

Project3

Improve the Bw-tree

Project Structure

- Download the ***project3.tar.gz*** file from Piazza
 - It's on the Resources tab

Project Structure

- Decompress the downloaded file using the given command
 - *tar -xvzf project3.tar.gz*

```
(pwd  
/home/jaechan  
  
tar -xvzf project3.tar.gz  
project3/  
project3/test/  
project3/test/bwtree_test.cc  
project3/test/bwtree_test.cpp  
project3/test/CMakeLists.txt  
project3/bwtree/  
project3/bwtree/src/  
project3/bwtree/src/bwtree.cc  
project3/bwtree/CMakeLists.txt  
project3/bwtree/include/  
project3/bwtree/include/bloom_filter.h  
project3/bwtree/include/zipf.h  
project3/bwtree/include/sorted_small_set.h  
project3/bwtree/include/bwtree_test_util.h  
project3/bwtree/include/multithread_test_util.h  
project3/bwtree/include/bwtree.h  
project3/bwtree/include/index_logger.h  
project3/bwtree/include/macros.h  
project3/bwtree/include/atomic_stack.h  
project3/bwtree/include/worker_pool.h  
project3/BwtreeConfig.h.in  
project3/.clang-format  
project3/CMakeLists.txt  
project3/main.cc)
```

Project Structure

- You will have a directory named *project3*

```
ls | grep project3
project3
project3.tar.gz

cd project3

total 32K
drwxrwxr-x 4 jaechan jaechan 4.0K Oct 27 02:14 .
drwxr-xr-x 20 jaechan jaechan 4.0K Oct 27 02:18 ..
drwxrwxr-x 4 jaechan jaechan 4.0K Sep 29 2021 bwtree
-rw-rw-r-- 1 jaechan jaechan 223 Oct 25 16:26 BwtreeConfig.h.in
-rw-rw-r-- 1 jaechan jaechan 3.3K Sep 28 2021 .clang-format
-rw-rw-r-- 1 jaechan jaechan 946 Oct 25 16:26 CMakeLists.txt
-rw-rw-r-- 1 jaechan jaechan 610 Oct 27 01:56 main.cc
drwxrwxr-x 2 jaechan jaechan 4.0K Oct 27 01:36 test
```

Project Structure

- The directory structure of the **project3** directory is like below

```
tree project3
project3
├── bwtree
│   ├── CMakeLists.txt
│   ├── include
│   │   ├── atomic_stack.h
│   │   ├── bloom_filter.h
│   │   ├── bwtree.h
│   │   ├── bwtree_test_util.h
│   │   ├── index_logger.h
│   │   ├── macros.h
│   │   ├── multithread_test_util.h
│   │   ├── sorted_small_set.h
│   │   ├── worker_pool.h
│   │   └── zipf.h
│   └── src
│       └── bwtree.cc
└── BwtreeConfig.h.in
CMakeLists.txt
main.cc
test
    ├── bwtree_test.cc
    └── CMakeLists.txt

4 directories, 17 files
```

Hanyang University

Project Structure

```
tree project3
project3
└── bwtree
    ├── CMakeLists.txt
    └── include
        ├── atomic_stack.h
        ├── bloom_filter.h
        ├── bwtree.h
        ├── bwtree_test_util.h
        ├── index_logger.h
        ├── macros.h
        ├── multithread_test_util.h
        ├── sorted_small_set.h
        └── worker_pool.h
        └── zipf.h
    └── src
        └── bwtree.cc
    └── BwtreeConfig.h.in
    └── CMakeLists.txt
    └── main.cc
    └── test
        └── bwtree_test.cc
        └── CMakeLists.txt
4 directories, 17 files
```

- ***project3***
 - Project directory
- ***bwtree***
 - Library directory
 - ***libbwtree.a*** file is created by the codes in this directory
- ***test***
 - Test directory
 - Your test codes are placed here

Project Structure

```
tree project3
project3
└── bwtree
    ├── CMakeLists.txt
    └── include
        ├── atomic_stack.h
        ├── bloom_filter.h
        ├── bwtree.h
        ├── bwtree_test_util.h
        ├── index_logger.h
        ├── macros.h
        ├── multithread_test_util.h
        ├── sorted_small_set.h
        └── worker_pool.h
            └── zipf.h
    └── src
        └── bwtree.cc
    └── BwtreeConfig.h.in
    └── CMakeLists.txt
    └── main.cc
    └── test
        └── bwtree_test.cc
            └── CMakeLists.txt
4 directories, 17 files
```

- ***bwtree/include***

- Put your headers inside the include directory

- ***bwtree/src***

- Put your sources inside the source directory

- ***bwtree/CMakeLists.txt***

- You **MUST** edit this file if you add any source/header files for your library

