

# KLEE Tutorial

Wednesday, April 03, 2019 2:33 PM

Install docker. See <https://docs.docker.com/docker-for-windows/>.

Visit <http://klee.github.io/docker/> for installing Klee on docker.

```
$ docker pull klee/klee:2.0
```

```
$ docker run --rm -ti --ulimit='stack=-1:-1' klee/klee:2.0
```

Tutorial: <http://klee.github.io/tutorials/testing-function/>

The Program:

```
int get_sign(int x) {
    if (x == 0)
        return 0;

    if (x < 0)
        return -1;
    else
        return 1;
}
```

To perform symbolic execution, make the input symbolic.

```
int main() {
    int a;
    klee_make_symbolic(&a, sizeof(a), "a");
    return get_sign(a);
}
```

```
$ cd klee_src/examples/get_sign
```

```
$ clang -I .././include -emit-llvm -c -g -O0 -Xclang -disable-O0-optnone get_sign.c
```

```
$ klee get_sign.bc
```

Note that bc refers to LLVM bitcode format.

```
KLEE: output directory = "klee-out-0"
```

```
KLEE: done: total instructions = 33
```

```
KLEE: done: completed paths = 3
```

```
KLEE: done: generated tests = 3
```

```
$ ls klee-last/
```

```
assembly.ll  run.istats  test000002.ktest
info        run.stats   test000003.ktest
messages.txt test000001.ktest warnings.txt
```

```
$ ktest-tool klee-last/test000001.ktest
```

```
ktest file : 'klee-last/test000001.ktest'
```

```
args      : ['get_sign.bc']
```

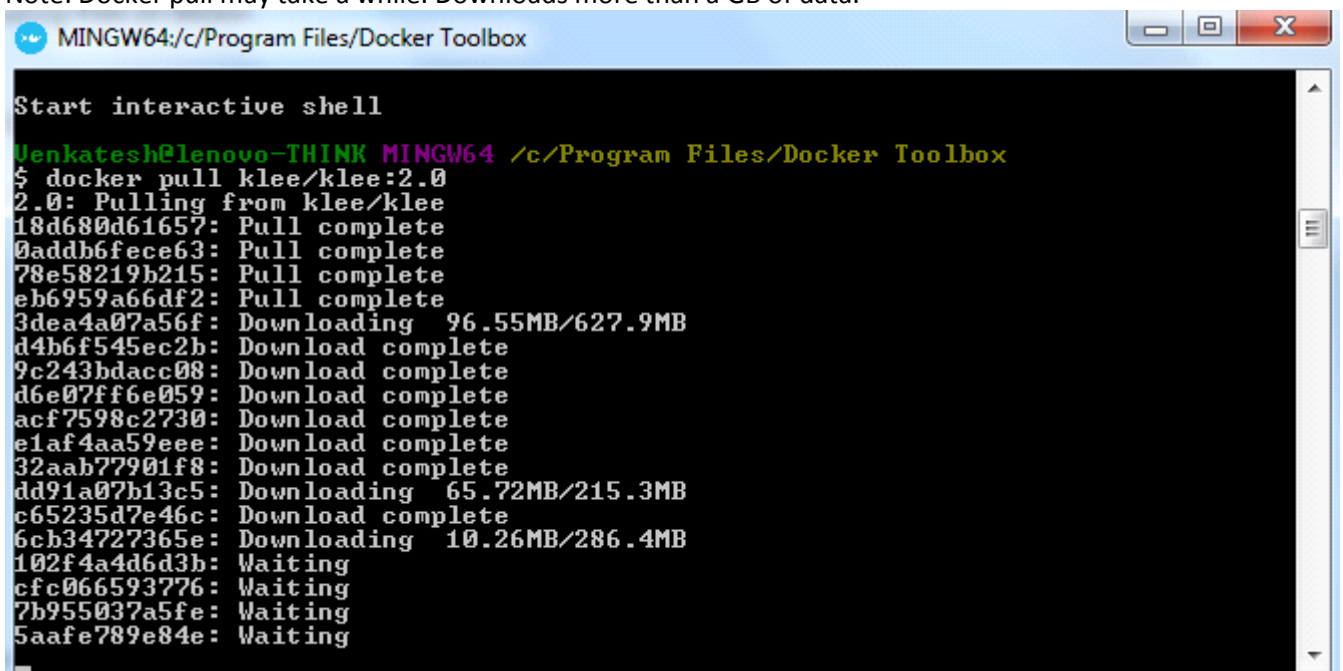
```
num objects: 1
object 0: name: 'a'
object 0: size: 4
object 0: data: b'\x00\x00\x00\x00'
object 0: hex : 0x00000000
object 0: int : 0
object 0: uint: 0
object 0: text: ....
```

```
$ ktest-tool klee-last/test000002.ktest
ktest file : 'klee-last/test000002.ktest'
args      : ['get_sign.bc']
num objects: 1
object 0: name: 'a'
object 0: size: 4
object 0: data: b'\x01\x01\x01\x01'
object 0: hex : 0x01010101
object 0: int : 16843009
object 0: uint: 16843009
object 0: text: ....
```

```
$ ktest-tool klee-last/test000003.ktest
ktest file : 'klee-last/test000003.ktest'
args      : ['get_sign.bc']
```

```
num objects: 1
object 0: name: 'a'
object 0: size: 4
object 0: data: b'\x00\x00\x00\x80'
object 0: hex : 0x00000080
object 0: int : -2147483648
object 0: uint: 2147483648
object 0: text: ....
```

Note: Docker pull may take a while. Downloads more than a GB of data.



