How to build KLEE

This is a collection of our notes about the installation of KLEE. This document contains a step by step recipe for building KLEE and its dependencies.

Manual build step by step

Introduction

The resulting directory structure:

```
klee
   org-klee
   opt-klee
   klee-uclibc (linux)
   llvm-3.4
   minisat
   stp
   z3
```

Usefull Links:

- The official (but buggy) installation manual
- Build LLVM on your own
- The old official installation manual
- More recent user installation for Ubuntu 14.04 LTS
- STP installation manual with build options
- metaSMT-Support for KLEE

Fedora 25 (with stow)

Step 1: Install required tools for the build

```
Packages to install:
bison cmake curl flex git boost-devel perftools-devel ninja-build graphviz doxygen

Set the install path:
export KLEE_DIR=/usr/local/stow/klee
```

Step 2: LLVM

Source	URL	Directory
llvm	http://releases.llvm.org/3.4.2/llvm-3.4.2.src.tar.gz	llvm-3.4
clang	http://releases.llvm.org/3.4.2/cfe-3.4.2.src.tar.gz	llvm-3.4/tools/clang
compiler rt	http://releases.llvm.org/3.4/compiler-rt-3.4.src.tar.gz	llvm-3.4/projects/compiler-rt
test suite	http://releases.llvm.org/3.4/test-suite-3.4.src.tar.gz	llvm-3.4/projects/test-suite

```
cd llvm-3.4
mkdir build_release
cd build_release

cmake -G "Ninja" \
 -DCMAKE_BUILD_TYPE:STRING='Release' \
 -DCMAKE_INSTALL_PREFIX="${KLEE_DIR}" \
 -DLLVM_TARGETS_TO_BUILD:STRING='host' \
 ..

ninja
ninja install
cd ..\..
```

Step 3: Minisat

```
git clone https://github.com/stp/minisat.git
cd minisat
mkdir build_release
cd build_release
cmake -G "Ninja" \
 -DCMAKE_INSTALL_PREFIX=${KLEE_DIR} \
 -DCMAKE_BUILD_TYPE='Release' \
 ..

ninja
ninja install
cd ..\..
```

Step 4: STP

```
git clone https://github.com/stp/stp.git
cd stp
git checkout stp-2.2.0
mkdir build_release
cd build_release
cmake -G "Ninja" \
 -DENABLE_PYTHON_INTERFACE:BOOL=OFF \
 -DCMAKE_BUILD_TYPE="Release" \
 -DTUNE_NATIVE:BOOL=ON \
 -DCMAKE_INSTALL_PREFIX=${KLEE_DIR} \
 -DMINISAT_INCLUDE_DIR="${KLEE_DIR}/include" \
 -DMINISAT_LIBRARY="${KLEE_DIR}/lib/libminisat.so"  \
 ..

ninja
ninja install
cd ../..
```

Step 5: Z3

```
git clone https://github.com/Z3Prover/z3.git
```

```
cd z3
python2 scripts/mk_make.py - --prefix=${KLEE_DIR}
cd build
make -j `nproc`
make install
cd ../..
```

Installation places a new shared object library in /usr/local/lib. Need to run `sudo ldconfig` so the os can find it.

Step 6: uclibc and the POSIX environment model

```
git clone https://github.com/klee/klee-uclibc.git
cd klee-uclibc
./configure --make-llvm-lib --with-llvm-config=${KLEE_DIR}/bin/llvm-config
make -j `nproc`
cd ..
```

Step 7: org-klee

Original klee code for reference. Not required to build.

```
git clone https://github.com/klee/klee.git org-klee
```

Step 8: opt-klee

```
git clone https://github.gatech.edu/arktos/opt-klee.git
cd opt-klee
mkdir build_release
cd build_release

cmake -G "Ninja" \
 -DCMAKE_INSTALL_PREFIX=${KLEE_DIR} \
 -DCMAKE_BUILD_TYPE:STRING='Release' \
 -DCMAKE_EXPORT_COMPILE_COMMANDS=ON \
 -DCMAKE_CXX_FLAGS="-fno-rtti" \
 -DUSE_CXX11=ON \
 -DENABLE_TCMALLOC=ON \
 -DENABLE_SOLVER_STP=ON \
 -DENABLE_SOLVER_Z3=ON \
 -DENABLE_POSIX_RUNTIME=ON \
 -DENABLE_KLEE_UCLIBC=ON \
 -DKLEE_UCLIBC_PATH="../../klee-uclibc" \
 -DENABLE_UNIT_TESTS=OFF \
 -DENABLE_SYSTEM_TESTS=OFF \
 -DLLVM_CONFIG_BINARY="${KLEE_DIR}/bin/llvm-config" \
 ..
```

