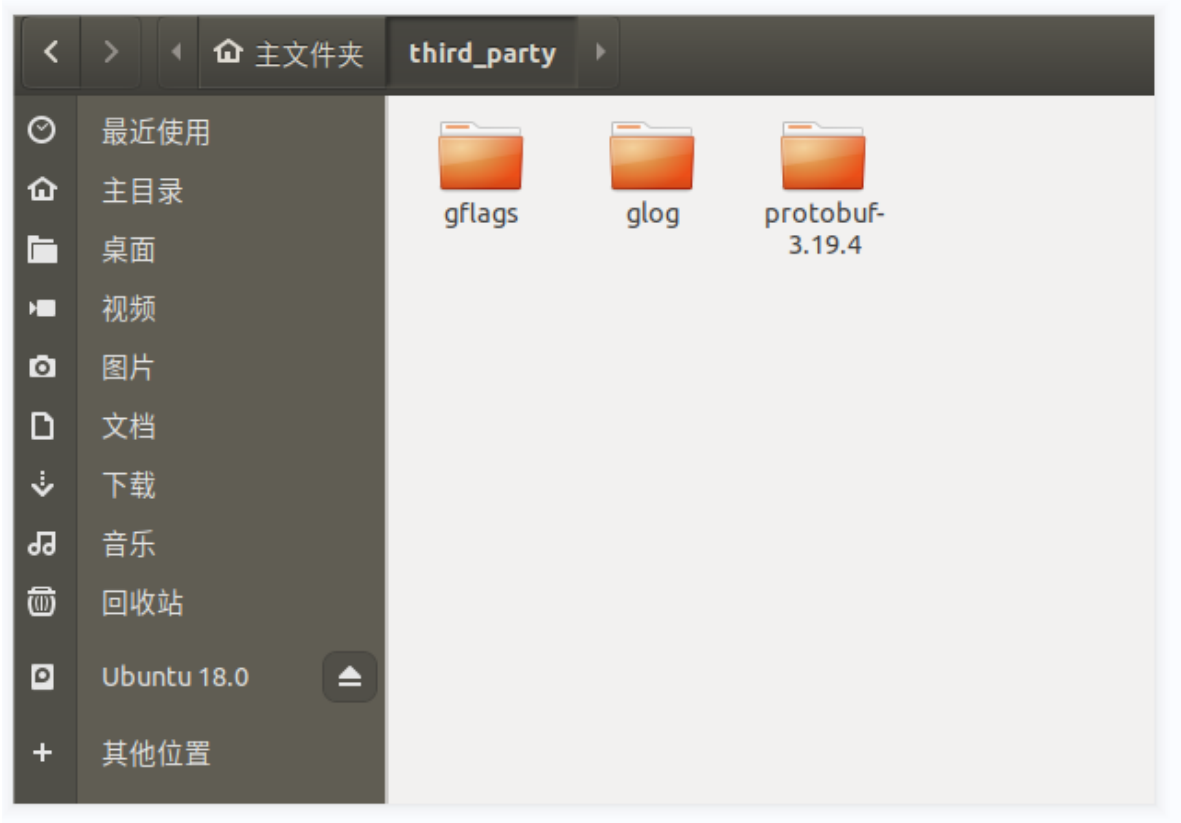
cmake学习笔记

1.third_party文件



protobuf、glog、gflags三个谷歌开源库安装在third_party文件下。

2.proto消息

编写了一个proto文件，使用cmake进行编译运行，结果如下：

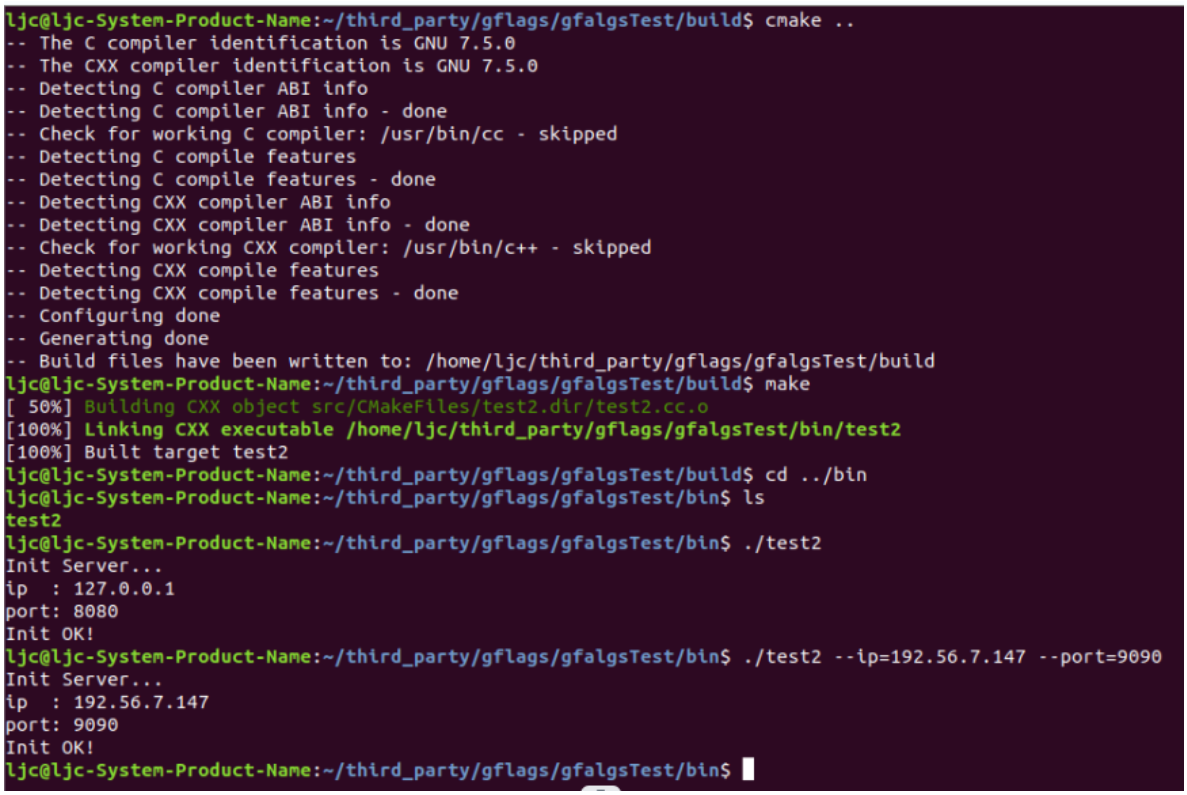
```
ljc@ljc-Systen-Product-Name:~/third_party/protobuf-3.19.4/cppTest/build$ cmake ..
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found Protobuf: /usr/local/lib/libprotobuf.so;-lpthread (found version "3.19.4")
-- protobuf library found
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ljc/third_party/protobuf-3.19.4/cppTest/build
ljc@ljc-Systen-Product-Name:~/third_party/protobuf-3.19.4/cppTest/build$ make
[ 25%] Running cpp protocol buffer compiler on AddressBook.proto
[libprotobuf WARNING google/protobuf/compiler/parser.cc:649] No syntax specified for the proto file: AddressBook.proto. Please use 'syntax = "proto2";' or 'syntax = "proto3";' to specify a syntax version.
 (Defaulted to proto2 syntax.)
[ 50%] Building CXX object src/CMakeFiles/cppTest.dir/main.cpp.o
[ 75%] Building CXX object src/CMakeFiles/cppTest.dir/AddressBook.pb.cc.o
[100%] Linking CXX executable /home/ljc/third_party/protobuf-3.19.4/cppTest/bin/cppTest
[100%] Built target cppTest
ljc@ljc-Systen-Product-Name:~/third_party/protobuf-3.19.4/cppTest/build$ cd ../bin
ljc@ljc-Systen-Product-Name:~/third_party/protobuf-3.19.4/cppTest/bin$ ls
cppTest
ljc@ljc-Systen-Product-Name:~/third_party/protobuf-3.19.4/cppTest/bin$ ./cppTest name.txt
name.txt: File not found.  Creating a new file.
Enter person ID number: 1
Enter name: xiaoming
Enter email address (blank for none): xiaoming@163.com
Enter a phone number (or leave blank to finish): 12345678
Is this a mobile, home, or work phone? home
Enter a phone number (or leave blank to finish):
ljc@ljc-Systen-Product-Name:~/third_party/protobuf-3.19.4/cppTest/bin$
```

