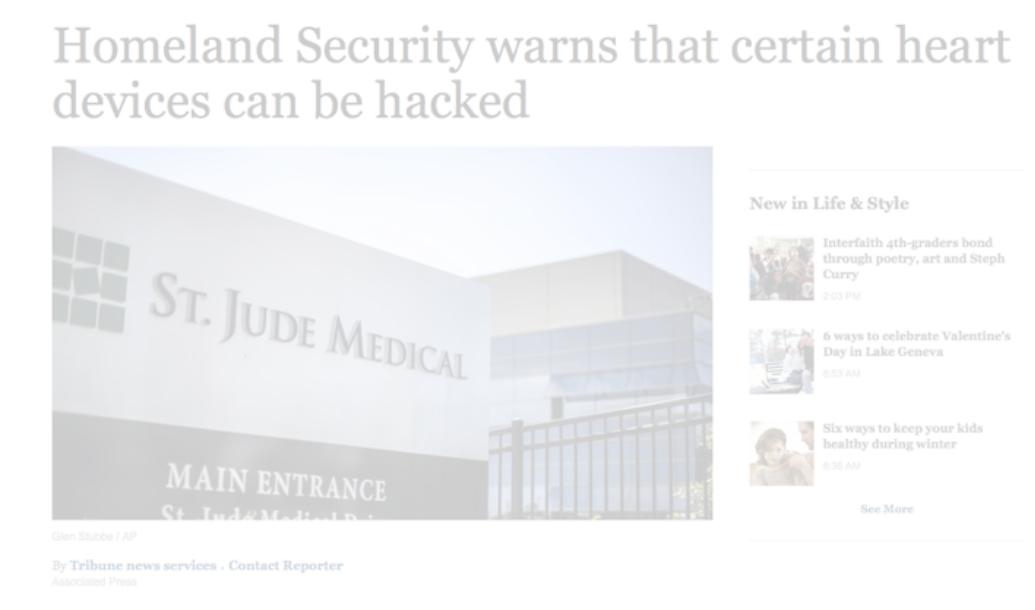
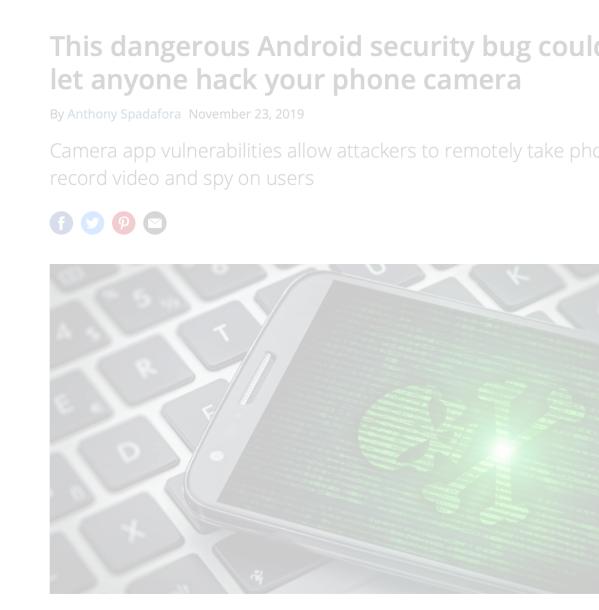
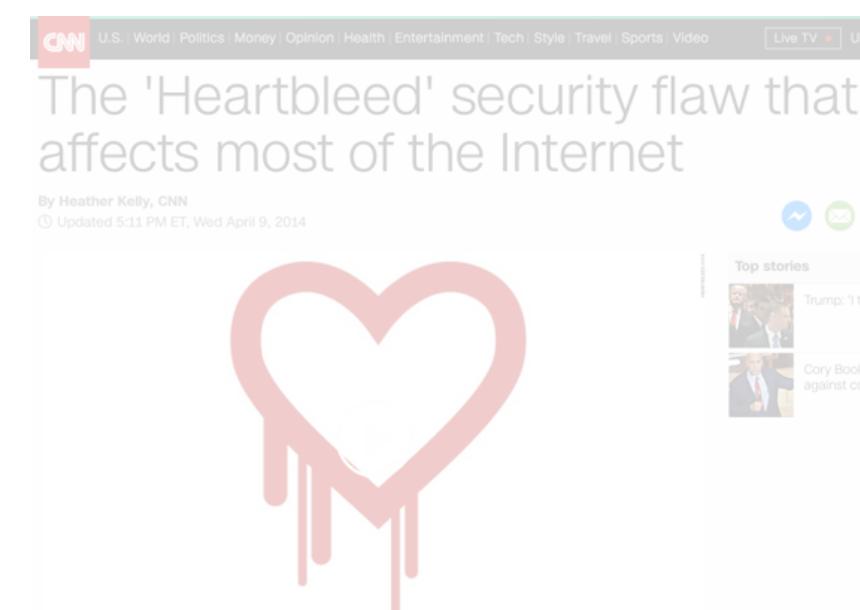
Software Bugs: A Persistent Problem