Basic Compile

- *cmake -S . -B build*

```
cd ~/project3
pwd
/home/jaechan/project3

cd ~/project3
ls
bwtree BwtreeConfig.h.in CMakeLists.txt main.cc test

cd ~/project3
cmake -S . -B build
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found Python: /usr/bin/python3.8 (found version "3.8.10") found components: Interpreter
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Configuring done
-- Generating done
-- Build files have been written to: /home/jaechan/project3/build
```

Basic Compile

- *cmake --build build*

```
cd ~/project3 > cmake --build build
Scanning dependencies of target bwtree
[  7%] Building CXX object CMakeFiles/bwtree.dir/src/bwtree.cc.o
[ 14%] Linking CXX static library ../../lib/libbwtree.a
[ 14%] Built target bwtree
Scanning dependencies of target bwtreeplus
[ 21%] Building CXX object CMakeFiles/bwtreeplus.dir/main.cc.o
[ 28%] Linking CXX executable bin/bwtreeplus
[ 28%] Built target bwtreeplus
Scanning dependencies of target gtest
[ 35%] Building CXX object _deps/googletest-build/googletest/CMakeFiles/gtest.dir/src/gtest-all.cc.o
[ 42%] Linking CXX static library ../../../../lib/libgtest.a
[ 42%] Built target gtest
Scanning dependencies of target gtest_main
[ 50%] Building CXX object _deps/googletest-build/googletest/CMakeFiles/gtest_main.dir/src/gtest_main.cc.o
[ 57%] Linking CXX static library ../../../../lib/libgtest_main.a
[ 57%] Built target gtest_main
Scanning dependencies of target bwtree_test
[ 64%] Building CXX object test/CMakeFiles/bwtree_test.dir/bwtree_test.cc.o
[ 71%] Linking CXX executable ../../bin/bwtree_test
[ 71%] Built target bwtree_test
Scanning dependencies of target gmock
[ 78%] Building CXX object _deps/googletest-build/googlemock/CMakeFiles/gmock.dir/src/gmock-all.cc.o
[ 85%] Linking CXX static library ../../../../lib/libgmock.a
[ 85%] Built target gmock
Scanning dependencies of target gmock_main
[ 92%] Building CXX object _deps/googletest-build/googlemock/CMakeFiles/gmock_main.dir/src/gmock_main.cc.o
[100%] Linking CXX static library ../../../../lib/libgmock_main.a
[100%] Built target gmock_main
```

Executables

- Basic test and executables are in ***build/bin*** directory.

```
q ~ ~/project3 ➔ cd build  
q ~ ~/project3/build ➔ cd bin  
q ~ ~/project3/build/bin ➔ ls  
bwtreeplus bwtree_test
```

- ./bwtree_test***

```
q ~ ~/project3/build/bin ➔ ./bwtree_test  
Running main() from /home/jaechan/project3/build/_deps/googletest-src/googletest/src/gtest_main.cc  
[=====] Running 3 tests from 2 test suites.  
[-----] Global test environment set-up.  
[-----] 1 test from BwtreeInitTest  
[ RUN   ] BwtreeInitTest.HandlesInitialization  
[   OK  ] BwtreeInitTest.HandlesInitialization (0 ms)  
[-----] 1 test from BwtreeInitTest (0 ms total)  
  
[-----] 2 tests from BwtreeTest  
[ RUN   ] BwtreeTest.ConcurrentRandomInsert  
[   OK  ] BwtreeTest.ConcurrentRandomInsert (5417 ms)  
[ RUN   ] BwtreeTest.ConcurrentMixed  
[   OK  ] BwtreeTest.ConcurrentMixed (31056 ms)  
[-----] 2 tests from BwtreeTest (36473 ms total)  
  
[-----] Global test environment tear-down  
[=====] 3 tests from 2 test suites ran. (36474 ms total)  
[ PASSED ] 3 tests.
```

Debugging

Build in Debug/Release Mode

- To attach GDB, you need to build the project in a different way.

```
cd ~/project3
mkdir debug
cd debug
cmake -DCMAKE_BUILD_TYPE=Debug ..
```