3.使用cmake配置可执行文件和库文件分别放在bin和lib下。

```
set (EXECUTABLE_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/bin)
set (LIBRARY_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/lib)
```

4.使用gflags完成项目参数配置

编写test2.cc 实现利用gflags完成项目参数配置，使用cmake进行编译运行，结果如下：



5.使用glog完成日志输出（未完成）

编写glog.cpp文件利用glog完成日志输出，使用cmake进行编译运行，结果如下：

```
ljc@ljc-System-Product-Name:~/third_party/glog/gfalgsTest/build$ cmake ..
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ljc/third_party/glog/gfalgsTest/build
ljc@ljc-System-Product-Name:~/third_party/glog/gfalgsTest/build$ make
[ 50%] Building CXX object src/CMakeFiles/glog_test.dir/glog.cpp.o
[100%] Linking CXX executable /home/ljc/third_party/glog/gfalgsTest/bin/glog_test
[100%] Built target glog_test
ljc@ljc-System-Product-Name:~/third_party/glog/gfalgsTest/build$ cd ../bin
ljc@ljc-System-Product-Name:~/third_party/glog/gfalgsTest/bin$ ls
glog_test
ljc@ljc-System-Product-Name:~/third_party/glog/gfalgsTest/bin$ ./glog_test
Could not create log file: No such file or directory
COULD NOT CREATE LOGFILE '20221008-190249.23954'!
I20221008 19:02:49.473332 23954 glog.cpp:21] Hello GLOG
W20221008 19:02:49.473402 23954 glog.cpp:22] warning test
E20221008 19:02:49.473559 23954 glog.cpp:23] error test
F20221008 19:02:49.473642 23954 glog.cpp:24] fatal
*** Check failure stack trace: ***
    @     0x7f4167d95724  google::LogMessage::Fail()
    @     0x7f4167d95670  google::LogMessage::SendToLog()
    @     0x7f4167d94e9b  google::LogMessage::Flush()
    @     0x7f4167d98882  google::LogMessageFatal::~LogMessageFatal()
    @     0x55cc04bf3f02  main
    @     0x7f4167385c87  __libc_start_main
    @     0x55cc04bf3cba  _start
    @              (nil)  (unknown)
已放弃 (核心已转储)
ljc@ljc-System-Product-Name:~/third_party/glog/gfalgsTest/bin$
```

6.项目shiyongCMake工程构建（必须使用子模块），以proto消息为例：

外层CMakeLists.txt:

```
cmake_minimum_required(VERSION 3.10)
PROJECT (cppTest)
add_subdirectory(src)
```

内层CMakeLists.txt:

```
SET(SRC_LIST main.cpp)

# Find required protobuf package
find_package(Protobuf REQUIRED)
if(PROTOBUF_FOUND)
    message(STATUS "protobuf library found")
else()
    message(FATAL_ERROR "protobuf library is needed but cant be found")
endif()

include_directories(${PROTOBUF_INCLUDE_DIRS})
INCLUDE_DIRECTORIES(${CMAKE_CURRENT_BINARY_DIR})
```

```
PROTOBUF_GENERATE_CPP(PROTO_SRCS PROTO_HDRS AddressBook.proto)

ADD_EXECUTABLE(cppTest ${SRC_LIST} ${PROTO_SRCS} ${PROTO_HDRS})

target_link_libraries(cppTest ${PROTOBUF_LIBRARIES})
set (EXECUTABLE_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/bin)
set (LIBRARY_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/lib)
```