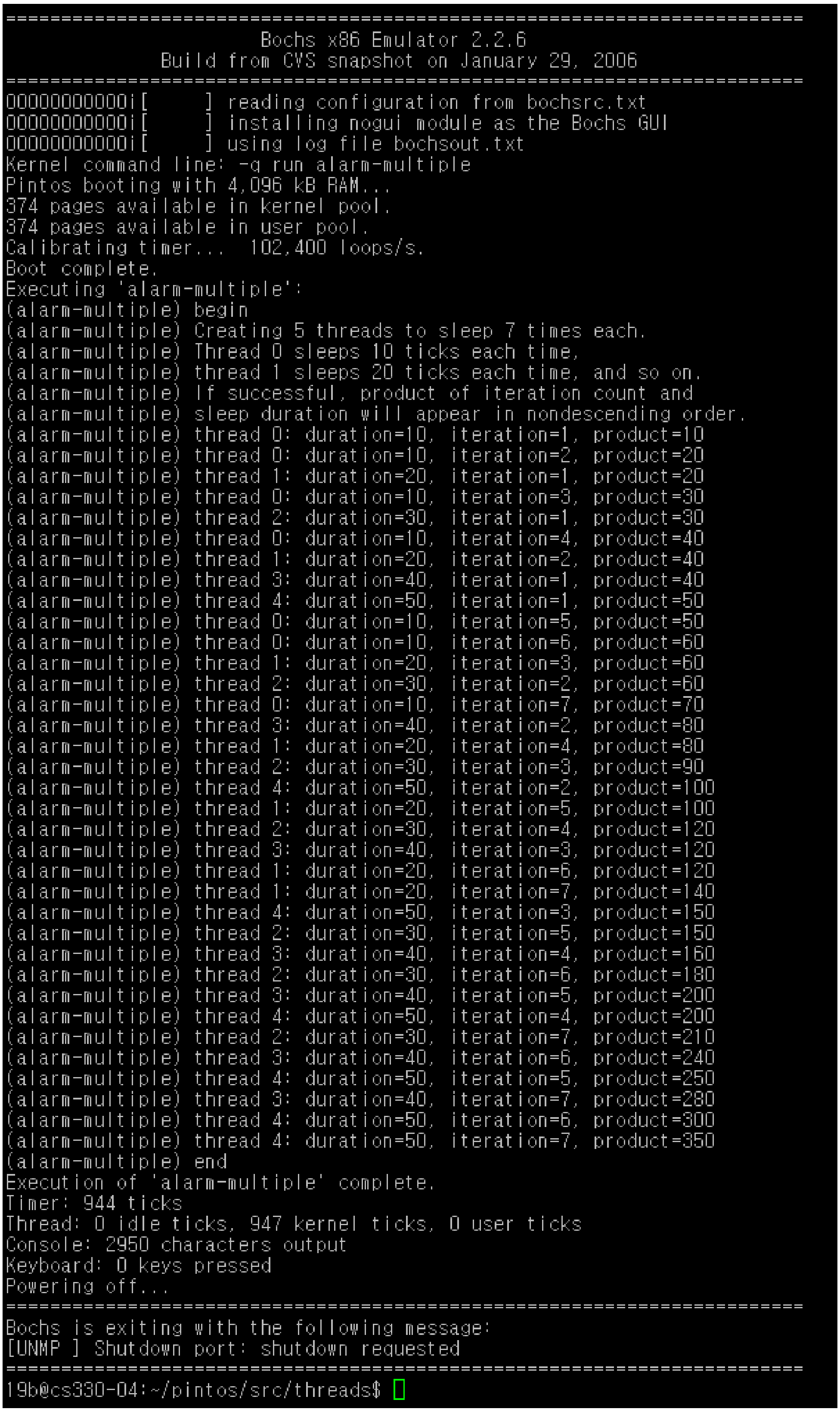
CS330 Project #0

Team 19: 20150031 강현우, 20150717 조영걸

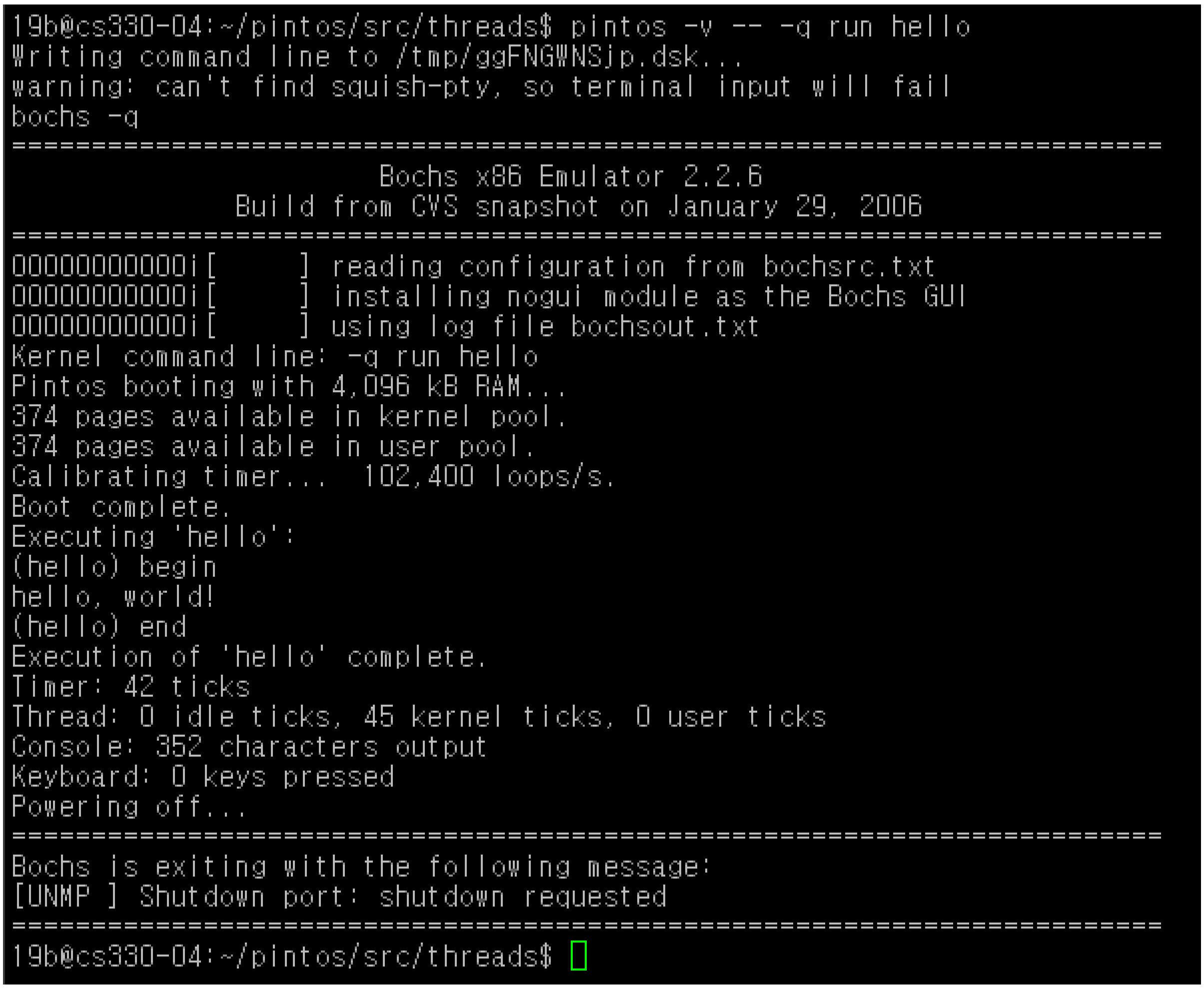
1. Testing the installation (alarm-multiple)



```
========================================================================
                       Bochs x86 Emulator 2.2.6
              Build from CVS snapshot on January 29, 2006
========================================================================
00000000000i[     ] reading configuration from bochsrc.txt
00000000000i[     ] installing nogui module as the Bochs GUI
00000000000i[     ] using log file bochsout.txt
Kernel command line: -q run alarm-multiple
Pintos booting with 4,096 kB RAM...
374 pages available in kernel pool.
374 pages available in user pool.
Calibrating timer...  102,400 loops/s.
Boot complete.
Executing 'alarm-multiple':
(alarm-multiple) begin
(alarm-multiple) Creating 5 threads to sleep 7 times each.
(alarm-multiple) Thread 0 sleeps 10 ticks each time,
(alarm-multiple) thread 1 sleeps 20 ticks each time, and so on.
(alarm-multiple) If successful, product of iteration count and
(alarm-multiple) sleep duration will appear in nondescending order.
(alarm-multiple) thread 0: duration=10, iteration=1, product=10
(alarm-multiple) thread 0: duration=10, iteration=2, product=20