Final installed build in /usr/local/stow/klee. Activate by `sudo stow --dir=/usr/local/stow klee`

MacOS Sierra (with homebrew)

Step 1: Install required tools for the build

```
Packages to install (home brew):
boost cmake ninja gperftools graphviz doxygen

Set the install path:
export KLEE_DIR=$(brew --prefix)/Cellar/klee/1.3.0
```

Step 2: LLVM

Source	URL	Directory
llvm	http://releases.llvm.org/3.4.2/llvm-3.4.2.src.tar.gz	llvm-3.4
clang	http://releases.llvm.org/3.4.2/cfe-3.4.2.src.tar.gz	llvm-3.4/tools/clang
compiler rt	http://releases.llvm.org/3.4/compiler-rt-3.4.src.tar.gz	llvm-3.4/projects/compiler-rt
test suite	http://releases.llvm.org/3.4/test-suite-3.4.src.tar.gz	llvm-3.4/projects/test-suite

```
cd llvm-3.4
mkdir build_release
cd build_release

cmake -G "Ninja" \
 -DCMAKE_BUILD_TYPE:STRING='Release' \
 -DCMAKE_INSTALL_PREFIX="${KLEE_DIR}" \
 -DLLVM_TARGETS_TO_BUILD:STRING='host' \
 ..

ninja
ninja install
cd ..\..
```

Step 3: Minisat

```
git clone https://github.com/stp/minisat.git
cd minisat
mkdir build_release
cd build_release
cmake -G "Ninja" \
 -DCMAKE_INSTALL_PREFIX=${KLEE_DIR} \
 -DCMAKE_BUILD_TYPE='Release' \
 ..

ninja
ninja install
cd ..\..
```

Step 4: STP

```
git clone https://github.com/stp/stp.git
```

```
cd stp
git checkout stp-2.2.0
mkdir build_release
cd build_release
cmake -G "Ninja" \
 -DENABLE_PYTHON_INTERFACE:BOOL=OFF \
 -DCMAKE_BUILD_TYPE="Release" \
 -DTUNE_NATIVE:BOOL=ON \
 -DCMAKE_INSTALL_PREFIX=${KLEE_DIR} \
 -DMINISAT_INCLUDE_DIR="${KLEE_DIR}/include" \
 ..

ninja
ninja install
cd ../..
```

Step 5: Z3

```
git clone https://github.com/Z3Prover/z3.git
cd z3
git checkout z3-4.5.0
python2 scripts/mk_make.py - --prefix=${KLEE_DIR}
cd build
make -j
make install
cd ../..
```

Step 6: org-klee

```
git clone https://github.com/klee/klee.git org-klee
```

Step 7: opt-klee

```
git clone https://github.gatech.edu/arktos/opt-klee.git
cd opt-klee
mkdir build_release
cd build_release

cmake -G "Ninja" \
 -DCMAKE_INSTALL_PREFIX=${KLEE_DIR} \
 -DCMAKE_BUILD_TYPE:STRING='Release' \
 -DCMAKE_EXPORT_COMPILE_COMMANDS=ON \
 -DCMAKE_CXX_FLAGS="-fno-rtti" \
 -DUSE_CXX11=ON \
 -DENABLE_TCMALLOC=ON \
 -DENABLE_SOLVER_STP=ON \
 -DENABLE_SOLVER_Z3=ON \
 -DENABLE_UNIT_TESTS=OFF \
 -DENABLE_SYSTEM_TESTS=OFF \
 -DLLVM_CONFIG_BINARY="${KLEE_DIR}/bin/llvm-config" \
 ..
```

```
ninja
ninja install
cd ../..
```

Final installed build in /usr/local/Cellar/klee/1.3.0. Activate by brew link klee