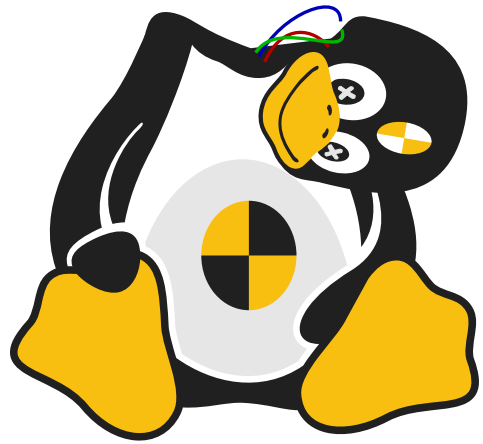
Linux Test Project introduction

"Breaking penguins since 2000"

Cyril Hrubis

SUSE Linux

4. February 2018



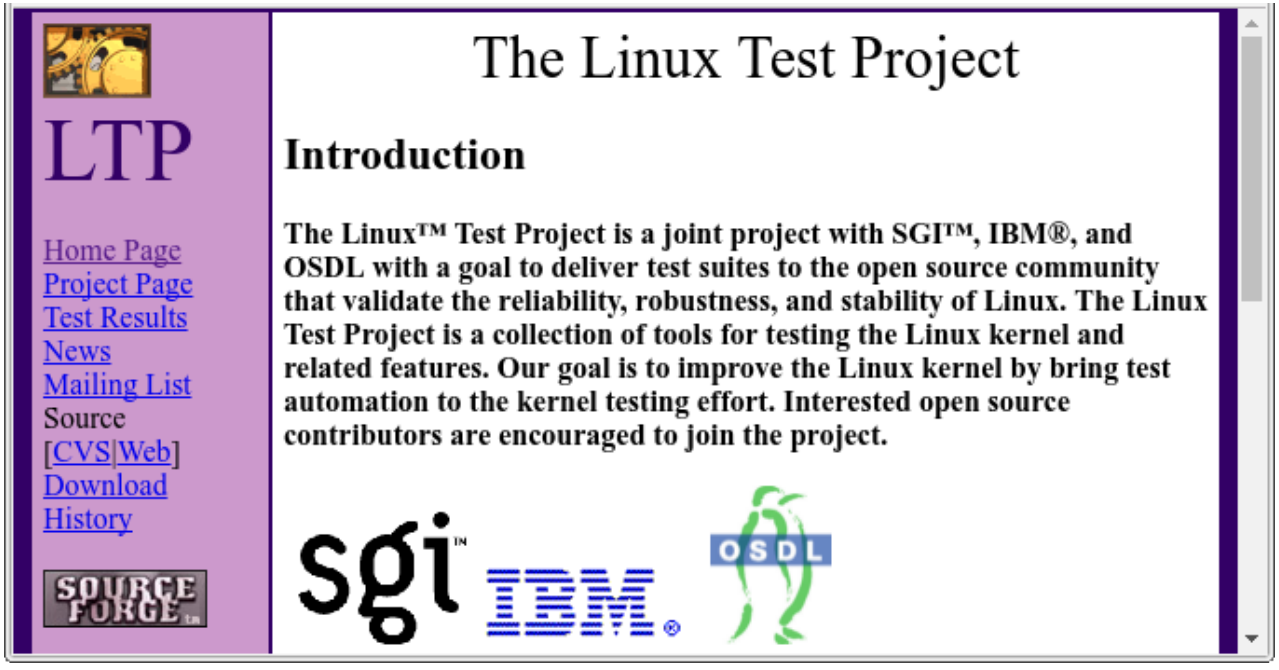
SUSE

About myself

Cyril Hrubis
(aka metan on freenode)

- Linux user and C programmer since 2000
- SUSE employee since 2007
- Kernel Automation QA since 2008
- LTP upstream developer since 2009
- ...

LTP Origin



- The sourceforge project was registered around 2000.
- In 2001 it contained about 100 simple syscalls tests and a few testsuites collected from other sources.

Historicall problems

- There was very little or no code review.
- Build was often failing for less common configurations.
- No build system, just bunch of random Makefiles.
- There was little or no documentation.

Historicall problems

- Fair amount of the testcases was failing randomly.
- LTP was put together from pieces of testsuites some of them dating back to the days of UNIX wars.
- Third party testsuites were poorly integrated if at all. “Sometimes stiches are still visible”
- IBM hired, in good faith, junior developers to work on syscall tests.

Current state “boring”

- LTP adopted LKML coding style.
- The git repository is hosted on GitHub.
- Development process centers around patch review on the mailing list.
- Quaterly releases with aprox. 260 patches and 33 authors per release.

Current state “boring”

- Travis is used for compile testing.
- We make sure that latest LTP works fine on currently maintained distros.
- Comprehensive test library that greatly simplifies writing testcases.
- We have API documentation and tutorial on wiki.

