Automated Software Testing: Fuzzing Techniques (Cont.)

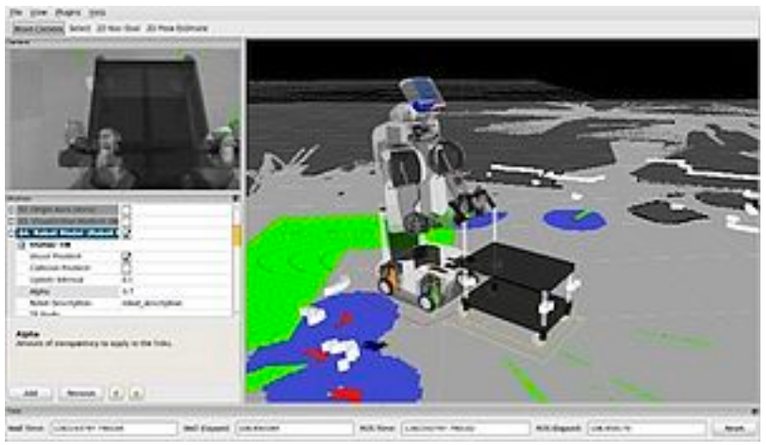
Zhoulai Fu

IT University of Copenhagen

September 30, 2021

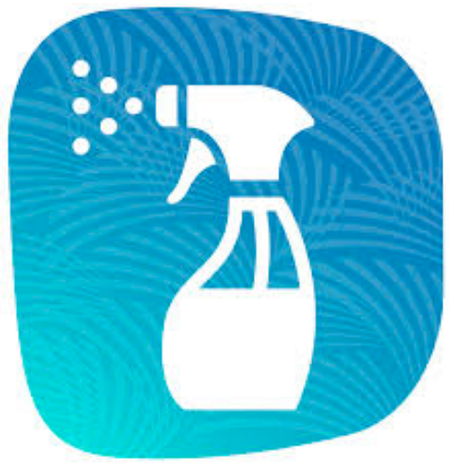
Today

Sanitizing techniques



Fuzzing for real-world

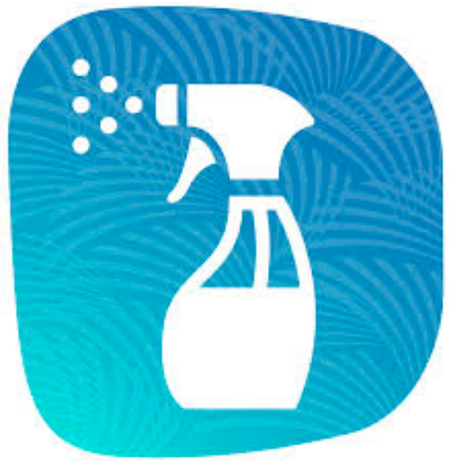
Two mini-projects



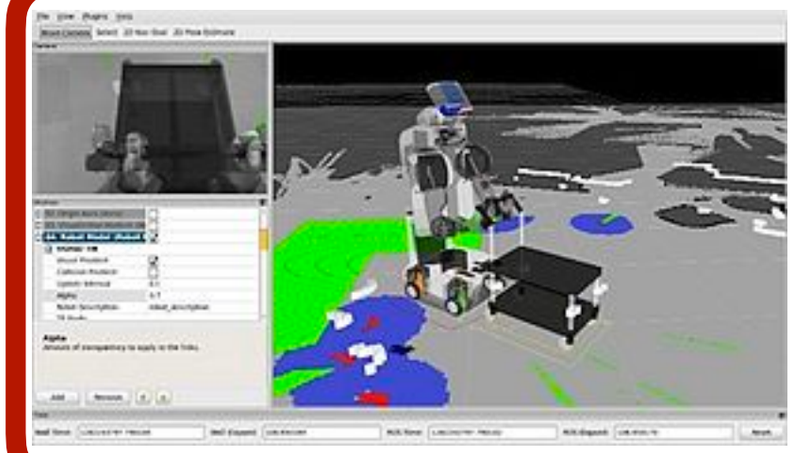
Sanitizing techniques

- In automate testing, an utmost challenge is _____
- Free spec includes _____
- Sanitizing is free spec integrated in today’s compilers