-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
d-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found Python: /usr/bin/python3.8 (found version "3.8.10") found components: Interpreter
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Check if compiler accepts -pthread
-- Check if compiler accepts -pthread - yes
-- Configuring done
-- Generating done
-- Build files have been written to: /home/jaechan/project3/debug

```
cd ~/project3/debug
cmake --build .
Scanning dependencies of target bwtree
[ 7%] Building CXX object bwtree/CMakeFiles/bwtree.dir/src/bwtree.cc.o
[ 14%] Linking CXX static library ../lib/libbwtree.a
[ 14%] Built target bwtree
Scanning dependencies of target bwtreeplus
[ 21%] Building CXX object CMakeFiles/bwtreeplus.dir/main.cc.o
[ 28%] Linking CXX executable bin/bwtreeplus
[ 28%] Built target bwtreeplus
Scanning dependencies of target gtest
[ 35%] Building CXX object _deps/googletest-build/googletest/CMakeFiles/gtest.dir/src/gtest-all.cc.o
[ 42%] Linking CXX static library ../../lib/libgtest.a
[ 42%] Built target gtest
Scanning dependencies of target gtest_main
[ 50%] Building CXX object _deps/googletest-build/googletest/CMakeFiles/gtest_main.dir/src/gtest_main.cc.o
[ 57%] Linking CXX static library ../../lib/libgtest_main.a
[ 57%] Built target gtest_main
Scanning dependencies of target bwtree_test
[ 64%] Building CXX object test/CMakeFiles/bwtree_test.dir/bwtree_test.cc.o
[ 71%] Linking CXX executable ..../bin/bwtree_test
[ 71%] Built target bwtree_test
Scanning dependencies of target gmock
[ 78%] Building CXX object _deps/googletest-build/googlemock/CMakeFiles/gmock.dir/src/gmock-all.cc.o
[ 85%] Linking CXX static library ../../lib/libgmock.a
[ 85%] Built target gmock
Scanning dependencies of target gmock_main
[ 92%] Building CXX object _deps/googletest-build/googlemock/CMakeFiles/gmock_main.dir/src/gmock_main.cc.o
[100%] Linking CXX static library ../../lib/libgmock_main.a
[100%] Built target gmock_main
```

For Release: *cmake -DCMAKE_BUILD_TYPE=Release ..*

Project Structure

Debug

```
[-----] 2 tests from BwtreeTest
[ RUN    ] BwtreeTest.ConcurrentRandomInsert
[      OK ] BwtreeTest.ConcurrentRandomInsert (5540 ms)
[ RUN    ] BwtreeTest.ConcurrentMixed
[      OK ] BwtreeTest.ConcurrentMixed (30805 ms)
[-----] 2 tests from BwtreeTest (36346 ms total)

[-----] Global test environment tear-down
[=====] 3 tests from 2 test suites ran. (36347 ms total)
[  PASSED ] 3 tests.
```

Release

```
[-----] 2 tests from BwtreeTest
[ RUN    ] BwtreeTest.ConcurrentRandomInsert
[      OK ] BwtreeTest.ConcurrentRandomInsert (2438 ms)
[ RUN    ] BwtreeTest.ConcurrentMixed
[      OK ] BwtreeTest.ConcurrentMixed (8865 ms)
[-----] 2 tests from BwtreeTest (11303 ms total)

[-----] Global test environment tear-down
[=====] 3 tests from 2 test suites ran. (11303 ms total)
[  PASSED ] 3 tests.
```

Totally different in terms of performance

GDB

- Use GDB to analyze the Bw-tree

```
q ~ ~/project3/debug cd bin
q ~ ~/project3/debug/bin gdb ./bwtreeplus
GNU gdb (Ubuntu 10.2-0ubuntu1~20.04~1) 10.2
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./bwtreeplus...
(gdb) b Insert
Breakpoint 1 at 0x550b: Insert. (5 locations)
(gdb) run
Starting program: /home/jaechan/project3/debug/bin/bwtreeplus
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0xfffff6dff700 (LWP 2902819)]
[New Thread 0xfffff65fe700 (LWP 2902820)]
```

```
[New Thread 0x7ffffef5f0700 (LWP 2902859)]
--Type <RET> for more, q to quit, c to continue without paging--
[Switching to Thread 0x7ffffedded700 (LWP 2902841)]

Thread 24 "bwtreeplus" hit Breakpoint 1, bwtree::BwTree<long, long, test::BwTreeTestUtil::KeyComparator,
  key=@0x7ffffeddeb488: 8, value=@0x7ffffeddeb490: 8, unique_key=false) at /home/jaechan/project3/bwtree
6566      NO_ASAN bool Insert(const KeyType &key, const ValueType &value, bool unique_key = false) {
(gdb) 
  □ project3 ➤ 119d 4h 57m ➤ 1 vi ➤ 2 gdb
```

