

# WSLでCmake

～CMAKEでビルドしてみよう～

# 導入

- ▶ Microsoft StoreでUbuntuを入れる
  - ▶ 18.04でも16.04でもいい。それ以外でもいいが古すぎるのはおすすめしない。自力で問題解決できる人は最新版でもいい。
  - ▶ ユーザ名とパスワードを設定する

# 導入

```
1  sudo apt-get update
2  sudo apt-get upgrade
3  sudo apt-get install vim
4  sudo apt-get update
5
6  mkdir /mnt/c/workspace
7  ln -s /mnt/c/workspace $HOME
8
9  # C C++
10 sudo apt install build-essential cmake
11
12 sudo apt install git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
13 # 並列処理と画像ファイルのライブラリ
14 sudo apt install libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev
15 # GUI系
16 sudo apt install libqt4-dev libqt4-opengl-dev libgtkglext1 libgtkglext1-dev
17
18 cd ~/workspace
19 git clone https://github.com/opencv/opencv.git
20 git clone https://github.com/opencv/opencv_contrib.git
21 cd opencv
22 mkdir build
23 cd build
24
25 cmake -D CMAKE_BUILD_TYPE=Release -D CMAKE_INSTALL_PREFIX=/usr/local
26 -D WITH_TBB=ON -D WITH_OPENGL=ON -D OPENCV_EXTRA_MODULES_PATH=~/workspace/opencv_contrib/modules -D
27 OPENCV_GENERATE_PKGCONFIG=ON ..
28 |
29 make -j7
30 sudo make install
```

# プログラミング言語

## ▶ コンパイラ言語

- ▶ C, C++, C#, Java, Pascal

## ▶ インタプリタ言語

- ▶ Python, Ruby, PHP, Javascript

# コンパイラ言語

- ▶ コンパイラ言語のソースコードをコンパイル（機械語）に直すプログラムのこと.
- ▶ 実行ファイルを動作させるには環境ごとにコンパイルする必要がある.

# ビルド（実行ファイルの生成）

1. 実行するためのソースコードや依存ファイルのリンク
2. 構成に適したコンパイラ

- ▶ ビルド（実行ファイルの生成）
  - ▶ コンパイル（機械語への変換）
    - ▶ メイク（ソースコードや依存ファイルのリンク）

makeコマンドでビルドする.

# Makeコマンド

- ▶ 依存関係の記述からターゲットとなるファイルを生成する
- ▶ 依存関係の記述があるファイルをmakefileという
- ▶ オブジェクト指向の場合、  
必要となる依存ファイルが膨大になったり依存関係が複雑になる場合が多い。この場合大量のmakefileが必要になることがある。

# Cmakeとは

- ▶ ビルドを自動化するためのフリーソフトウェア
- ▶ makefileを自動で生成してくれる
- ▶ 大規模プロジェクトの場合  
Cmakeのほうがビルドが楽になる場合がある
- ▶ Cmakeの設定ファイルはCMakeLists.txt



# Cmakeのバージョン確認

- ▶ Cmake --version で確認
- ▶ Cmakeのバージョンによってできることが違う
  - ▶ C++17の使用など
- ▶ 違うバージョンを入れたい場合は一度既存のものをアンインストールする。

```
1  sudo apt remove cmake
2
3  wget 欲しいバージョンのURL
4  chmod +x 上記でダウンロードしたファイル名
5  sudo ./上記でダウンロードしたファイル名
6
7  # 利用規約の同意やインストール先を聞かれるので同意するならyを二回
```