DEMO



Sanitizing techniques



Fuzzing for real-world

Two mini-projects

How fuzzing works?

```
69  int main(int argc, char *argv[])
70  {
71          char *usage = "Usage: %s\n"
72                                  "Text utility - accepts comma
73                                  "\tInput              | Output
74                                  "\t-------------------+-------
75                                  "\tu <N> <string>    | Upperc
76                                  "\thead <N> <string> | The fi
77          char input[INPUTSIZE] = {0};
78
79          // Slurp input
80          if (read(STDIN_FILENO, input, INPUTSIZE) < 0)
81          {
82                  fprintf(stderr, "Couldn't read stdin.\n");
83          }
84
85          int ret = process(input);
```

- Need “main”, the fuzz driver
- Instrument code to monitor coverage.
- Generate input data.

How fuzzing works?

From vulnerable.c

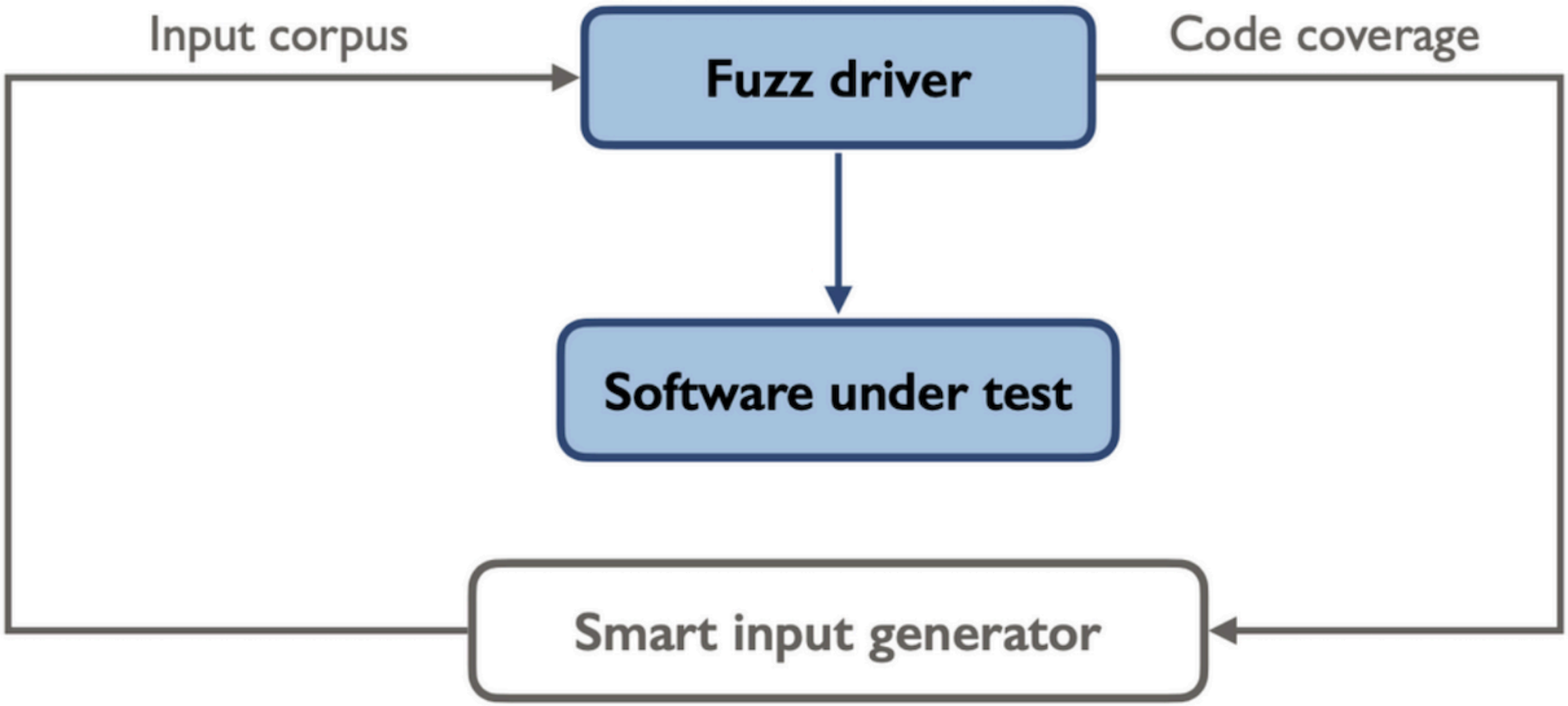
```
69  int main(int argc, char *argv[])
70  {
71          char *usage = "Usage: %s\n"
72                                  "Text utility - accepts comma
73                                  "\tInput             | Output
74                                  "\t------------------+-------
75                                  "\tu <N> <string>    | Upperc
76                                  "\thead <N> <string> | The fi
77          char input[INPUTSIZE] = {0};
78
79          // Slurp input
80          if (read(STDIN_FILENO, input, INPUTSIZE) < 0)
81          {
82                  fprintf(stderr, "Couldn't read stdin.\n");
83          }
84
85          int ret = process(input);
```