Appendix

GoogleTest

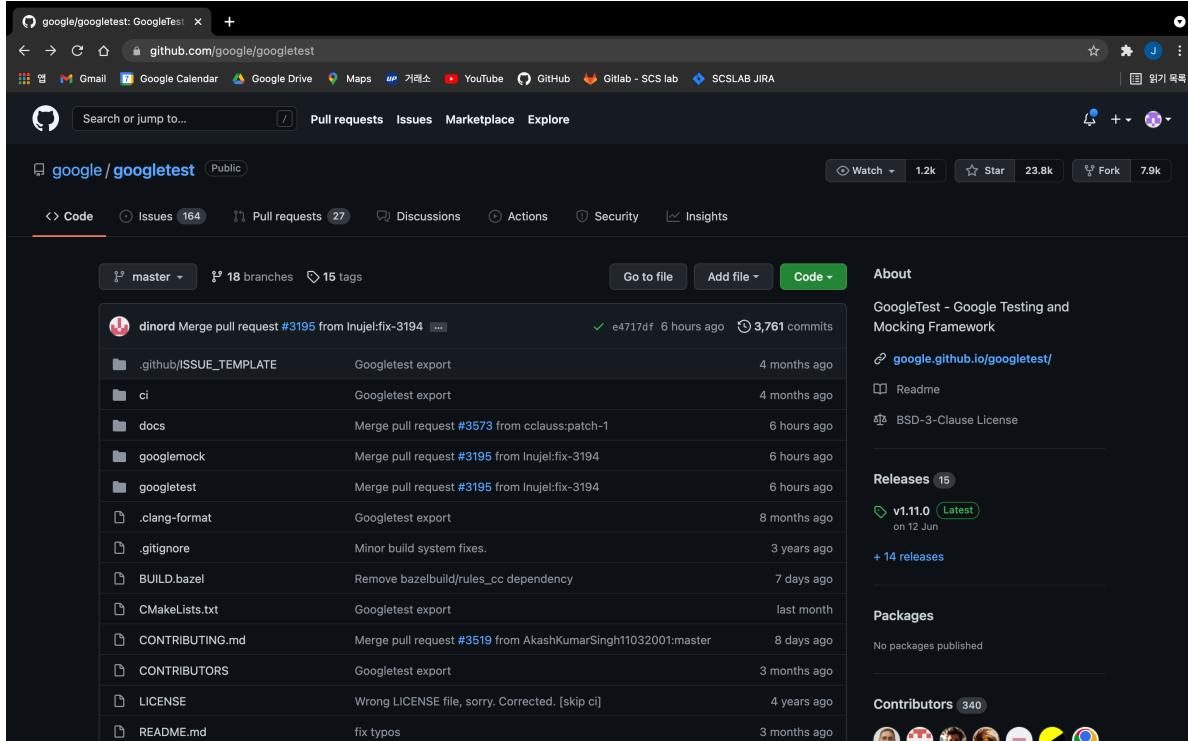
GoogleTest

```
q ~ c/googletest touch CMakeLists.txt  
q ~ c/googletest touch basic_test.cc  
q ~ c/googletest ls  
basic_test.cc CMakeLists.txt  
q ~ c/googletest
```

- Create a file named ‘CMakeList.txt’
- Create a file named ‘basic_test.cc’
- Make sure the files are within the directory that you’ve created (i.e., googletest)

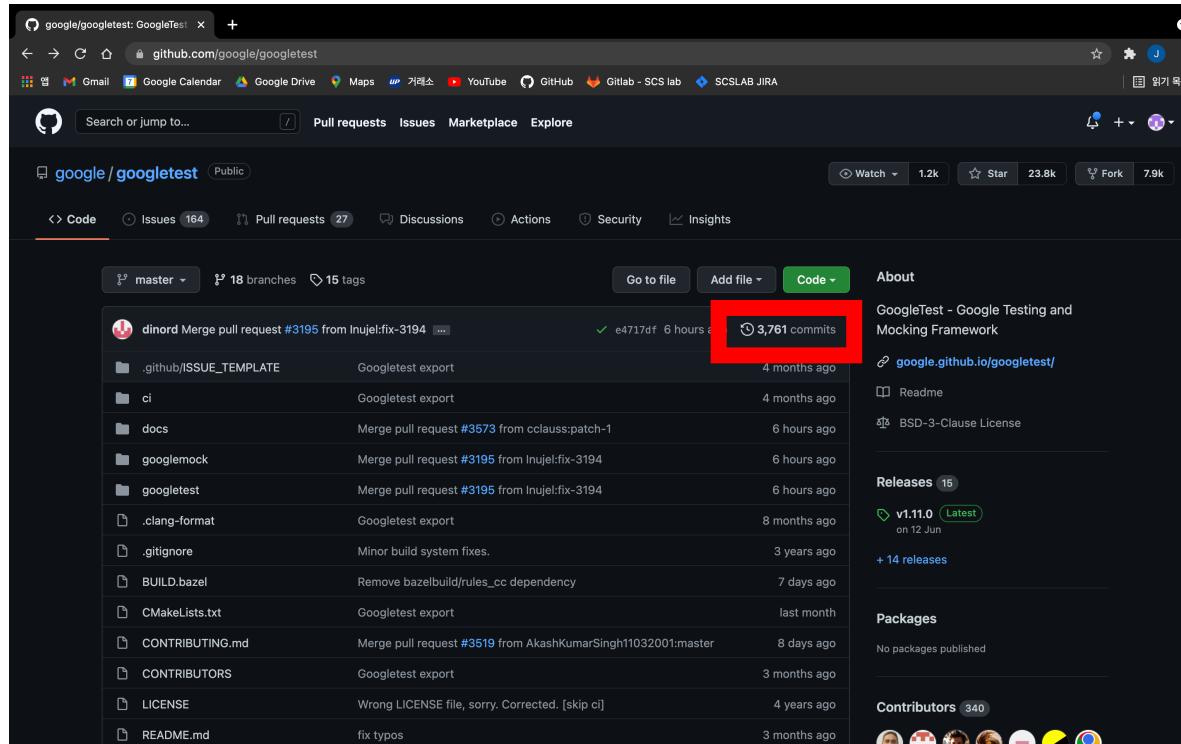
GoogleTest

- Go to the googletest github repository
- <https://github.com/google/googletest>