# CMakeLists.txtの例

**M** CMakeLists.txt

```
1  cmake_minimum_required(VERSION 3.16)
2  project(SandBox)
3
4  find_package(Boost REQUIRED COMPONENTS system)
5  find_package(Threads REQUIRED)
6  find_package(OpenCV REQUIRED)
7
8  include_directories(Boost::boost)
9
10 add_executable(main main.cpp)
11 target_include_directories(main
12     PRIVATE
13     ${OpenCV_INCLUDE_DIRS}
14     ${PROJECT_SOURCE_DIR}
15     ${LIBRARY_PATH}
16 )
17 target_link_libraries(main
18     Boost::system Threads::Threads
19     ${OpenCV_LIBS}
20     ${PROJECT_SOURCE_DIR}
21 )
22
23 set(TARGETS main)
24
25 set_property(TARGET ${TARGETS} PROPERTY CXX_STANDARD 17)
26 set_property(TARGET ${TARGETS} PROPERTY CXX_STANDARD_REQUIRED ON)
27 set_property(TARGET ${TARGETS} PROPERTY CXX_EXTENSION OFF)
```

cmake\_minimum\_required(VERSION 3.16) — Cmakeバージョン

project(SandBox) — プロジェクト名

find\_package(Boost REQUIRED COMPONENTS system)  
find\_package(Threads REQUIRED)  
find\_package(OpenCV REQUIRED) } — 使用パッケージ (lib)

include\_directories(Boost::boost) — ヘッダのあるディレクトリ

add\_executable(main main.cpp) — 実行ファイル名 追加するファイル...

target\_include\_directories(main  
PRIVATE  
\${OpenCV\_INCLUDE\_DIRS}  
\${PROJECT\_SOURCE\_DIR}  
\${LIBRARY\_PATH}  
) — ヘッダファイルがあるディレクトリ  
PATHを指定

target\_link\_libraries(main  
Boost::system Threads::Threads  
\${OpenCV\_LIBS}  
\${PROJECT\_SOURCE\_DIR}  
) — Libファイルがあるディレクトリ  
PATHを指定

set(TARGETS main) — 変数の設定

set\_property(TARGET \${TARGETS} PROPERTY CXX\_STANDARD 17)  
set\_property(TARGET \${TARGETS} PROPERTY CXX\_STANDARD\_REQUIRED ON)  
set\_property(TARGET \${TARGETS} PROPERTY CXX\_EXTENSION OFF) } — C++17を使うために  
必要な設定

対応している場合の記述

実行ファイルと  
依存ファイルの関係

# ビルドするファイルの場所

M CMakeLists.txt

```
1  cmake_minimum_required(VERSION 3.16)
2  project(SandBox)
3
4  find_package(Boost REQUIRED COMPONENTS system)
5  find_package(Threads REQUIRED)
6  find_package(OpenCV REQUIRED)
7
8  include_directories(Boost::boost)
9
10 add_executable(main main.cpp)
11 target_include_directories(main
12     PRIVATE
13     ${OpenCV_INCLUDE_DIRS}
14     ${PROJECT_SOURCE_DIR}
15     ${LIBRARY_PATH}
16 )
17 target_link_libraries(main
18     Boost::system Threads::Threads
19     ${OpenCV_LIBS}
20     ${PROJECT_SOURCE_DIR}
21 )
22
23 set(TARGETS main)
24
25 set_property(TARGET ${TARGETS} PROPERTY CXX_STANDARD 17)
26 set_property(TARGET ${TARGETS} PROPERTY CXX_STANDARD_REQUIRED ON)
27 set_property(TARGET ${TARGETS} PROPERTY CXX_EXTENSION OFF)
```

} /プロジェクトディレクトリ  
/CMakeLists.txt  
/ソースファイル

# ビルドファイルの構成

/プロジェクトディレクトリ

  /CMakeLists.txt

  /srcディレクトリ

  /includeディレクトリ

  /libsディレクトリ

# CMakeLists.txt

```
1  cmake_minimum_required(VERSION 3.10)
2  project(SandBox)
3
4  find_package(OpenCV REQUIRED)
5  add_executable(main src/main.cpp)
6
7  target_include_directories(main
8      PRIVATE
9      ${OpenCV_INCLUDE_DIRS}
10     ${PROJECT_SOURCE_DIR}/src
11     ${LIBRARY_PATH}
12 )
13 target_link_libraries(main
14     ${OpenCV_LIBS}
15     ${PROJECT_SOURCE_DIR}
16 )
```



実際に動かしてみよう

