Summary of Activities on Symbolic Execution using Klee-uClibc/uClibc++

1. Conversion of the sample code in C to C++. It can be found in Codes/examples. The codes that used the c++ libraries encountered segmentation faults.
2. Understanding the differences in the code between uClibc and Klee-uClibc. It is documented in CodeAnalysis.pdf.
3. Changing the uClibc++ header files by wrapping it in C section as :

   #ifdef __cplusplus
   extern "C" {
   #include <stdint.h>
   #include <stddef.h>
   #endif
4. Reading papers/projects related to other approaches to symbolic execution in C++. Some of the papers/projects referred to were:
   1. LLBMC: Bounded Model Checking of C and C++ Programs Using a Compiler IR*
      http://llbmc.org/usage.html
      http://llbmc.org/files/papers/VSTTE12.pdf

   2. Klover
      https://www.cs.utah.edu/~ligd/publications/KLOVER-IL.pdf
      https://www.cs.utah.edu/~ligd/publications/LAZY_HVC13

   3. GKlee
      http://formalverification.cs.utah.edu/GKLEE/
      http://formalverification.cs.utah.edu/pdf/PPoPP12-GKLEE-Extended-Version.pdf

   4. Cloud9
      https://sites.google.com/site/dslabepfl/proj/cloud9
      http://dslab.epfl.ch/pubs/cloud9.pdf
      http://dslab.epfl.ch/pubs/cloud9-ladis.pdf

5. Building uClibc++ with Klee by modifying the configuration files of klee. It works with the same parameters as building klee with uClibc. The path of the uClibc++ library needs to specified in the *–with-uclibc* parameter. The modified uClibc++ code can be found in Codes/klee-c++. The modified klee code can be found in Codes/klee-uClibc++.
6. Changing the implementation of string compare method in uClibc++as mentioned in the Klover project. It did not behave as expected. There is no segementation faults but the different paths are not generated.

Other Useful Info

Scripts used to compile Klee using uClibc++

./configure --with-stp=<path_to_stp_installation> --with-uclibc=<path_to_uClibc++> --enable-posix-runtime --with-llvmsrc=<path_to_llvm> --with-llvmobj=<path_to_llvm> --with-llvmcc=<path_to_clang> --with-llvmcxx=<path_to_clang++>

Compile C++ Programs using Klee

clang++-3.4 -emit-llvm -c -g <name_of_c++_program>