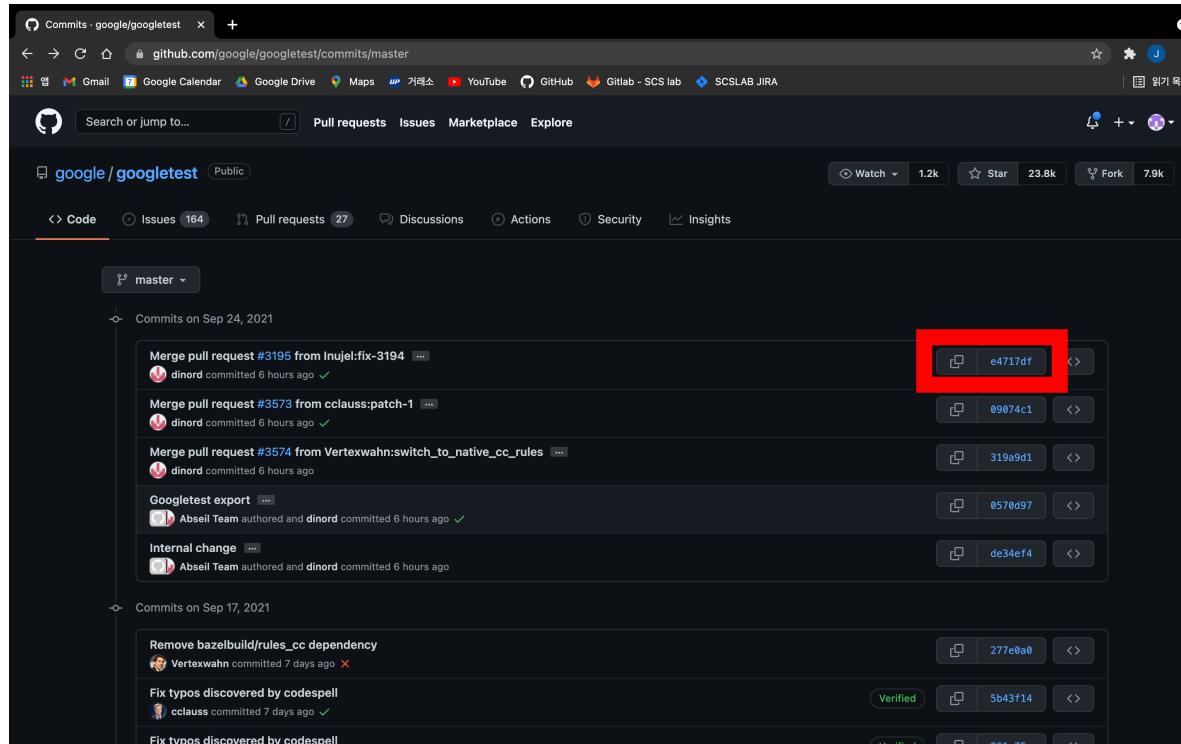
GoogleTest

- Press ‘commits’ as in the red box in the picture below



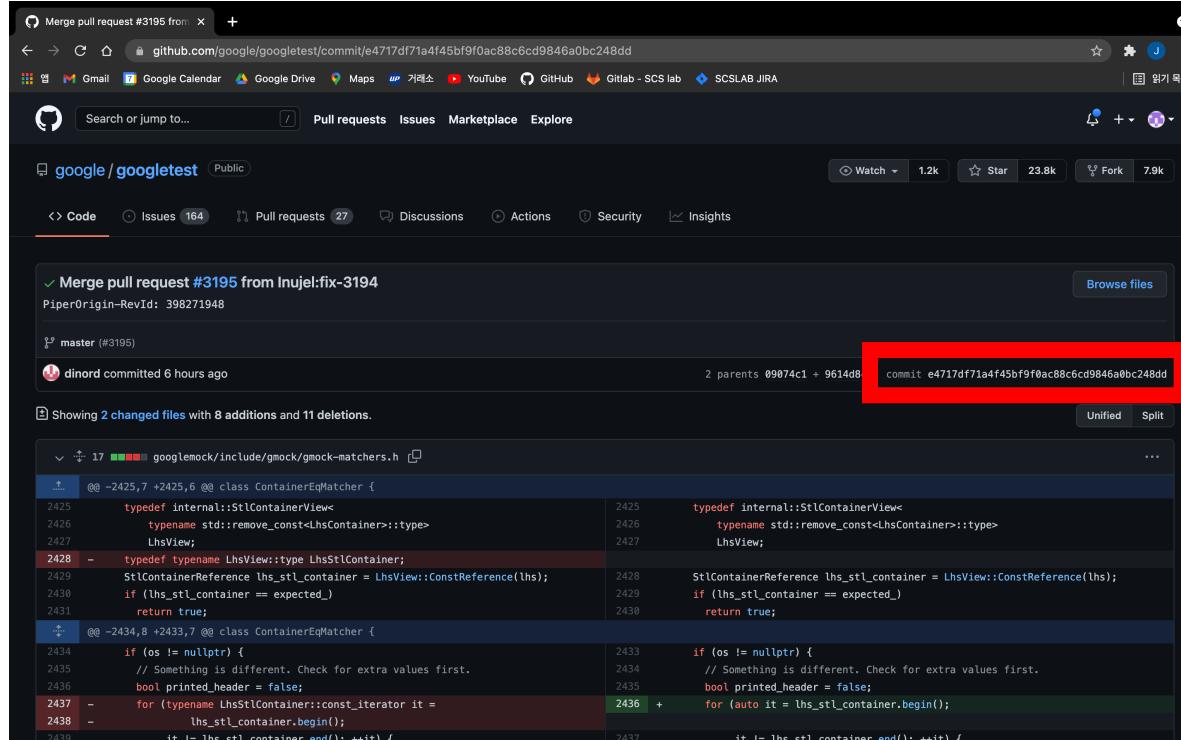
GoogleTest

- Press the latest commit as in the red box in the picture below



GoogleTest

- Memo the latest commit's hash somewhere
- We will use it in the following steps