(alarm-multiple) thread 1: duration=20, iteration=1, product=20
(alarm-multiple) thread 0: duration=10, iteration=3, product=30
(alarm-multiple) thread 2: duration=30, iteration=1, product=30
(alarm-multiple) thread 0: duration=10, iteration=4, product=40
(alarm-multiple) thread 1: duration=20, iteration=2, product=40
(alarm-multiple) thread 3: duration=40, iteration=1, product=40
(alarm-multiple) thread 4: duration=50, iteration=1, product=50
(alarm-multiple) thread 0: duration=10, iteration=5, product=50
(alarm-multiple) thread 0: duration=10, iteration=6, product=60
(alarm-multiple) thread 1: duration=20, iteration=3, product=60
(alarm-multiple) thread 2: duration=30, iteration=2, product=60
(alarm-multiple) thread 0: duration=10, iteration=7, product=70
(alarm-multiple) thread 3: duration=40, iteration=2, product=80
(alarm-multiple) thread 1: duration=20, iteration=4, product=80
(alarm-multiple) thread 2: duration=30, iteration=3, product=90
(alarm-multiple) thread 4: duration=50, iteration=2, product=100
(alarm-multiple) thread 1: duration=20, iteration=5, product=100
(alarm-multiple) thread 2: duration=30, iteration=4, product=120
(alarm-multiple) thread 3: duration=40, iteration=3, product=120
(alarm-multiple) thread 1: duration=20, iteration=6, product=120