- A long time ago, for for everyone



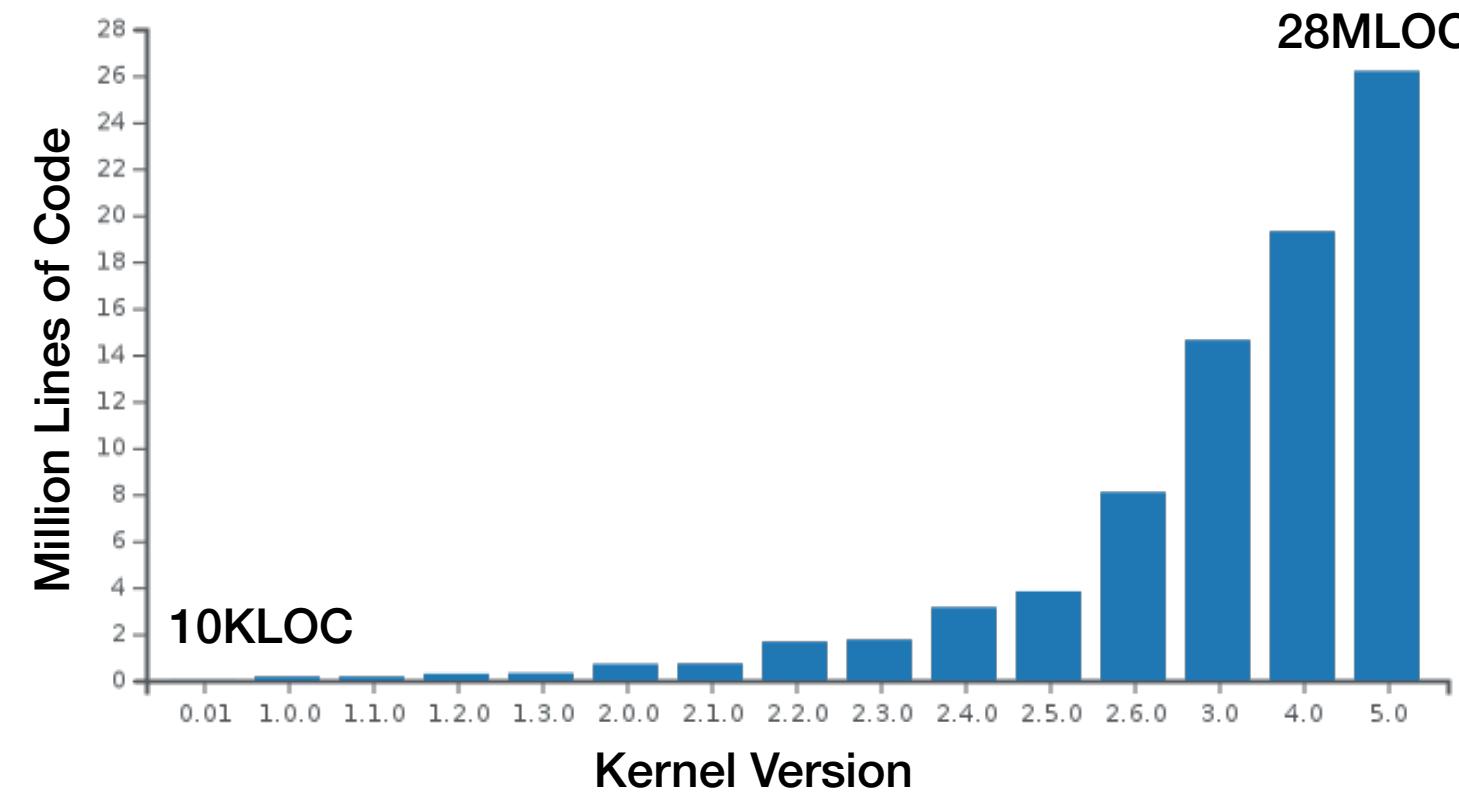
- Unfortunately, it's still true... your company probably has one.