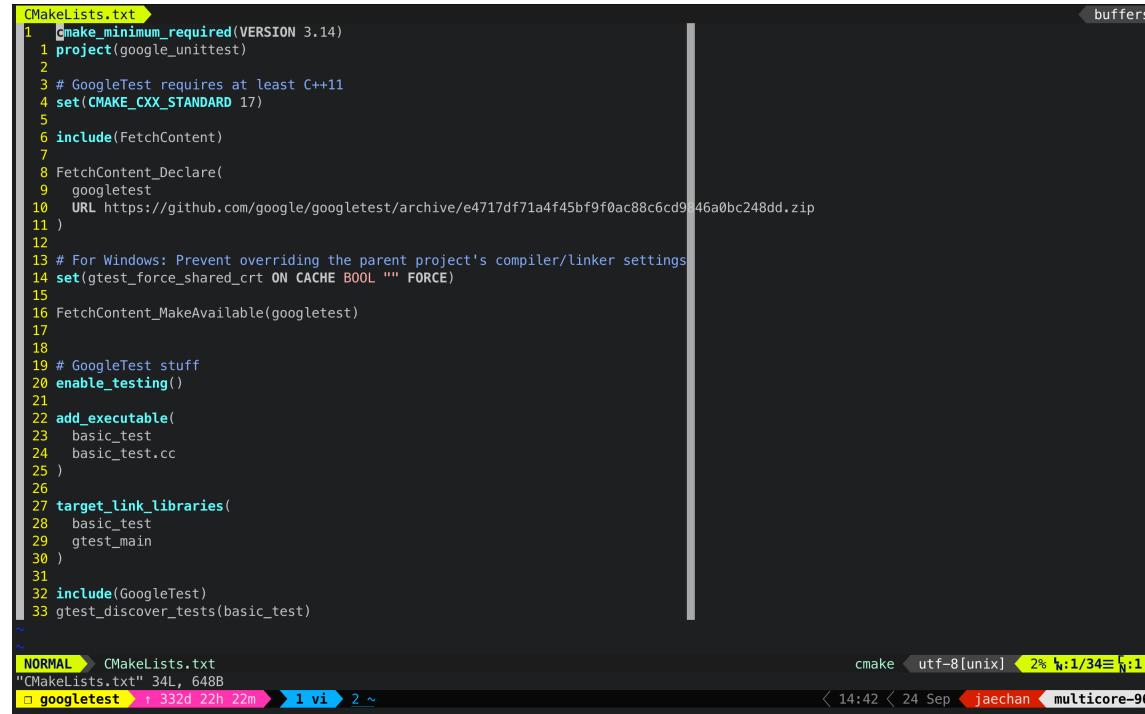


The screenshot shows a GitHub pull request page for the 'google/googlemock' repository. The pull request is titled 'Merge pull request #3195 from Inujel:fix-3194'. The commit hash 'e4717df71a4f45bf9f0ac88c6cd9846a0bc248dd' is highlighted with a red box. The code diff shows changes in the file 'googlemock/include/gmock/gmock-matchers.h'. The commit message is 'PiperOrigin-RevId: 398271948' and it was committed by 'dinord' 6 hours ago.

```
diff --git a/googlemock/include/gmock/gmock-matchers.h b/googlemock/include/gmock/gmock-matchers.h
@@ -2425,7 +2425,6 @@ class ContainerEqMatcher {
2425     typedef internal::StlContainerViews
2426     typename std::remove_const<LhsContainer>::type>
2427     LhsView;
2428 -    typedef typename LhsView::type LhsStlContainer;
2429     StlContainerReference lhs_stl_container = LhsView::ConstReference(lhs);
2430     if (lhs_stl_container == expected_)
2431         return true;
2432 -    @@ -2434,8 +2433,7 @@ class ContainerEqMatcher {
2433         if (os != nullptr) {
2434             // Something is different. Check for extra values first.
2435             bool printed_header = false;
2436             for (typename LhsStlContainer::const_iterator it =
2437                 lhs_stl_container.begin(),
2438                 it != lhs_stl_container.end(); ++it) {
2439                 if (!printed_header) {
```

GoogleTest

- Write the following text within your CMakeLists.txt file
 - See the next page for a larger picture
- Remember to change the URL's hash to the latest hash you've fetched



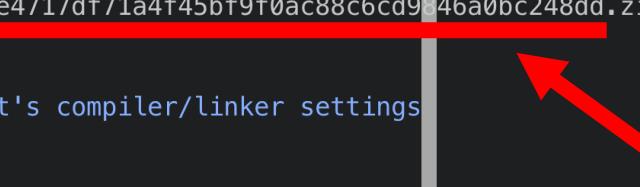
A screenshot of a terminal window showing a CMakeLists.txt file being edited in Vim. The code in the buffer is:

```
cmake_minimum_required(VERSION 3.14)
project(google_unittest)
# GoogleTest requires at least C++11
set(CMAKE_CXX_STANDARD 17)
include(FetchContent)
FetchContent_Declare(
    googletest
    URL https://github.com/google/googletest/archive/e4717df71a4f45bf9f0ac88c6cd946a0bc248dd.zip
)