LTP Goals

The goal of the project was and is:
"Validate the reliability, robustness, and stability of Linux."

- LTP focuses on functionality, regression and stress testing for the Linux kernel and related features.
- LTP does not include benchmarking, there are MMTests from Mel Gorman covering that.
- For filesystems testing it's better to be combined with xfstests.

LTP Challenges

- LTP project goal is a bit too broad.
- It's difficult to even estimate how much kernel-userspace API does exists.
- LTP is large, roughly 4000 C sources and 500 scripts.
- Mostly contains complicated low level code.
- Sometimes documentation for Kernel API/ABI is missing, wrong or misleading.
- Kernel API/ABI cannot be changed, unless it can (cgroups). "WE DO NOT BREAK USERSPACE!"

LTP Content

LTP contains:

- ~1200 syscall testcases
- ~1600 POSIX conformance tests
- Regression tests for Linux CVEs (dirtyc0w, stack_clash, meltdown, ...)
- Various I/O stress tests
- Network related tests
- Realtime testsuite
- Linux container, controller, and namespace tests
- ...

LTP Test Design Goals

- Languages of choice are C and portable shell.
- Each test is an executable.
- Each test is as self-contained as possible.
- Each test runs automatically.
- Overall test status is passed as an exit value.
- Additional information is printed to stdout.
- Global parameters are passed via environment variables.

FAQ: To test or not to test?

"The upstream kernel is thoroughly tested so there is not point in testing it in-house, right?"

Turns out that this only applies if you haven't applied any patches on the top of the upstream kernel.

FAQ: How to run LTP test(s)?

We have to compile LTP from released tarball or git first.

(We have mini howto for compiling LTP in `doc/` directory.)

But basically it should be as easy as:

- `git clone https://github.com/.../ltp.git`
- `cd ltp && make autotools`

or

- `wget https://.../ltp-full-20180118.tar.bz2`
- `tar xf ltp-* && cd ltp-*`

then

- `configure`
- `make -j$(getconf _NPROCESSORS_ONLN)`

FAQ: How to run LTP test?

Most of the testcases can be executed from the source tree:

- `cd testcases/kernel/syscalls/fcntl`
- `PATH=$PATH:$PWD ./fcntl02`

```
tst_test.c:980: INFO: Timeout per run is 0h 05m 00s
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 0) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 1) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 2) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 3) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 10) returned 10
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 100) returned 100

Summary:
passed   6
failed   0
skipped  0
warnings 0
```

FAQ: How to run LTP test?

Alternatively LTP can be installed.

- `cd ltp && make install && cd /opt/ltp`
- `./runltp -f syscalls -s fcntl02`

```
...
<<<test_output>>>
tst_test.c:980: INFO: Timeout per run is 0h 05m 00s
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 0) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 1) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 2) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 3) returned 4
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 10) returned 10
fcntl02.c:70: PASS: fcntl(fcntl02_13303, F_DUPFD, 100) returned 100

Summary:
passed   6
failed   0
skipped  0
warnings 0
<<<execution_status>>>
...
```

FAQ: How to run LTP network test?

Network test usually needs two machines with LTP installed but then can also fall back to netork namespaces.

- `/opt/ltp/testscripts/network.sh -6`

```
network_settings 1 TINFO: initialize 'lhost' 'ltp_ns_veth2' interface
network_settings 1 TINFO: set local addr 10.0.0.2/24
network_settings 1 TINFO: set local addr fd00:1:1:1::2/64
network_settings 1 TINFO: initialize 'rhost' 'ltp_ns_veth1' interface
network_settings 1 TINFO: set remote addr 10.0.0.1/24
network_settings 1 TINFO: set remote addr fd00:1:1:1::1/64
network_settings 1 TINFO: wait for IPv6 DAD completion 1/5 sec
network_settings 1 TINFO: Network config (local -- remote):
network_settings 1 TINFO: ltp_ns_veth2 -- ltp_ns_veth1
network_settings 1 TINFO: 10.0.0.2/24 -- 10.0.0.1/24
network_settings 1 TINFO: fd00:1:1:1::2/64 -- fd00:1:1:1::1/64
<<<test_start>>>
tag=ping601 stime=1517391093
cmdline="ping01.sh -6"
contacts=""
analysis=exit
<<<test_output>>>
ping01 1 TINFO: ping6 with 8 16 32 64 128 256 512 1024 2048 4064 ICMP
ping01 1 TPASS: ping6 -c 3 -s 8 fd00:1:1:1::1 >/dev/null passed as exp
...
```

Contact

- GIT repository:
  `https://github.com/linux-test-project/ltp`
- Mailing list:
  `https://lists.linux.it/listinfo/ltp`
- Wiki:
  `https://github.com/linux-test-project/ltp/wiki`
- IRC:
  `#ltp on freenode.net`