모던 C++ 오픈소스 개발

CI 환경에서 일어난 일들

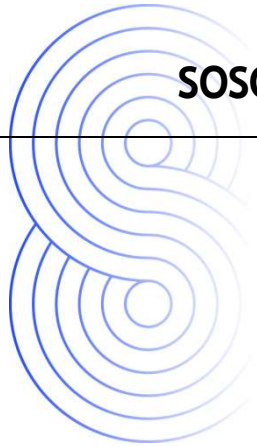
(주) 알체라 | R&D | 박동하

2018/10/18

SOSCON 2018
SAMSUNG OPEN SOURCE CONFERENCE 2018

세션 내용

발표자 소개



박동하: https://github.com/luncliff

일하는 곳:

(주) 알체라

크로스 플랫폼 라이브러리 설계/구현, 분산 시스템 구축

커뮤니티:

C++ Korea / C++, OpenSource Study

관심분야:

C++ Standard TS

동시성 프로그래밍, 코드 품질 …

#include <background>

C++은 ~~오래된~~ 발전 중인 언어
지속적인 표준 갱신. C++ 11 이후로는 Modern C++ (현재 C++ 17)

문법, 의미구조, 표준 라이브러리…

언어는 발전 하고 있는데… 그 이외의 부분은?

“저는 C++ 개발자입니다. "

주변의 반응


SNOW
Seed
MARIMITE
오타쿠는
아니라구요!!
애니메이션은
좋아하지만
설득력 없는 설득을 하는 사람이 있습니다.

C++는 변태들을 위한 언어?

플랫폼	Windows, Mac OS, Linux Distributions…
컴파일러	www.cppreference.com 에는 12 종류…
툴체인/SDK	GNU, LLVM, Visual C++ …
빌드시스템	Unix Makefiles, Visual Studio, Ninja, Eclipse, Gradle …

다른 언어는 쉽게 하던데….

장구한 진입장벽들

문법/표준을 제외하고도 **개발자에게 요구되는 지식/경험이 너무 많다.**

언어는 공부하면 되는데…

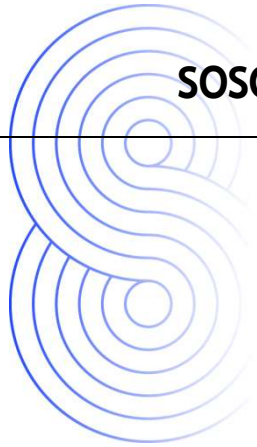
- 개발 환경 구성
- 툴체인 숙련도
- 라이브러리 관리
- …

https://www.youtube.com/watch?v=x1TsOHyJPpw (~35초)

Open Source with C++

https://github.com/fffaraz/awesome-cpp

Open Source ≒ DIY



코드가 공개되었다 != 바로 가져다 개발할 수 있다

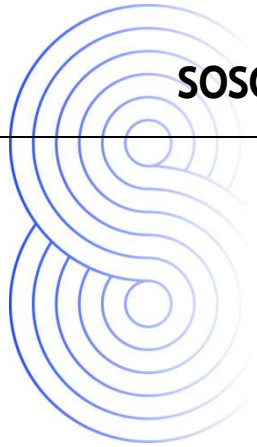
일단 프로젝트를 빌드 하는 것부터가 난관

Ex) https://github.com/ceres-solver/ceres-solver/

프로젝트 구축(?)단계에서 개발자가 겪는 문제들은
사용자들에게도 전파된다.

어떤 형태로 **설명서**를 제공할 것인가?

프로젝트가 보여줘야 하는 것?



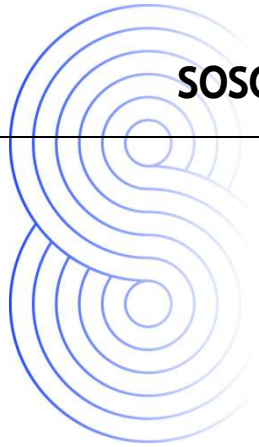
코드를 공개하는 것 만으로는 충분하지 않다

- 빌드/설치를 어떻게 하는가(절차)?
- 어떤 환경(조건)에서 이를 수행해야 하는가?