# For Windows: Prevent overriding the parent project's compiler/linker settings
set(gtest_force_shared_crt ON CACHE BOOL "" FORCE)
FetchContent_MakeAvailable(googletest)
# GoogleTest stuff
enable_testing()
add_executable(
    basic_test
    basic_test.cc
)
target_link_libraries(
    basic_test
    gtest_main
)
include(GoogleTest)
gtest_discover_tests(basic_test)
```

The terminal status bar shows:

```
NORMAL > CMakeLists.txt
"CMakelists.txt" 34L, 648B
googletest + 332d 22h 22m 1 vi 2 ~
cmake < utf-8[unix] < 2% ↵:1/34☰ ↵:1
< 14:42 < 24 Sep ↵ jaechan ↵ multicore-96
```

```
1 cmake_minimum_required(VERSION 3.14)
2 project(google_unittest)
3 # GoogleTest requires at least C++11
4 set(CMAKE_CXX_STANDARD 17)
5
6 include(FetchContent)
7
8 FetchContent_Declare(
9   gtest
10  URL https://github.com/google/googletest/archive/e4717df71a4f45bf9f0ac88c6cd9846a0bc248dd.zip
11 )
12
13 # For Windows: Prevent overriding the parent project's compiler/linker settings
14 set(gtest_force_shared_crt ON CACHE BOOL "" FORCE)
15
16 FetchContent_MakeAvailable(gtest)
17
18
19 # GoogleTest stuff
20 enable_testing()
21
22 add_executable(
23   basic_test
24   basic_test.cc
25 )
26
27 target_link_libraries(
28   basic_test
29   gtest_main
30 )
31
32 include(Googletest)
33 gtest_discover_tests(basic_test)
```



Put your hash here

NORMAL >> CMakeLists.txt

"CMakeLists.txt" 34L, 648B

22 googletest >> ↑ 332d 22h 22m >> 1 vi >> 2 ~

cmake < utf-8[unix] < 2% ↵:1/34≡ ↵:1 ↵

GoogleTest



```
basic_test.cc
1 #include <gtest/gtest.h>
1 #include <cmath>
2 #include <vector>
3
4 int Factorial(int n) {
5     if (n == 0) {
6         return 1;
7     }
8
9     int result = 1;
10    for (int i = 2; i <= n; i++) {
11        result *= i;
12    }
13    return result;
14 }
15
16 bool IsPrime(int n) {
17    if (n < 2) {
18        return false;
19    }
20
21    for (int i = 2; i <= sqrt(n); i++) {
22        if (n % i == 0) {
23            return false;
24        }
25    }
26    return true;
27 }
28
29
```

NORMAL ➤ basic_test.cc cp

- Open `basic_test.cc` and create the following two functions, *Factorial* and *IsPrime*
- Make sure that you include '***gtest/gtest.h***'

