[Project 0] Report

20150138 김바울

1 INSTALLATION

#pintos -v -- -q run alarm-multiple 을 실행한 결과:

```
cs20150138@cs330-3:~/pintos/src/threads$ pintos -v -- -q run alarm-multiple
Writing command line to /tmp/nglk1vNHXs.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 alarm-multiple
Pintos booting with 4,096 kB RAM...
375 pages available in kernel pool.
374 pages available in user pool.
Calibrating timer...  204,600 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 0: duration=10, iteration=5, product=50
```

```
(alarm-multiple) thread 4: duration=50, iteration=1, product=50
(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=6, 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 2: duration=30, iteration=5, product=150
(alarm-multiple) thread 4: duration=50, iteration=3, 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 4: duration=50, iteration=4, product=200
(alarm-multiple) thread 3: duration=40, iteration=5, 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: 923 ticks
Thread: 0 idle ticks, 926 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
========================================================================
```

Bochs x86 Emulator 2.2.6 이 구동이 되면서,
alarm-multiple 이라는 테스트가 실행이 되는 것을 확인할 수 있었다.

2 HELLO PROGRAM

#pintos -v -- -q run hello 를 실행한 결과:

```
cs20150138@cs330-3:~/pintos/src/threads$ pintos -v -- -q run hello
Writing command line to /tmp/jVvzBxmsR7.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...
375 pages available in kernel pool.
374 pages available in user pool.
Calibrating timer...  204,600 loops/s.
Boot complete.
Executing 'hello':
(hello) begin
hello, world!
(hello) end
Execution of 'hello' complete.
Timer: 32 ticks
Thread: 0 idle ticks, 34 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
========================================================================
```

```
Executing 'hello':
(hello) begin
hello, world!
(hello) end
Execution of 'hello' complete
```

위의 출력을 통해 “hello, world!”라는 문자열을 출력하는 테스트 hello 가 실행됨을 볼 수 있다.

2.1 How Test works

run 명령어는 pintos/src/tests/threads/ 에 있는 test 들을 실행한다. 이 테스트들은 같은 디렉토리 안의 test.c 에 정의되어있으며, 테스트 함수들은 test.h 에 설정되어있다. 그러므로 이 파일을 수정해야 한다.

test.c (일부분)

```
static const struct test tests[] =
  {
    {"hello", test_hello},
    {"alarm-single", test_alarm_single},
    ....
```

test.h (일부분)

```
typedef void test_func (void);

extern test_func test_hello;
extern test_func test_alarm_single;
....
```

hello 라는 이름의 테스트를 만들고, 그에 해당하는 테스트 함수를 test_hello 라고 정의해두었다. 이 함수는 hello.c 라는 같은 디렉토리의 새로운 파일에 정의해둔다.

hello.c

```
#include <stdio.h>
#include "tests/threads/tests.h"
#include "threads/init.h"
#include "threads/malloc.h"
#include "threads/synch.h"
#include "threads/thread.h"
#include "devices/timer.h"

void test_hello (void)
{
  printf("hello, world!\n");
}
```

이 새로 생성한 파일은 makefile 이 생성할 때 포함되어있지 않으므로, 같은 디렉토리에 있는 Make.tests 파일을 수정한다.

Make.test (일부분)

```
# Sources for tests.
tests/threads_SRC  = tests/threads/tests.c
tests/threads_SRC += tests/threads/hello.c
```

이렇게 함으로써 make 할 때 hello.c 가 컴파일 되고, test_hello 함수가 hello 라는 이름을 가진 테스트로 등록이 된다.

`run hello` 명령은 test_hello 를 실행하여 “hello, world!”를 출력하게 한다.