Brief Introduction to kselftest

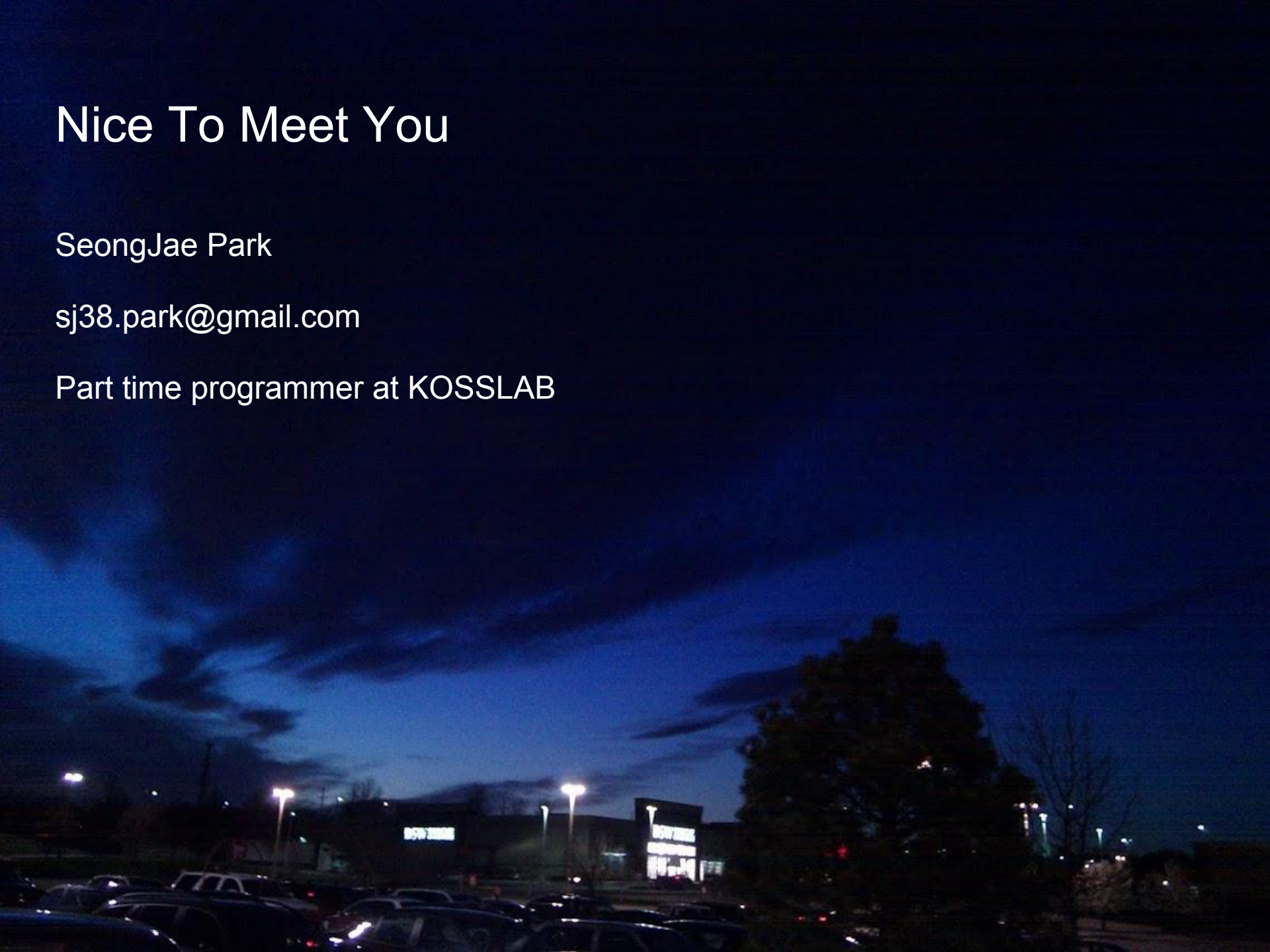
SeongJae Park <sj38.park@gmail.com>

Nice To Meet You

SeongJae Park

sj38.park@gmail.com

Part time programmer at KOSSLAB



``Go out and test'', but, how?