(alarm-multiple) thread 1: duration=20, iteration=7, product=140
(alarm-multiple) thread 4: duration=50, iteration=3, product=150
(alarm-multiple) thread 2: duration=30, iteration=5, product=150
(alarm-multiple) thread 3: duration=40, iteration=4, product=160
(alarm-multiple) thread 2: duration=30, iteration=6, product=180
(alarm-multiple) thread 3: duration=40, iteration=5, product=200
(alarm-multiple) thread 4: duration=50, iteration=4, product=200
(alarm-multiple) thread 2: duration=30, iteration=7, product=210
(alarm-multiple) thread 3: duration=40, iteration=6, product=240
(alarm-multiple) thread 4: duration=50, iteration=5, product=250
(alarm-multiple) thread 3: duration=40, iteration=7, product=280
(alarm-multiple) thread 4: duration=50, iteration=6, product=300
(alarm-multiple) thread 4: duration=50, iteration=7, product=350
(alarm-multiple) end
Execution of 'alarm-multiple' complete.
Timer: 944 ticks
Thread: 0 idle ticks, 947 kernel ticks, 0 user ticks
Console: 2950 characters output
Keyboard: 0 keys pressed
Powering off...
========================================================================
Bochs is exiting with the following message:
[UNMP ] Shutdown port: shutdown requested
========================================================================
19b@cs330-04:~/pintos/src/threads$
```