- IBM Systems Sciences Institute, 2015

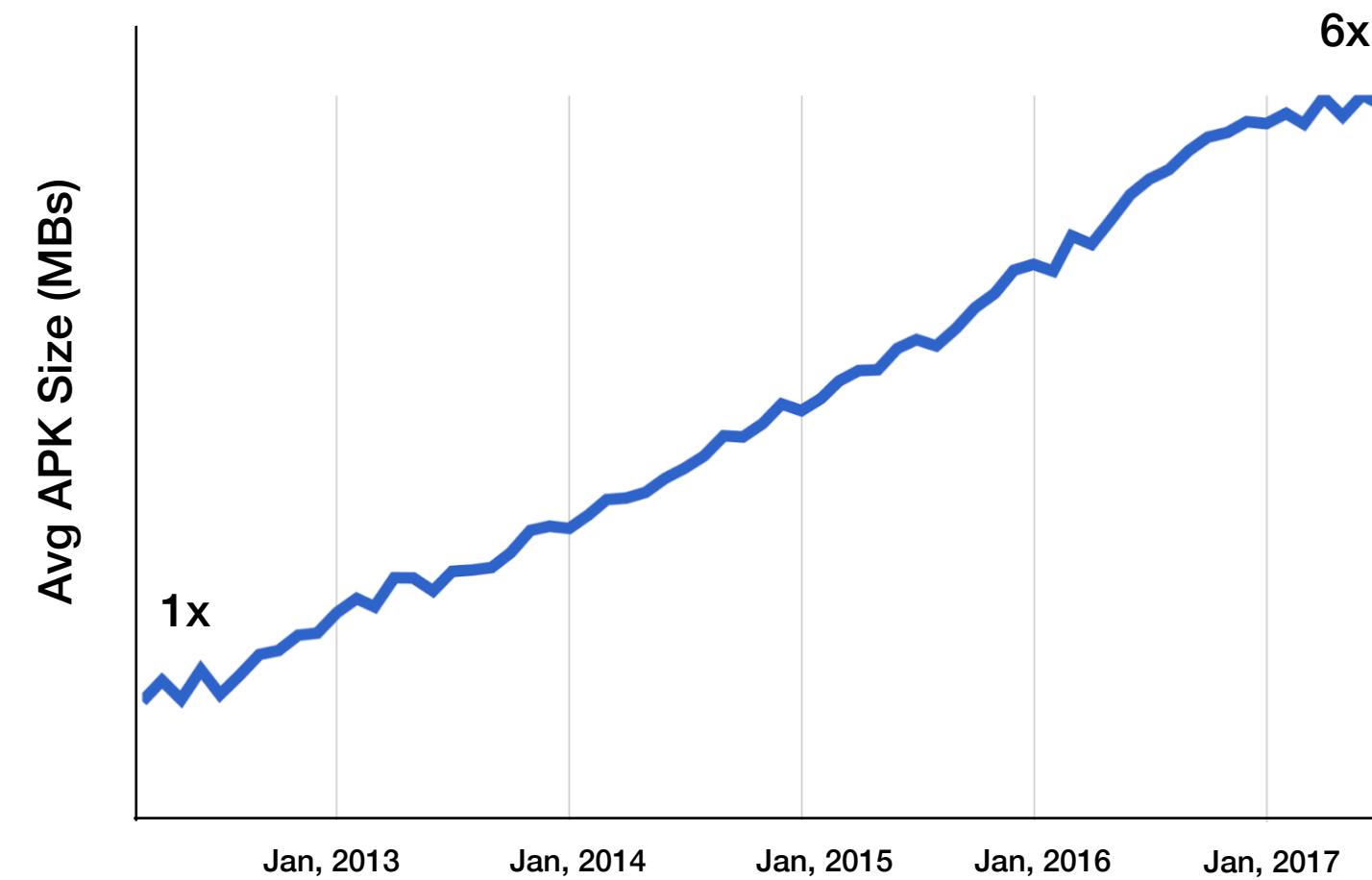


Why Software Still Fails?

