




# Lecture 6 Coverage



Zvonimir Rakamarić  
University of Utah

material acknowledgements: John Regehr

# How Much Testing is Enough?

- ▶ Assign “score” to your testing effort
- ▶ That score is typically called *coverage*

# Test Partitioning vs Coverage

- ▶ Manual test partitioning is often hard (impossible) to come up with for large complicated systems
  - ▶ Think whole OS or text processor
- ▶ Coverage is an automatic way that achieves partitioning
- ▶ Example: executed functions

# Test Coverage

- ▶ Score describing which partitions were covered
- ▶ Measures proportion of program exercised during testing
- ▶ Pros
  - ▶ Score
  - ▶ Relatively clear action items: we (roughly) know what to do when it is not 100%
    - ▶ Rethink your test suite and do not just blindly try to increase coverage
- ▶ Cons
  - ▶ White-box – cannot find what is not implemented
  - ▶ What if it is less than 100%?
  - ▶ Can we miss bugs even if it is 100%?

# Coverage Metrics

- ▶ There is many of them
- ▶ Appear often in practice
  - ▶ Statement coverage (example, can it fail?)
  - ▶ Line coverage (almost the same as statement coverage, multiple statements on the same line)
  - ▶ Branch coverage
- ▶ Others
  - ▶ Loop (0, once, more than once)
  - ▶ Modified condition/decision coverage (MC/DC)
    - ▶ Mission critical software (avionics), focus on conditionals
  - ▶ Path coverage
    - ▶ Not achievable in practice. Why? Show example

# When Coverage Does Not Work?

- ▶ “errors of omissions” – something is not implemented
  - ▶ Example: forgot to check some error code
- ▶ Possible solution
  - ▶ Test partitioning based on specification
- ▶ Which one to use in the end?
  
- ▶ Dealing with infeasible code
  - ▶ Sometimes you can mark it as such
  
- ▶ Convince yourself that code that is not covered is not crucial

# How to Use Code Coverage?

- ▶ Write good tests
- ▶ Measure coverage to get more information about tests
  - ▶ Good – get feedback to improve tests further
  - ▶ Poor – maybe rethink your testing strategy
- ▶ Do good tests imply good coverage?
- ▶ Does good coverage imply good tests?

# Using gcov

- ▶ Exercise using gcov for reporting code coverage