- “main” function
- interacts with software under test
- takes an input from stdin or file

Aft-fuzz

?

Afl-gcc

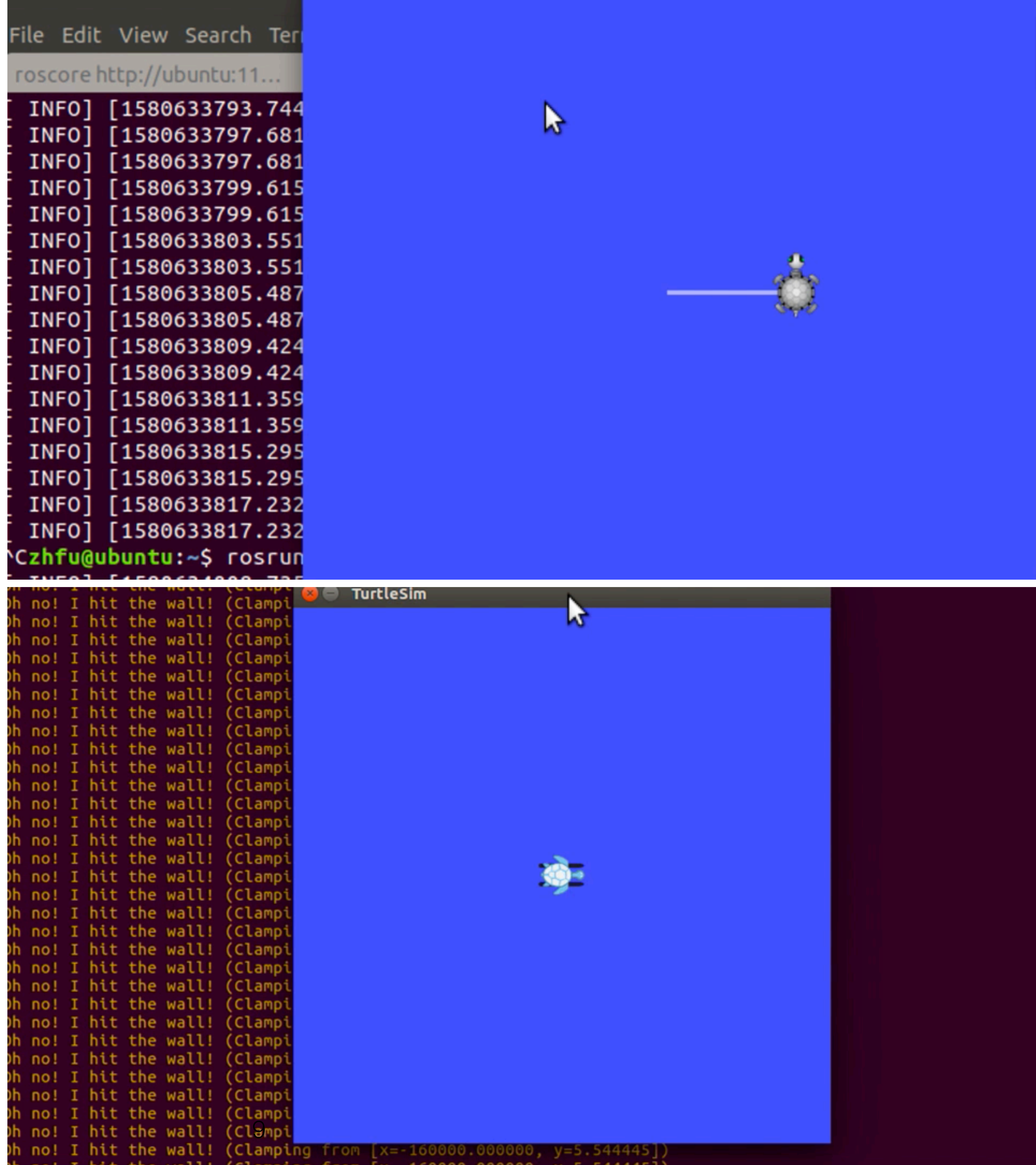


How to get Fuzz Driver?

How to get Fuzz Driver?

- FuzzGen: Automatic Fuzzer Generation
- IntelliGen: Automatic Driver Synthesis for Fuzz Testing
- FUDGE: Fuzz Driver Generation at Scale
- WINNIE : Fuzzing Windows Applications with Harness Synthesis and Fast Cloning
  - Written manually in general
  - But can be automated for software involving floating-point calculation!

Fuzzing Robot Operating System ([ros.org](ros.org))



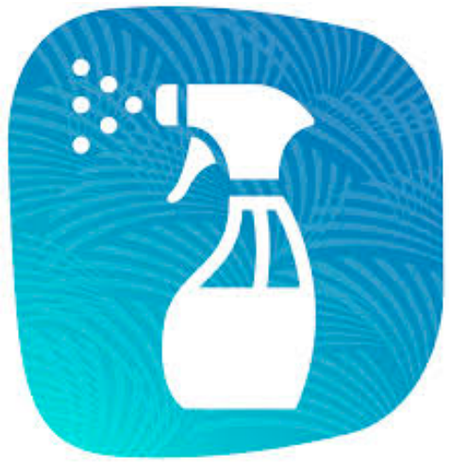