Size of Linux Kernel



Avg. Size of Android Apps



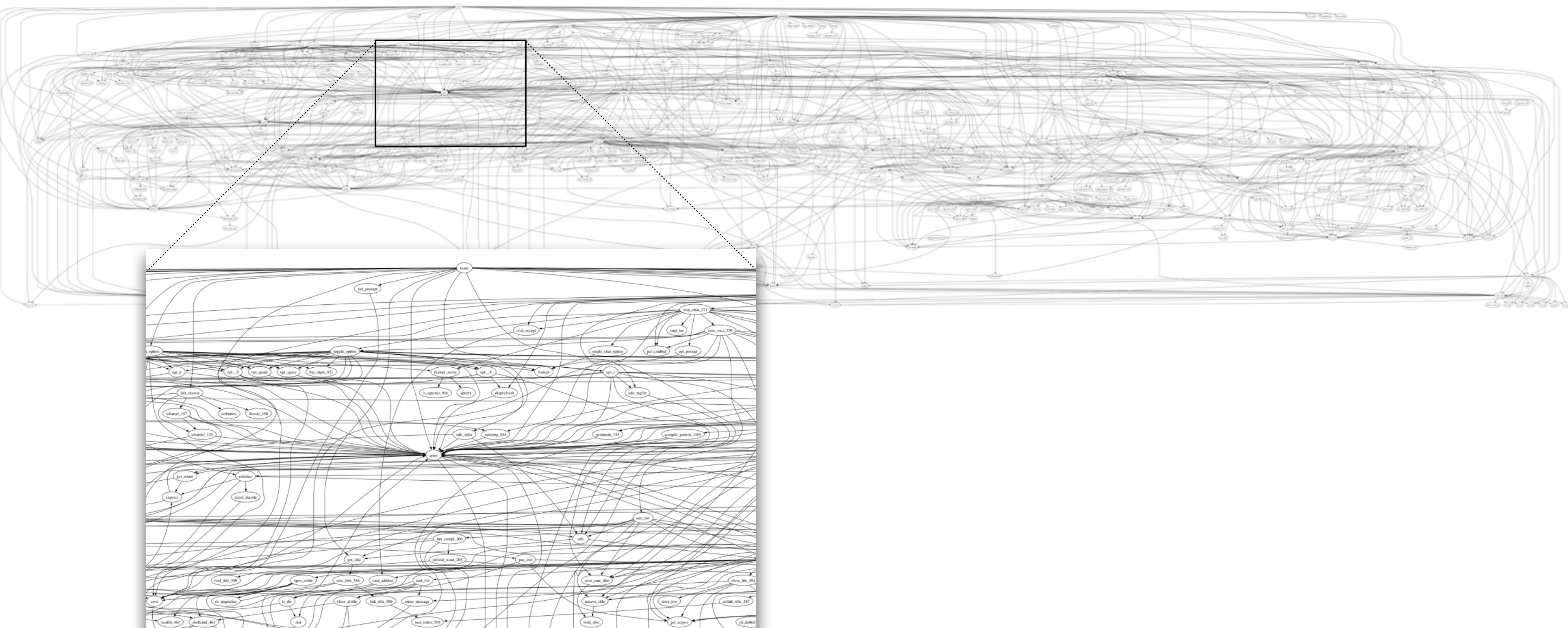
X



**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!

A screenshot of a Windows desktop showing the Google Chrome logo. The desktop background is the classic Windows 7/8 blue gradient with a window pane effect. The Chrome logo is centered on the screen.

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.