각각에 대해 답해보면…

- 절차: 빌드 시스템 파일, 혹은 Meta 정보
- 환경: 구체적인 설정 파일. CI 서비스 YAML + Log

필요한 능력?



- 환경에 대한 이해
  - Platform / Framework
  - Library
  - Architecture
  - …
- Toolchain에 대한 이해 // 삽질 경험
- Shell Script Programming // 필수

CMake (Cross-Platform Make)

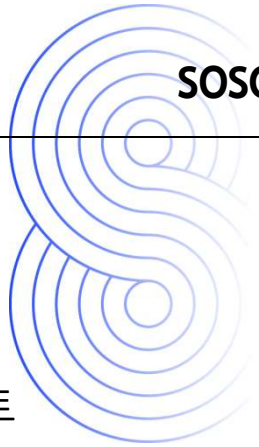
빌드 시스템 생성기(Build System Generator)
프로젝트 생성/빌드/설치 단계들을 추상화

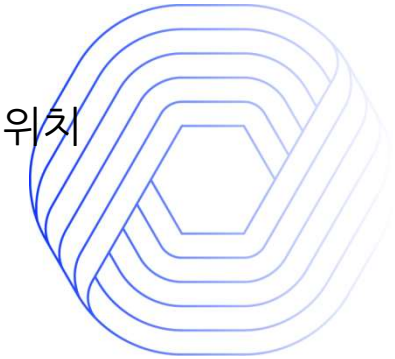
```
project(${ProjectName} LANGUAGES CXX)

add_executable(${PROJECT_NAME} main.cpp)

install(TARGETS ${PROJECT_NAME}
    DESTINATION ${CMAKE_INSTALL_PREFIX}
)
```

프로젝트

빌드 결과물

빌드 이후 설치 위치





CMake: Include/Link


```
target_include_directories(${PROJECT_NAME}
PRIVATE
    ${CMAKE_CURRENT_SOURCE_DIR}
)

target_link_libraries(${PROJECT_NAME}
PRIVATE
    GSL
)
```

CMake: 컴파일러 옵션


```
if(MSVC)
    target_compile_options(${PROJECT_NAME}
    PRIVATE
        /std:c++latest
    )
else()
    target_compile_options(${PROJECT_NAME}
    PRIVATE
        -std=c++17
    )
endif()
```

Continuous Integration

코드 변경내역에 대한 **지속적인** Build, Test, Deploy, Notify

빠른 Error/Bug 발견
변경 전/후 상태에 대한 확신 // 쓸 수 있는(Runnable) 코드

Status Badge: 코드의 현재 상태는 어떠한가? (신뢰성)

Marketplace

Continuous integration

SOSCON 2018

Search

Name or description

Now offering free trials
Free for 14 days

View apps

Categories

- Chat
- Code quality
- Code review
- Continuous integration
  - Mobile CI
  - Container CI

Automatically build and test your code as you push it to GitHub, preventing bugs from being deployed to production.

https://github.com/marketplace

Codefresh
A complete toolchain for delivering containers to production

Cloud 66 Skycap
Skycap is a container native CI/CD tool

Travis CI
Test and deploy with confidence

Buddy
One-click delivery automation for Web Developers

CircleCI
Automatically build, test, and deploy your project in minutes

Cloud 66 for Rails
Build, deploy, and maintain your Rails apps on any cloud or server

Percy
Continuous visual testing and reviews for web apps

Semaphore
Test and deploy at the push of a button

App Center
Continuously build, test, release, and monitor apps for every platform

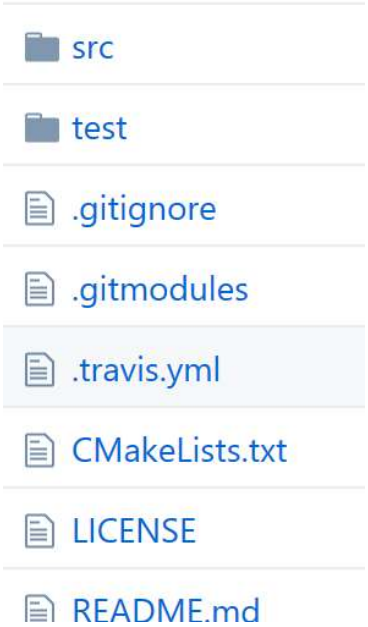
AppVeyor
Cloud service for building, testing and deploying Windows apps