Before

```
1 void forward(ros::Publisher twist_pub) {
2    if (hasReachedGoal()) { ... }
3    else commandTurtle(twist_pub, 1.0, 0.0);
4 }
5 ...
6 int main(int argc, char** argv){
7   ...
8 }
```

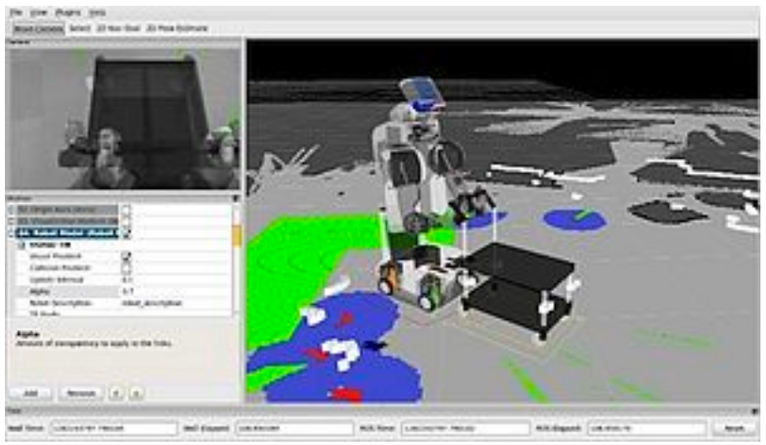
Fuzzing Robot Operating System

After

```
1  double g1, g2;
2  void forward(ros::Publisher twist_pub) {
3    if (hasReachedGoal()) { ... }
4    else commandTurtle(twist_pub, g1, g2);
5  }
6  ...
7  int main(int argc, char** argv) {
8    if (scanf("%lf,%lf", &g1,&g2) != 2) return -1;
9    ...
10 }
```



Sanitizing techniques



Fuzzing for real-world

Two mini-projects