IBM Developer

Power developer portal Blogs Feedback

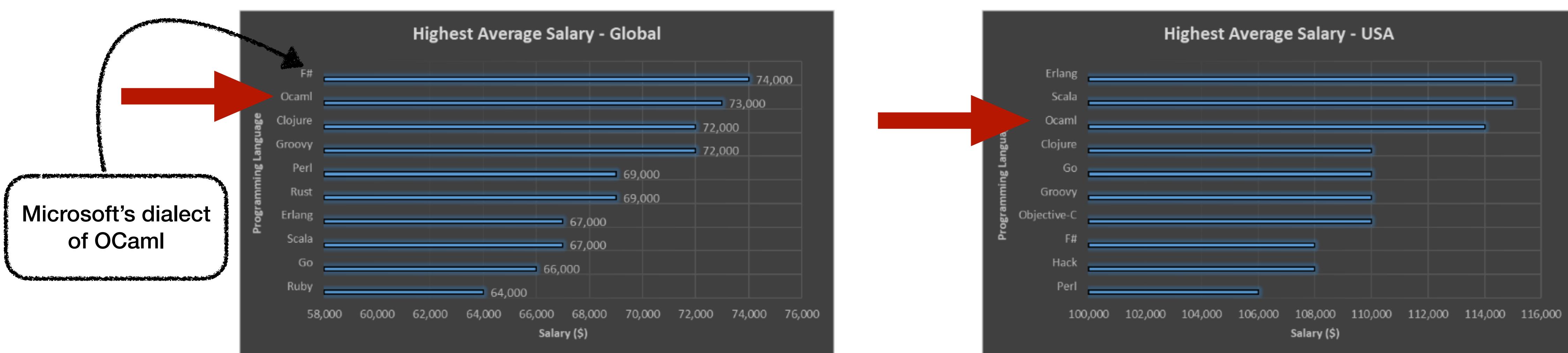
Announcements Compilers

IBM C/C++ and Fortran compilers to adopt LLVM open source infrastructure

SiyuanZhang
Published on February 23, 2020 / Updated on February 26, 2020

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
 $A = \text{false}$
 $B = \text{ture}$
 $C = \text{true}$

Satisfiability modulo theory (SMT)

$$x + 2 = y \implies f[\text{read } a, x, 3] = f[y - x + 1]$$

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

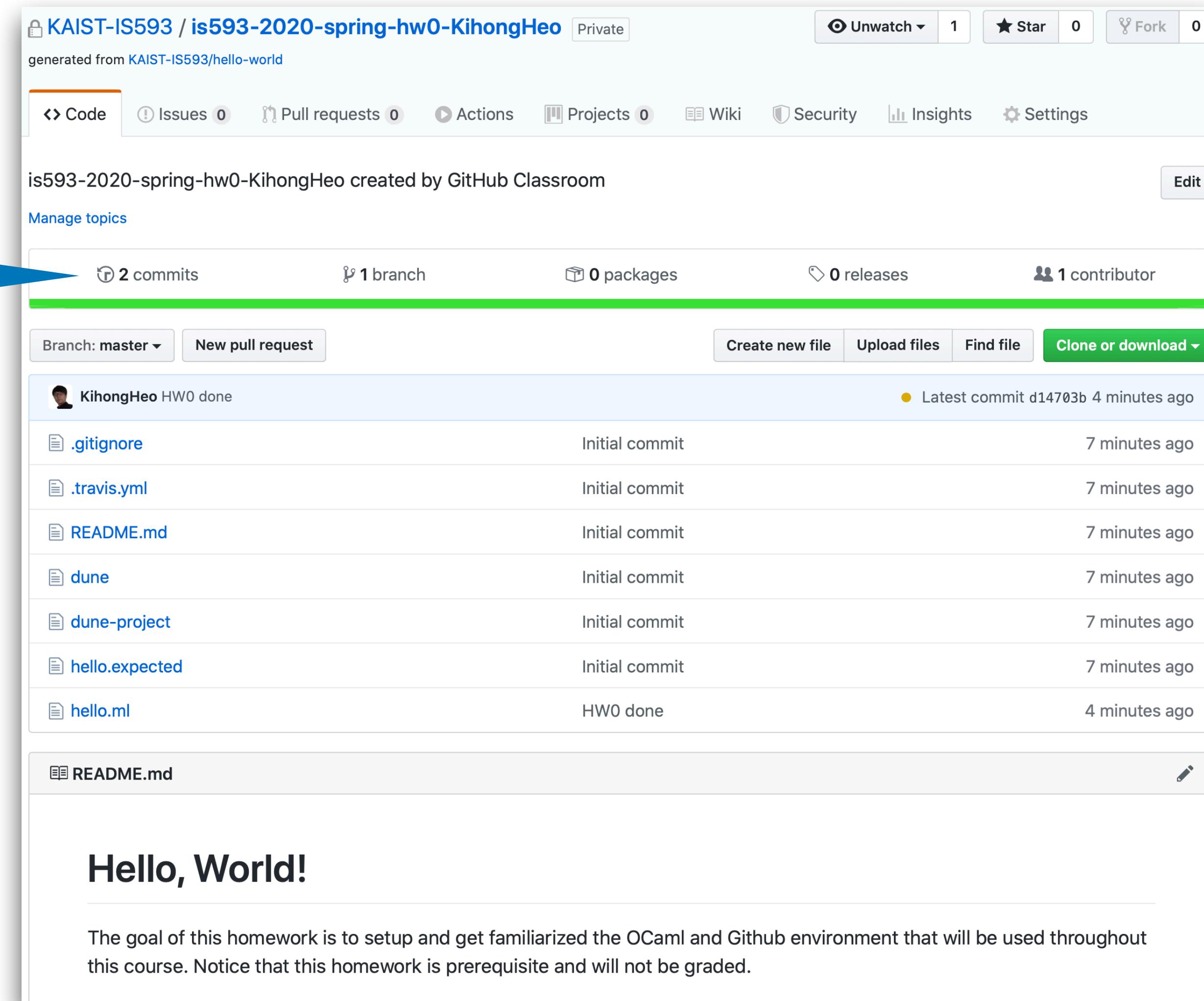
Homework 0: Hello, World!