Cirrus CI
Innovative and affordable CI for all

Azure Pipelines
Continuously build, test, and deploy to

Travis CI

- Non-Windows 에서 가장 보편적으로 사용되는 서비스
- 다소 불친절한 설명 … But 검색으로 해결 가능한 수준
- YAML 파일로 설정
  - 직관적으로 조합을 구성하기 쉬움
- **Public 프로젝트는 Free**



물론 여기에도 문제가 있습니다

프로젝트에서 필요한 환경 구성, 빌드 절차에 따라서 여러 이슈가 발생

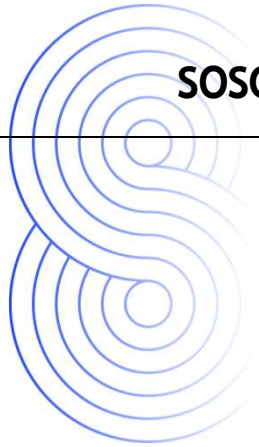
구축단계에서 **계획보다 많은 시간이 소요**될 소지가 매우매우 높음

엘리자가 말했어요

세상은 생각대로
되지 않는다고

XX

Case: 의존성 관리



필요한 프로그램이 있다? Installer를 쓴다

하지만 라이브러리는?

Package Manager

- **일관성** 있게 라이브러리를 관리할 수단
- C++ 라이브러리의 구성요소?
  - include / lib

C++ 개발자의 선택지?

- Conan : 문서화 Good. 여러가지 Tool에서 사용 가능
- **vcpkg** : Desktop으로 작업하는 경우 가장 간단한 방법
- NuGet : Visual Studio에서만 작업한다면 괜찮은 선택

vcpkg 설치

vcpkg 폴더에 라이브러리를 모아놓는 방식

설치

```
> git clone https://github.com/Microsoft/vcpkg.git
> cd vcpkg

PS> .\bootstrap-vcpkg.bat
Linux:~/$ ./bootstrap-vcpkg.sh
```

다운로드

```
PS> .\vcpkg install sdl2 curl
Linux:~/$ ./vcpkg install sdl2 curl
```

vcpkg 사용

CMake Toolchain 으로 전달

```
PS D:\src\cmake-test\build> cmake .. \
    "-DCMAKE_TOOLCHAIN_FILE=D:\src\vcpkg\scripts\buildsystems\vcpkg.cmake"
```

```
# CMakeLists.txt
cmake_minimum_required(VERSION 3.0)
project(test)

find_package(Sqlite3 REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main sqlite3)
```

vcpkg install sqlite

이제 CI에서 이걸…응?

실패: AppleClang 에서 Filesystem 지원 안함

```
156 CMake Error at CMakeLists.txt:10 (message):
157   Building the vcpkg tool requires support for the C++ Filesystem TS.
158
159   Apple clang versions 9 and below do not have support for it.
160
161   Please install gcc6 or newer from homebrew (brew install gcc6).
162
163   If you would like to try anyway, set VCPKG_ALLOW_APPLE_CLANG.
164
165
166 -- Configuring incomplete, errors occurred!
```

대신 귀여운 GCC를 드리겠습니다

실패: oclint가 기본 include path 에 위치. Overwrite 필요

```
==> Installing gcc
==> Downloading https://homebrew.bintray.com/bottles/gcc-8.2.0.high_sierra.bottl
==> Pouring gcc-8.2.0.high_sierra.bottle.1.tar.gz
Error: The `brew link` step did not complete successfully
The formula built, but is not symlinked into /usr/local
Could not symlink include/c++
Target /usr/local/include/c++
already exists. You may want to remove it:
  rm '/usr/local/include/c++'

To force the link and overwrite all conflicting files:
  brew link --overwrite gcc

To list all files that would be deleted:
  brew link --overwrite --dry-run gcc

Possible conflicting files are:
/usr/local/include/c++ -> /usr/local/Caskroom/oclint/0.13.1,17.4.0/oclint-0.13.1/include/c++
```

