Introduction

2022.03.07

SWPP Practice Session

Seunghyeon Nam (with lots of derived works)

About practice session

- Software Development Principles & Practices
- Covers more practical issues related to actual development
- Monday 18:30 ~ 20:20 (KST), online session
- No attendance check, but come for your own benefit :)

Schedules (tentative)

- Week 1: Practice session intro & setup
- Week 2: Git tutorial
- Week 3~?: LLVM and IR
- Mid April~: Project introduction, collaboration, and tips

Sign Up for GitHub

- A web service for collaborative development

  https://github.com

- Create GitHub account and submit your username by 3/13!

  More details in this GitHub issue

- Announcements and updates will be posted on GitHub Issues
  - They will not be posted on eTL!

Development Environment

- Use Linux or macOS
- If you're new to Linux, try Ubuntu Desktop.

  Download Ubuntu Desktop
- Or, use WSL*Windows Subsystem for Linux* if you use Windows 10.

  Official WSL installation guide
- macOS users: Disable iCloud sync for your project directories!

Development Environment

- Your compiler should support C++17 standards
- LLVM and project skeletons use CMake

  Download Cmake

- Using Ninja is recommended for faster build

  Download Ninja

- You can also get CMake and Ninja via package managers

Development Environment

- We’ll use LLVM throughout this semester
  - Most assignments are about LLVM
  - Term project is based on LLVM
- Try building LLVM from source on your own!
  - First try getting used to CLI$_{\textit{command–line interface}}$ if you’re not familiar with it
  - Also, check if your development environment is well–configured

Development Environment

- install-llvm.sh
  - Start from this script if you're not familiar with build systems
  - Downloads and installs LLVM along with its dependencies
  - macOS users should slightly modify the script
  - swpp202201/practice/install-llvm.sh

Development Environment

We recommend using Visual Studio Code

Download Visual Studio Code

- Lightweight and portable (Windows, macOS, Linux, x86, ARM, …)
- Integrated git and GitHub functionalities
- Vast amount of extensions
- ~~Quicker response from TA~~

Development Environment

Useful extensions for Visual Studio Code

- C/C++: Syntax highlighting, error squiggle, autocomplete, formatting, file link, and many more!
- CMake: Quick configuration, build shortcuts
- LLVM: LLVM IR syntax highlighting

Development Environment

Use Remote extensions for remote server or WSL

- Remote – SSH for servers connected through SSH
- Remote – WSL ‘connects’ to Linux subsystem from Windows
- Most extensions can be installed on remote side as well

How to Ask Google

Google always have answers

- Well… almost always
- But you have to ‘properly’ ask them
- Coming up with good questions actually save your time and energy!

This is such a sad story
O panzar of the lake, what is your wisdon
They always ask what my wisdom is, never how my wisdom is

How to Ask Google

- DO: ask in short noun form
  - linux download file from url
  - adding object to c++ vector
- DON’T: come up with full sentence
  - How can I download files from url in linux terminal?
  - I want to add an object to a c++ vector

How to Ask Google

- DO: ask about error message ‘templates’
  - error: invalid use of ‘void’
  - error: binding reference of type [omit!] discards qualifiers
- DO: ask about library objects, functions, etc
  - llvm::PassManager
  - std::accumulate

How to Ask Google

- DON'T: include your object/function name
  - error: binding reference of type 'result::Result<std::unique_ptr<llvm::Module>, std::unique_ptr<std::exception> >&&' to 'std::remove_reference<const result::Result<std::unique_ptr<llvm::Module>, std::unique_ptr<std::exception> >&>::type' {aka 'const result::Result<std::unique_ptr<llvm::Module>, std::unique_ptr<std::exception> >'} discards qualifiers
  - Unfortunately, this isn't trivial in C++ due to complex template substitution rules

How to Ask Google

- DO: put the programming language name at the front
  - c++ int to float
  - python int to float
- DON'T: omit the language name
  - int to float → c++? JavaScript? LLVM IR?