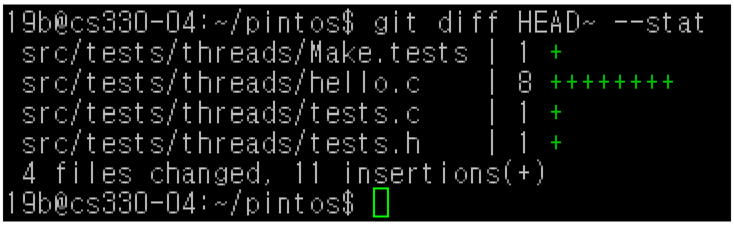
2. Test program (hello)

- screenshot of execution



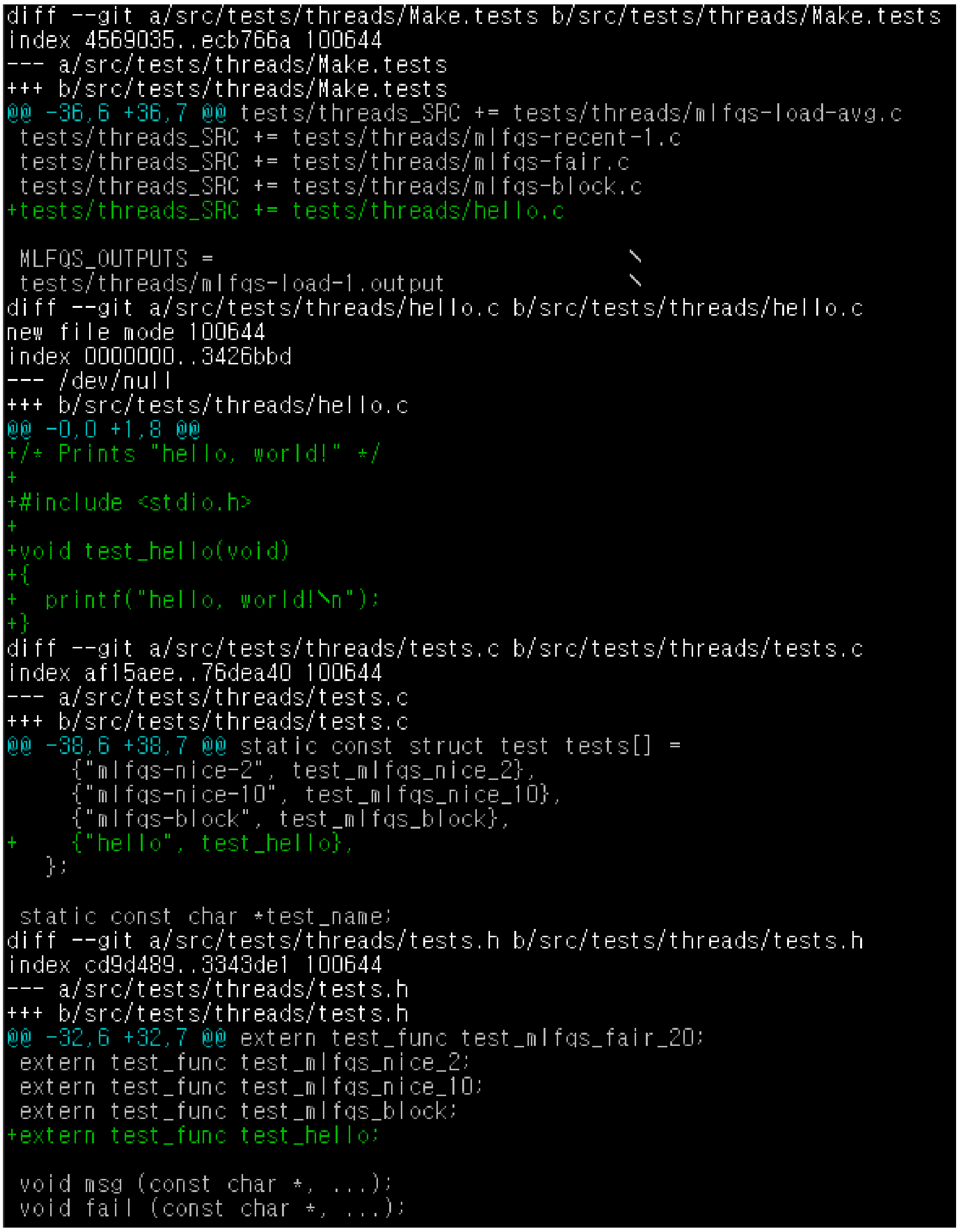
```
19b@cs330-04:~/pintos/src/threads$ pintos -v -- -q run hello
Writing command line to /tmp/ggFNGWNSjp.dsk...
warning: can't find squish-pty, so terminal input will fail
bochs -q
========================================================================
                       Bochs x86 Emulator 2.2.6
              Build from CVS snapshot on January 29, 2006
========================================================================
00000000000i[     ] reading configuration from bochsrc.txt
00000000000i[     ] installing nogui module as the Bochs GUI
00000000000i[     ] using log file bochsout.txt
Kernel command line: -q run hello
Pintos booting with 4,096 kB RAM...
374 pages available in kernel pool.
374 pages available in user pool.
Calibrating timer...  102,400 loops/s.
Boot complete.
Executing 'hello':
(hello) begin
hello, world!
(hello) end
Execution of 'hello' complete.
Timer: 42 ticks
Thread: 0 idle ticks, 45 kernel ticks, 0 user ticks
Console: 352 characters output
Keyboard: 0 keys pressed
Powering off...
========================================================================
Bochs is exiting with the following message:
[UNMP ] Shutdown port: shutdown requested
========================================================================
19b@cs330-04:~/pintos/src/threads$
```