1. Accept the invitation
and have your repository

The screenshot shows a GitHub repository page. At the top, it displays the repository name 'KAIST-IS593 / is593-2020-spring-hw0-KihongHeo' as private. It has 1 watch, 0 stars, and 0 forks. Below the header, there are tabs for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. A blue arrow points from the text '1. Accept the invitation and have your repository' to the 'Code' tab. The main content area shows the repository was created by GitHub Classroom. It has 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor (KihongHeo). The commit list shows an initial commit by KihongHeo, dated 1 minute ago, which includes files like .gitignore, .travis.yml, README.md, dune, dune-project, hello.expected, and hello.ml. Below the commit list is a large 'Hello, World!' message in bold. A note at the bottom states: 'The goal of this homework is to setup and get familiarized the OCaml and Github environment that will be used throughout this course. Notice that this homework is prerequisite and will not be graded.'

Homework 0: Hello, World!

2. Commit your code



The screenshot shows a GitHub repository page for 'KAIST-IS593 / is593-2020-spring-hw0-KihongHeo'. The repository is private, has 1 watch, 0 stars, and 0 forks. It was generated from 'KAIST-IS593/hello-world'. The repository was created by GitHub Classroom. It contains 2 commits, 1 branch, 0 packages, 0 releases, and 1 contributor (KihongHeo). The latest commit is 'd14703b' made 4 minutes ago. The commit history includes:

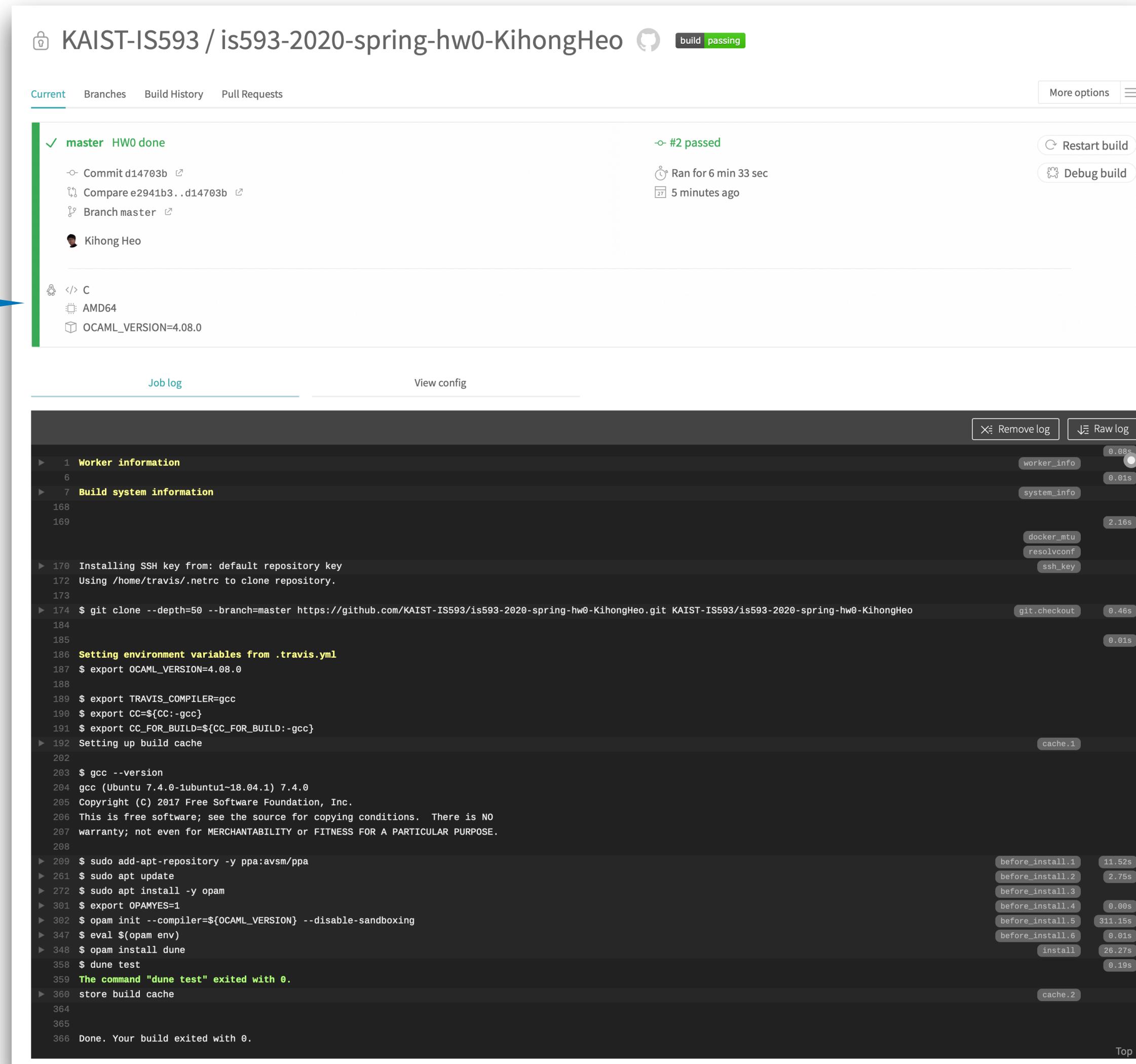
- KihongHeo HW0 done (Initial commit, 7 minutes ago)
- .gitignore (Initial commit, 7 minutes ago)
- .travis.yml (Initial commit, 7 minutes ago)
- README.md (Initial commit, 7 minutes ago)
- dune (Initial commit, 7 minutes ago)
- dune-project (Initial commit, 7 minutes ago)
- hello.expected (Initial commit, 7 minutes ago)
- hello.ml (HW0 done, 4 minutes ago)