GoogleTest

```
30 // Tests factorial of 0
1 TEST(FactorialTest, HandlesZeroInput) {
2     // Tests if the value of Factorial(0) is EQ(equal) to 1
3     EXPECT_EQ(Factorial(0), 1);
4 }
5
6 // Tests factorial of positive numbers
7 TEST(FactorialTest, HandlesPositiveInput) {
8     EXPECT_EQ(Factorial(1), 1);
9     EXPECT_EQ(Factorial(2), 2);
10    EXPECT_EQ(Factorial(3), 6);
11    EXPECT_EQ(Factorial(8), 40320);
12 }
```

```
44 // Tests if 1 is a prime or not
1 TEST(PrimeTest, HandlesOneInput) {
2     // 1 is not a prime
3     EXPECT_FALSE(IsPrime(1));
4 }
5
6 // Tests if a given positive number is a prime or not
7 TEST(PrimeTest, HandlesPositiveInput) {
8     EXPECT_TRUE(IsPrime(2));
9     EXPECT_TRUE(IsPrime(3));
10    EXPECT_FALSE(IsPrime(4));
11    EXPECT_TRUE(IsPrime(5));
12 }
```

- Create some basic tests.

- The format is:

TEST(TestSuiteName, TestName) {

... test body ...

}

- You can find the basic macros at
(e.g., EXPECT_EQ, EXPECT_TRUE)

<https://google.github.io/googletest/reference/assertions.html>

GoogleTest

```
q ➜ ~/c/googletest ➔ cmake -S . -B build
-- The C compiler identification is GNU 10.3.0
-- The CXX compiler identification is GNU 10.3.0
-- 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 Python: /usr/bin/python3.6 (found version "3.6.9")
  found components: Interpreter
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Configuring done
-- Generating done
-- Build files have been written to: /home/jaechan/code/googletest/build
```

- Run the following command in the ‘googletest’ directory
- *cmake -S . -B build*
- If you don’t have *cmake*, install it.

GoogleTest

```
q ➜ ~/c/googletest ➔ ls
basic_test.cc  build  CMakeLists.txt  memo.txt

q ➜ ~/c/googletest ➔ ls build
'basic_test[1]_include.cmake'  CTestTestfile.cmake
bin                      _deps
CMakeCache.txt            lib
CMakeFiles                  Makefile
cmake_install.cmake

q ➜ ~/c/googletest ➔
```

- On success, you will have the directory ‘build’ within your googletest directory.

GoogleTest

```
q ➜ ~/c/googletest ➔ cmake --build build
[ 10%] Building CXX object _deps/googletest-build/googletes
t/CMakeFiles/gtest.dir/src/gtest-all.cc.o
[ 20%] Linking CXX static library ../../lib/libgtest.a
[ 20%] Built target gtest
[ 30%] Building CXX object _deps/googletest-build/googletes
t/CMakeFiles/gtest_main.dir/src/gtest_main.cc.o
[ 40%] Linking CXX static library ../../lib/libgtest_ma
n.a
[ 40%] Built target gtest_main
[ 50%] Building CXX object CMakeFiles/basic_test.dir/basic_
test.cc.o
[ 60%] Linking CXX executable basic_test
[ 60%] Built target basic_test
[ 70%] Building CXX object _deps/googletest-build/googlemoc
k/CMakeFiles/gmock.dir/src/gmock-all.cc.o
[ 80%] Linking CXX static library ../../lib/libgmock.a
[ 80%] Built target gmock
[ 90%] Building CXX object _deps/googletest-build/googlemoc
k/CMakeFiles/gmock_main.dir/src/gmock_main.cc.o
[100%] Linking CXX static library ../../lib/libgmock_ma
n.a
[100%] Built target gmock_main
q ➜ ~/c/googletest ➔
```

- Build your test cases with the following command
- `cmake --build build`

GoogleTest

```
q ➤ ~ / c / googletest ➤ cd build
q ➤ ~ / c / g / build ➤ ctest
Test project /home/jaechan/code/googletest/build
  Start 1: FactorialTest.HandlesZeroInput
1/4 Test #1: FactorialTest.HandlesZeroInput ..... Passed
d 0.00 sec
  Start 2: FactorialTest.HandlesPositiveInput
2/4 Test #2: FactorialTest.HandlesPositiveInput ... Passed
d 0.01 sec
  Start 3: PrimeTest.HandlesOneInput
3/4 Test #3: PrimeTest.HandlesOneInput ..... Passed
d 0.00 sec
  Start 4: PrimeTest.HandlesPositiveInput
4/4 Test #4: PrimeTest.HandlesPositiveInput ..... Passed
d 0.00 sec

100% tests passed, 0 tests failed out of 4

Total Test time (real) = 0.02 sec
q ➤ ~ / c / g / build ➤
```

- Go to the ‘build’ directory
- Execute the command **ctest**
- You’ll see that all your basic tests has passed.
- From here on, add more tests and try out something new by yourself.
- Refer to the document for more information.

Thank you
