Realistic Verification Scenarios

http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems

Pavel Parízek

FACULTY OF MATHEMATICS AND PHYSICS
Charles University

Concurrent HashMap (Java)

- Where to get it
  - `<JDK DIR>/src.zip:java/util/concurrent`

- Task 1: identify relevant correctness properties
- Task 2: determine suitable verification approach
- Task 3: maybe actually use tools that you know

GNU core utilities

- Where to get the source code
  - http://ftp.gnu.org/gnu/coreutils/coreutils-8.12.tar.gz
- Selected target programs: `cp`, `su`
- Task 1: identify relevant correctness properties
- Task 2: determine suitable verification approach
- Task 3: maybe actually use tools that you know

Department of Distributed and Dependable Systems D3S

HelenOS

- Source code
  - http://www.helenos.org/releases/HelenOS-0.5.0.tar.bz2

- Task 1: identify relevant correctness properties
- Task 2: determine suitable verification approach
- Task 3: how would you proceed with analysis of such large program

Interesting paper

- D. Engler, et al. **A few billion lines of code later: using static analysis to find bugs in the real world**, Communications of the ACM, 53(2), February 2010
  - http://web.stanford.edu/~engler/BLOC-coverity.pdf
  - http://www.coverity.com/ (https://www.synopsys.com/)

- Key challenges
  - Parsing the real code (different standards of C/C++/Java)
  - Integration with the internally used build system
    - "Make" variants, not everybody uses Make, some GUI tools (IDE)
  - Bug reports: false positives, developers not understand
  - Social: convince managers that they want to fix bugs

Department of Distributed and Dependable Systems D3S

SV-COMP

- Competition on Software Verification

- http://sv-comp.sosy-lab.org/
- http://sv-comp.sosy-lab.org/2022/

Department of Distributed and Dependable Systems