Oclint를 지우고 빌드 해보자!

```
==> Uninstalling Cask oclint
==> Unlinking Binary '/usr/local/bin/oclint'.
==> Unlinking Binary '/usr/local/bin/oclint-json-compilation-database'.
==> Unlinking Binary '/usr/local/bin/oclint-xcodebuild'.
==> Unlinking Binary '/usr/local/lib/oclint'.
==> Unlinking Binary '/usr/local/lib/clang'.
==> Unlinking Binary '/usr/local/include/c++'.
```

시스템 헤더 파일이 없어졌다(...)

```
                 from /usr/local/Cellar/gcc/8.2.0/include/c++/8.2.0/bits/ch
                 from /usr/local/Cellar/gcc/8.2.0/include/c++/8.2.0/string:
                 from ../include/vcpkg/base/chrono.h:4,
                 from ../src/vcpkg.cpp:13:
/usr/local/Cellar/gcc/8.2.0/lib/gcc/8/gcc/x86_64-apple-darwin17.7.0/8.2.0/i
_stdio.h: No such file or directory
 #include <_stdio.h>
          ^~~~~~~~~~
compilation terminated.
```

미리 빌드해놓으면?

Pre-release

2018

0178015

Prebuilt of vcpkg

luncliff released this on Sep 14

Assets 4

installed-win64.zip

vcpkg-mac.zip

Source code (zip)

Source code (tar.gz)

CI Build 에서 다운로드/실행

실패: GCC 라이브러리가 없어서 실행 X

```
154 dyld: Symbol not found: __ZNKSt19__codecvt_utf8_baseIwE10do_unshiftER11__mbstate_tPcS3_RS3_
155   Referenced from: /Users/travis/build/vcpkg/vcpkg
156   Expected in: /usr/lib/libstdc++.6.0.9.dylib
157  in /Users/travis/build/vcpkg/vcpkg
158 /Users/travis/.travis/job_stages: line 98:  1877 Abort trap: 6           vcpkg version
159 dyld: Symbol not found:   ZNKSt19  codecvt utf8 baseIwE10do unshiftER11  mbstate tPcS3 RS3
```

해치웠나?

vcpkg 의 라이브러리 설치 == **빌드**

```
▶ 118  $ if [ "${TRAVIS_OS_NAME}" == "osx" ]; then brew update > /dev/null 2>&1; brew cask unins
▶ 146  $ if [ "${TRAVIS_OS_NAME}" == "osx" ]; then wget
▼ 166  $ if [ "${TRAVIS_OS_NAME}" == "osx" ]; then export PATH=$PATH:${TRAVIS_BUILD_DIR}/vcpkg;
       vcpkg install tbb; vcpkg install opencv; fi
```

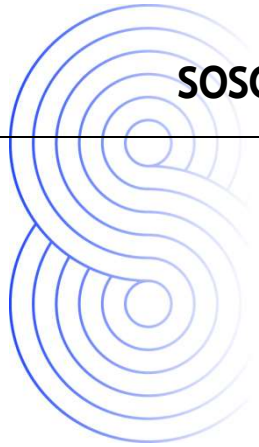
Intel TBB + OpenCV 를 설치하는데 필요한 시간을 구하시오. (5점)

install.1	178.09s
install.2	1.62s
install.3	1198.70s

Install GCC 8

Intel TBB + OpenCV

Case: 툴체인 지원 X



앞서 잠깐 언급된 상황:
표준 C++ 라이브러리를 지원하지 않는다면?
혹은 원하는 버전과 다르다면?