- Linus Torvalds always want us to test the kernel, but, how?

```
Date: Mon, 19 Jun 2017 23:04:15 +0800
From: Linus Torvalds <torvalds@linux-foundation.org>
To: Linux Kernel Mailing List <linux-kernel@vger.kernel.org>
Subject: Linux 4.12-rc6
...
The good news is that rc6 is smaller than rc5 was, and I think we're
...
Go out and test,

                    Linus
```

The Way to Test Kernel

- Simplest way: Build, boot, and use the kernel
  - It covers many important test cases, but limited and not funny (We do this *just for fun*, right?)

The Ways to Test Linux Kernel Automatically

- Many automatic test suites, framework, and services for the kernel exists
  - Linux Test Project (https://linux-test-project.github.io/)
  - Zero-day service (https://01.org/lkp/documentation/0-day-test-service)
  - Kernel-ci.org (https://kernelci.org/)
  - mmtests (https://github.com/gormanm/mmtests)
- However, fetching, configuring, running, and waiting them could be a little difficult to some developers
- Need more developer-friendly tests for more test

kselftest: Kernel Self Test

- Set of developer-friendly test framework and tests for the kernel
- Goal is to help developers do test more easily and more frequently
- May not cover entire case,
  but incomplete test is much better than just praying
- It is contained in the kernel source code at `/tools/testing/selftest/.`
  If you have the source code, you have the test
- Runs quickly: under 20 minutes
- Discussed from 2014 kernel summit (https://lwn.net/Articles/608959/)
- Maintained by Shuah Kahn <shuahkh@osg.samsung.com>

Kselftest is Contained in The Kernel Source Code

- You have the test if you have the kernel source code
  - Under `tools/testing/selftests`
- For newest selftest, use the git repository at https://git.kernel.org/pub/scm/linux/kernel/git/shuah/linux-kselftest.git/
- Select `master`, `next`, `devel`, or `fixes` branch for your purpose
- `next`: Content for upcoming merge window
- `fixes`: Fixes to current -rc release are contained
- `devel`: Experimental patches are contained

Kselftest Runs Quickly

- Many tests require long running time
  - Mel Gorman's test takes thirteen days
  - Paul McKenney's rcutorture takes six hours (https://lwn.net/Articles/608959/)
- Long running tests will not be accepted as default test
- Current goal is under 20 minutes
- Kselftest default run takes about 7 minutes on my PC

Build and Run

- First, boot with the kernel
- Build
  - # make -C tools/testing/selftests
- Run
  - # make -C tools/testing/selftests run_tests
  - The command does build if not built yet
- Simplest way to build and run:
  - # make kselftest
- Some tests require root privilege, but running them without the privilege should not make serious problem

Running Subset of Tests

- If you have interest in specific tests only, you can build and run them only
  - $ make TARGETS=<name of tests> kselftest

Install kselftest

- The tests can be installed in anywhere (e.g., other system that connected with nfs) you want
  - $ cd tools/testing/selftests; ./kselftest_install.sh [install location]
- To run the installed tests
  - $ cd [install localtion/]kselftest; ./run_kselftest.sh

Packaging kselftest

- The tests can be packaged as tarball
  - $ ./gen_kselftest_tar.sh [tar|targz|tarbz2|tarxz]
- The package can be installed and run at anywhere (e.g., tiny testing machine without gcc or make)
  - $ tar xf ./kselftest.tar.gz; cd kselftest; ./run_kselftest.sh

Adding Your New Test

- Insert your test code and `Makefile` for the test under selftests/<test name>/
- The `Makefile` should include `selftests/lib.mk` and define your test program(s) as `TEST_PROGS`
- Add your `<test name>` to `TARGETS` in selftests/Makefile

```
selftests/sjpark$ cat Makefile
CFLAGS = -Wall -O2 -g

TEST_PROGS := sjtest
TEST_GEN_FILES = sjtest

include ../lib.mk

selftests/sjpark$ cat sjtest.c
#include <stdio.h>

int main(void)
{
        printf("hello world\n");
        return 0;
}
```

Kselftest can be a Start Point to the Kernel

- In many case, test code is a good document to the program
  - If you can judge a behavior of an instance as success or fail,
    you are understanding the function
- If you are new to the kernel, kselftest can be a good start point

Conclusion

- Kselftest is a developer-friendly test framework and tests for the Linux kernel
- The test is contained in the kernel source code itself
- The test runs quickly under 20 minutes
- The test can be packaged, installed, and run easily
- The test code can be a good start point to the Linux kernel

Thank You