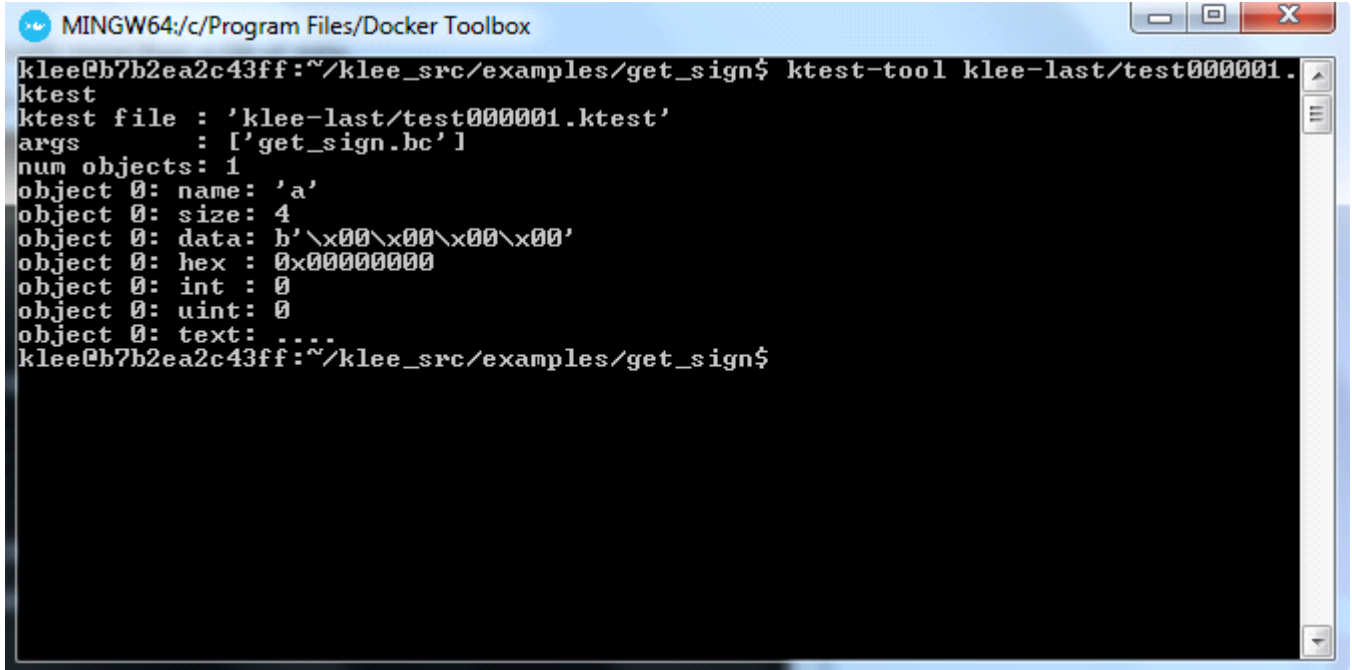
```
MINGW64:/c/Program Files/Docker Toolbox
Start interactive shell
Venkatesh@lenovo-THINK MINGW64 /c/Program Files/Docker Toolbox
$ docker pull klee/kee:2.0
2.0: Pulling from klee/kee
18d680d61657: Pull complete
0addb6fece63: Pull complete
78e58219b215: Pull complete
eb6959a66df2: Pull complete
3dea4a07a56f: Downloading 96.55MB/627.9MB
d4b6f545ec2b: Download complete
9c243bdacc08: Download complete
d6e07ff6e059: Download complete
acf7598c2730: Download complete
e1af4aa59eee: Download complete
32aab77901f8: Download complete
dd91a07b13c5: Downloading 65.72MB/215.3MB
c65235d7e46c: Download complete
6cb34727365e: Downloading 10.26MB/286.4MB
102f4a4d6d3b: Waiting
cfc066593776: Waiting
7b955037a5fe: Waiting
5aafe789e84e: Waiting
```

Docker is ready. Let us run the get\_sign example.

```
klee@b7b2ea2c43ff:~/klee_src/examples/get_sign$ klee get_sign.bc
KLEE: output directory is "/home/klee/klee_src/examples/get_sign/klee-out-1"
KLEE: Using STP solver backend

KLEE: done: total instructions = 33
KLEE: done: completed paths = 3
KLEE: done: generated tests = 3
klee@b7b2ea2c43ff:~/klee_src/examples/get_sign$
```

Klee identified three paths and generated three corresponding tests. Here is one of them.

A screenshot of a terminal window titled "MINGW64:/c/Program Files/Docker Toolbox". The terminal shows the execution of the "ktest-tool" command on a file named "klee-last/test000001.ktest". The output lists the file name, arguments, and details for one object, including its name, size, data, hex representation, and integer/unsigned integer/text values.

```
klee@b7b2ea2c43ff:~/klee_src/examples/get_sign$ ktest-tool klee-last/test000001.ktest
ktest
ktest file : 'klee-last/test000001.ktest'
args       : ['get_sign.bc']
num objects: 1
object 0: name: 'a'
object 0: size: 4
object 0: data: b'\x00\x00\x00\x00'
object 0: hex : 0x00000000
object 0: int  : 0
object 0: uint : 0
object 0: text: ....
klee@b7b2ea2c43ff:~/klee_src/examples/get_sign$
```