2가지 접근이 가능

- 표준 라이브러리를 통째로 설치 (!)
- 해당 기능을 우회/대체 (Boost를 사용할 수 있다면 비교적 수월)

접근 1: libc++ 설치

- libcxx @ cb9e96f
- libcxxabi @ 9245d48
- llvm @ 5136df4
- src
- test

Submodule + Script

```
# Build libcxx
- if [ "${TRAVIS_OS_NAME}" == "linux" ]; then
    mkdir -p prebuilt && pushd prebuilt;
    cmake ../libcxx -DLLVM_PATH=../llvm -DLIBCXX_CXX_ABI=libc
    make -j10 install;
    rm CMakeCache.txt;
    tree ./include;
    popd;
  fi
# Build libcxxabi
- if [ "${TRAVIS_OS_NAME}" == "linux" ]; then
    mkdir -p prebuilt && pushd prebuilt;
    cmake ../libcxxabi -DLLVM_PATH=../llvm/ -DLIBCXXABI_LIBCX
    make -j10 install;
    rm CMakeCache.txt;
    tree ./lib;
    popd;
  fi
```

접근 2: 우회


```
#ifdef HEARTHSTONEPP_WINDOWS
#include <filesystem>
#endif
#ifdef HEARTHSTONEPP_LINUX
#include <experimental/filesystem>
#endif
#ifdef HEARTHSTC
#include <sys/st
#endif

#ifndef HEARTHST
namespace filesy
#endif
```

마법의 #ifdef

```
42      #ifndef HEARTHSTONEPP_MACOSX
43          if (!filesystem::exists("Datas/" + email + ".json"))
44      #else
45          struct stat buf;
46          std::string path = "Datas/" + email + ".json";
47          if (stat(path.c_str(), &buf) == -1)
48      #endif
```

Case: NDK 라이브러리

Travis CI 이미지에 NDK가 없다! 수동 설치...

준비에 3~4분정도 소요....

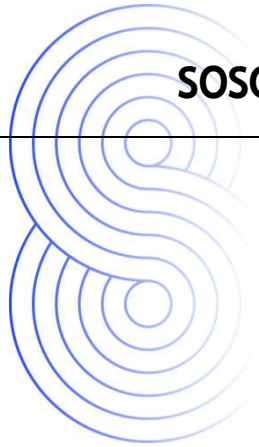
https://travis-ci.org/luncliff/Muffin

```
34    install:
35      - yes | sdkmanager --update > /dev/null 2>&1;
36      - sdkmanager ndk-bundle > /dev/null 2>&1;
37      - wget https://services.gradle.org/distributions/gradle-4.10.2-bin.zip;
38      - unzip -q gradle-4.10.2-bin.zip;
39      - export GRADLE=./gradle-4.10.2/bin/gradle
```

Azure Pipelines 에서는? 빌드 완료까지 2분

https://dev.azure.com/luncliff/personal/_build/results?buildId=76&view=logs

고민거리++



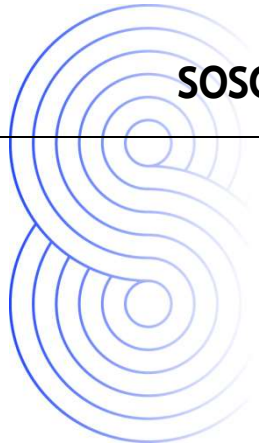
CI 환경에서 테스트는?

- 대부분 Unit Test에는 적합
- Emulator가 필수적인가? 설정하기 어려울수도
- CI 이미지에서는 사용할 수 없는 자원이 필요하다면? Ex) 카메라

복잡한 시나리오 == 개발 주기를 저해할 가능성

- 준비/확인을 위해 소비되는 시간이 크다면 재고가 필요함
- 빠른 것은 개발자의 Local Machine

Best Practice



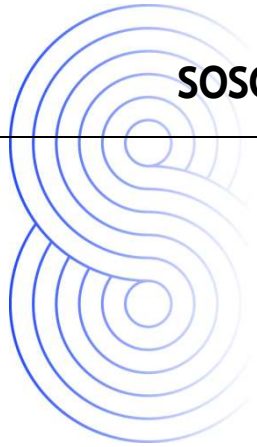
지원하지 않는 경우를 분명히 하라

플랫폼마다 프로그램 사용 방법을 계획하라

다른 프로젝트를 참고하라

빌드 조합은 필요한 만큼만 사용하라

마치면서



Status Badge는 없지만
참고할만한 YAML 파일을 가진 프로젝트들

https://github.com/google/benchmark

https://github.com/openssl/openssl

https://github.com/Microsoft/cpprestsdk

https://github.com/ocornut/imgui

Thank You!

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018