- diff --stat



```
19b@cs330-04:~/pintos$ git diff HEAD~ --stat
 src/tests/threads/Make.tests | 1 +
 src/tests/threads/hello.c    | 8 ++++++++
 src/tests/threads/tests.c    | 1 +
 src/tests/threads/tests.h    | 1 +
 4 files changed, 11 insertions(+)
19b@cs330-04:~/pintos$
```

- what we modified (git diff)



```
diff --git a/src/tests/threads/Make.tests b/src/tests/threads/Make.tests
index 4569035..ecb766a 100644
--- a/src/tests/threads/Make.tests
+++ b/src/tests/threads/Make.tests
@@ -36,6 +36,7 @@ tests/threads_SRC += tests/threads/mlfqs-load-avg.c
 tests/threads_SRC += tests/threads/mlfqs-recent-1.c
 tests/threads_SRC += tests/threads/mlfqs-fair.c
 tests/threads_SRC += tests/threads/mlfqs-block.c
+tests/threads_SRC += tests/threads/hello.c

 MLFQS_OUTPUTS =                                 \
 tests/threads/mlfqs-load-1.output               \
diff --git a/src/tests/threads/hello.c b/src/tests/threads/hello.c
new file mode 100644
index 0000000..3426bbd
--- /dev/null
+++ b/src/tests/threads/hello.c
@@ -0,0 +1,8 @@
+/* Prints "hello, world!" */
+
+#include <stdio.h>
+
+void test_hello(void)
+{
+  printf("hello, world!\n");
+}
diff --git a/src/tests/threads/tests.c b/src/tests/threads/tests.c
index af15aee..76dea40 100644
--- a/src/tests/threads/tests.c
+++ b/src/tests/threads/tests.c
@@ -38,6 +38,7 @@ static const struct test tests[] =
     {"mlfqs-nice-2", test_mlfqs_nice_2},
     {"mlfqs-nice-10", test_mlfqs_nice_10},
     {"mlfqs-block", test_mlfqs_block},
+    {"hello", test_hello},
   };

 static const char *test_name;
diff --git a/src/tests/threads/tests.h b/src/tests/threads/tests.h
index cd9d489..3343de1 100644
--- a/src/tests/threads/tests.h
+++ b/src/tests/threads/tests.h
@@ -32,6 +32,7 @@ extern test_func test_mlfqs_fair_20;
 extern test_func test_mlfqs_nice_2;
 extern test_func test_mlfqs_nice_10;
 extern test_func test_mlfqs_block;
+extern test_func test_hello;

 void msg (const char *, ...);
 void fail (const char *, ...);
```

- how it works

The main function in <src/threads/init.c> executes the test when there is a parameter 'run'. First it calls run_test function in <src/tests/threads/tests.c>, and it finds the corresponding test function name from the table, **`static const struct test tests[]`**. So we added our hello function, test_hello, into the table. And we also added the function to tests.h with extern keyword.

And then we also added our hello test into <src/tests/threads/Make.tests>, so that it could be built and able to run.

Therefore, the test hello could be executed appropriately when we execute the test with the command `pintos -v -- -q run hello` .