Essentials: LIT testing

LLVM Social Berlin #18, 2019-08-28

Motivation

Want functionality from LLVM? Add a LIT test for it! -- Why?

➔ Brutal pace of change breaks features all the time.

  The test suite tends to “define” what functionality exists.

➔ “Breaks test XY” is a valid reason:

  - ◆ to request changes in a review
  - ◆ to revert a change post-review

➔ All the good reasons for pragmatic test-driven development.

Build and test LLVM

```
$ git clone https://github.com/llvm/llvm-project
$ cmake -G Ninja -B llvm-build -S llvm-project/llvm \
        -DLLVM_ENABLE_PROJECTS="clang;libcxx;lldb" \
        -DLLVM_TARGETS_TO_BUILD=host
$ ninja -C llvm-build check-llvm
[2134/2135] Running the LLVM regression tests
Testing Time: 324.21s
  Expected Passes    : 19116
  Expected Failures  : 57
  Unsupported Tests  : 14020
```

Build and test LLVM

```
$ ninja -t commands check-llvm | tail -1
cd /path/to/llvm-build/test && /usr/local/bin/python
↪ /path/to/llvm-build/./bin/llvm-lit -sv /path/to/llvm-build/test
```

Python script for executing LLVM style test suites, summarizing their results, and providing indication of failures.

Directory with a `lit.site.cfg.py`

Show test output for failures

Reduce amount of output

https://llvm.org/docs/CommandGuide/lit.html

Build and test Clang

```
$ ninja -t commands check-clang | tail -1
cd /path/to/llvm-build/tools/clang/test && /usr/local/bin/python
↪ /path/to/llvm-build/./bin/llvm-lit -sv
↪ --param clang_site_config=/path/to/llvm-build/tools/clang/test/lit.site.cfg
↪ --param USE_Z3_SOLVER=0 /path/to/llvm-build/tools/clang/test
$ ninja check-clang
[1123/1124] Running the Clang regression tests
Testing Time: 519.39s
  Expected Passes    : 14924
  Expected Failures  : 18
  Unsupported Tests  : 333
```

Build and test JitFromScratch

```
$ ninja -t commands check | tail -1
cd /path/to/jitfromscratch-build/test && /usr/local/bin/python3.7
↪️ /path/to/llvm-build/./bin/llvm-lit -v /path/to/jitfromscratch-build/test
$ ninja check
[5/6] Running tests
-- Testing: 3 tests, 3 threads --
PASS: JitFromScratch :: stderr.test-debug (1 of 3)
PASS: JitFromScratch :: stdout.test (2 of 3)
PASS: JitFromScratch :: objcache.test-debug (3 of 3)
Testing Time: 0.45s
  Expected Passes: 3
```

The JitFromScratch `check` target

```
configure_lit_site_cfg(
  ${CMAKE_CURRENT_SOURCE_DIR}/lit.site.cfg.py.in
  ${CMAKE_CURRENT_BINARY_DIR}/lit.site.cfg.py
  MAIN_CONFIG
  ${CMAKE_CURRENT_SOURCE_DIR}/lit.cfg.py
)
add_lit_target(check "Running tests"
  ${CMAKE_CURRENT_BINARY_DIR}
  DEPENDS JitFromScratch FileCheck
)
```

https://github.com/weliveindetail/JitFromScratch/commit/d84cc78

The JitFromScratch `lit.cfg.py`

```
config.name = 'JitFromScratch'

config.test_format = lit.formats.ShTest()
config.test_source_root = os.path.dirname(__file__)
config.suffixes = ['.test']

# Add binary directories for JitFromScratch and FileCheck executables
llvm_config.with_environment('PATH', config.jitfromscratch_build_dir,
                                     ↪ append_path=True)
llvm_config.with_environment('PATH', config.llvm_tools_dir, append_path=True)
```

https://github.com/weliveindetail/JitFromScratch/blob/llvm09/master/test/lit.site.cfg.py.in

The JitFromScratch `lit.site.cfg.py.in`

```
config.llvm_tools_dir = "@LLVM_TOOLS_DIR@"
config.lit_tools_dir = "@LLVM_LIT_TOOLS_DIR@"
config.python_executable = "@PYTHON_EXECUTABLE@"
config.jitfromscratch_build_dir = "@CMAKE_BINARY_DIR@/@CMAKE_CFG_INTDIR@"

import lit.llvm
lit.llvm.initialize(lit_config, config)

lit_config.load_config(config, "@CMAKE_SOURCE_DIR@/test/lit.cfg.py")
```

https://github.com/weliveindetail/JitFromScratch/blob/llvm09/master/test/lit.cfg.py

The JitFromScratch `stdout.test`

`RUN` lines determine commands to execute

Comment on the test case

```
# Check that integer distances are printed as expected
# RUN: JitFromScratch | FileCheck %s
# CHECK: Integer Distances: 3, 0, 3
```

LIT-specific placeholder for *path of this file*

`CHECK` lines tell FileCheck what output is expected

Executable found in `llvm_tools_dir`

Executable found in `jitfromscratch_build_dir`

https://github.com/weliveindetail/JitFromScratch/blob/llvm09/master/test/stdout.test

The JitFromScratch `objcache.test-debug`

```
# RUN: JitFromScratch -debug -debug-only=jitfromscratch 2>&1 |
     ↪ FileCheck -check-prefix=CACHELESS-CHECK %s
# CACHELESS-CHECK-NOT: Write cached object 'cache/JitFromScratch.o'
# CACHELESS-CHECK: Integer Distances: 3, 0, 3

# RUN: JitFromScratch -cache-dir=cache/ -debug -debug-only=jitfromscratch
     ↪ 2>&1 | FileCheck -check-prefix=WRITE-CHECK %s
# WRITE-CHECK: Create new cache directory 'cache/'
# WRITE-CHECK: Write cached object 'cache/JitFromScratch.o'
# WRITE-CHECK: Integer Distances: 3, 0, 3
```

https://github.com/weliveindetail/JitFromScratch/blob/llvm09/master/test/objcache.test-debug

A real-world test

```
# REQUIRES: target-x86_64
# XFAIL: system-windows
# RUN: %clang -g -S -emit-llvm --target=x86_64-unknown-unknown-elf
              ↪️ -o %t.ll %p/Inputs/jitbp.cpp
# RUN: %lldb -b -o 'settings set plugin.jit-loader.gdb.enable on'
                ↪️ -o 'b jitbp' -o 'run -jit-kind=mcjit %t.ll' lli | FileCheck
%s
# CHECK: Breakpoint 1: no locations (pending).
# CHECK: (lldb) run -jit-kind=mcjit {{.*}}/jitbp_elf.test.tmp.ll
# CHECK: Process {{.*}} stopped
# CHECK: JIT(0x{{.*}})`jitbp:
# CHECK: Process {{.*}} launched: {{.*}}
```

https://github.com/llvm/llvm-project/.../jitbp_elf.test

A real-world test in-tree

```
$ ninja -C llvm-build FileCheck llvm-config lli clang lldb
$ llvm-build/bin/llvm-lit -vv llvm-project/lldb/lit/Breakpoint/jitbp_elf.test
-- Testing: 1 tests, single process --
PASS: LLDB :: Breakpoint/jitbp_elf.test (1 of 1)
Testing Time: 9.85s
  Expected Passes: 1

1 warning(s) in tests.
```

https://reviews.llvm.org/D61611