The README.md file contains the text "Hello, World!".

The goal of this homework is to setup and get familiarized the OCaml and Github environment that will be used throughout this course. Notice that this homework is prerequisite and will not be graded.

Homework 0: Hello, World!

3. See your result



The screenshot shows a GitHub Actions build summary for the repository 'KAIST-IS593 / is593-2020-spring-hw0-KihongHeo'. The build status is 'passing'. The summary includes details about the commit (d14703b), comparison (e2941b3..d14703b), and branch (master). It also shows the author (Kihong Heo) and the environment (C, AMD64, OCAML_VERSION=4.08.0). Below the summary is a detailed job log showing the build process. The log starts with worker information, build system information, and cloning the repository. It then sets environment variables from .travis.yml, installs dependencies, and runs tests using dune. The log concludes with a success message: 'Done. Your build exited with 0.'

```
1 Worker information
6
7 Build system information
168
169

170 Installing SSH key from: default repository key
172 Using /home/travis/.netrc to clone repository.
173
174 $ git clone --depth=50 --branch=master https://github.com/KAIST-IS593/is593-2020-spring-hw0-KihongHeo.git KAIST-IS593/is593-2020-spring-hw0-KihongHeo
184
185
186 Setting environment variables from .travis.yml
187 $ export OCAML_VERSION=4.08.0
188
189 $ export TRAVIS_COMPILER=gcc
190 $ export CC=${CC:-gcc}
191 $ export CC_FOR_BUILD=${CC_FOR_BUILD:-gcc}
192 Setting up build cache
202
203 $ gcc --version
204 gcc (Ubuntu 7.4.0-1ubuntu1-18.04.1) 7.4.0
205 Copyright (C) 2017 Free Software Foundation, Inc.
206 This is free software; see the source for copying conditions. There is NO
207 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
208
209 $ sudo add-apt-repository -y ppa:avsm/ppa
210 $ sudo apt update
212 $ sudo apt install -y opam
213 $ export OPAMYES=1
214 $ opam init --compiler=${OCAML_VERSION} --disable-sandboxing
215 $ eval $(opam env)
216 $ opam install dune
217 $ dune test
218 The command "dune test" exited with 0.
219 store build cache
220
221 Done. Your build exited with 0.
```

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)