Statische Codeanalyse

Wo ist der Fehler in meinem Programm?

Wolfgang Dautermann

FH JOANNEUM

Chemnitzer Linuxtage 2014

Statische Codeanalyse

Analyse durch Inspektion des Sourcecodes – keine Programmausführung

- Lint: 1979 entwickelt
- Ausgleich der Schwächen von Compilern
- Open Source: Splint (Secure Programming Lint, `http://www.splint.org/`)
- (+ etliche ähnliche Tools für div. Programmiersprachen...)

Compiler haben Schwächen?

...ich benutze eh brav die Option `-Wall`

Welche Fehler sind in folgendem Programm?

```
#include <stdio.h>
int main(int argc, char * argv[])
{
        if (argc = 10+1) {
                printf("10 Argumente!\n");
        }
        return 0;
        printf("Programm beendet\n");
}
```

Wir compilieren unser Programm...

- `gcc main.c`
- `gcc -Wall main.c`
- `gcc -Wall -Wextra main.c`
- `clang main.c`
- `/opt/open64/bin/opencc main.c`
- `/opt/open64/bin/opencc -Wall main.c`
- `/opt/oracle/solarisstudio12.3/bin/suncc main.c`
- `/opt/tinycc/bin/tcc main.c`

Schaun wir mal, welche Fehler erkannt und gemeldet werden...

Codechecker – splint

Eigenschaften

- Syntaxfehler werden (möglicherweise) nicht erkannt
  (Das ist Compiler-Aufgabe)
- (syntaktisch richtige) fehlerhafte Programmkonstrukte werden erkannt.
- **lesbare** Fehlermeldungen und Verbesserungsvorschläge.
- Zusätzliche Steuerungsmöglichkeiten über Kommandozeilenparameter und spezielle Kommentare (Annotations)

splint - Codechecker

splint main.c

```
$ splint main.c
Splint 3.1.2 --- 16 Jul 2012

main.c: (in function main)
main.c:4:6: Test expression for if is assignment expression: argc = 10 + 1
  The condition test is an assignment expression. Probably, you mean to use ==
  instead of =. If an assignment is intended, add an extra parentheses nesting
  (e.g., if ((a = b)) ...) to suppress this message. (Use -predassign to
  inhibit warning)
main.c:4:6: Test expression for if not boolean, type int: argc = 10 + 1
  Test expression type is not boolean or int. (Use -predboolint to inhibit
  warning)
main.c:8:2: Unreachable code: printf("Programm...
  This code will never be reached on any possible execution. (Use -unreachable
  to inhibit warning)
main.c:2:27: Parameter argv not used
  A function parameter is not used in the body of the function. If the argument
  is needed for type compatibility or future plans, use /*@unused@*/ in the
  argument declaration. (Use -paramuse to inhibit warning)

Finished checking --- 4 code warnings
```

splint - Codechecker

Optionen von Splint

- `splint --help`
- `splint --help ...`
- Härtegrade:
  `splint --weak / --standard / --checks / --strict`

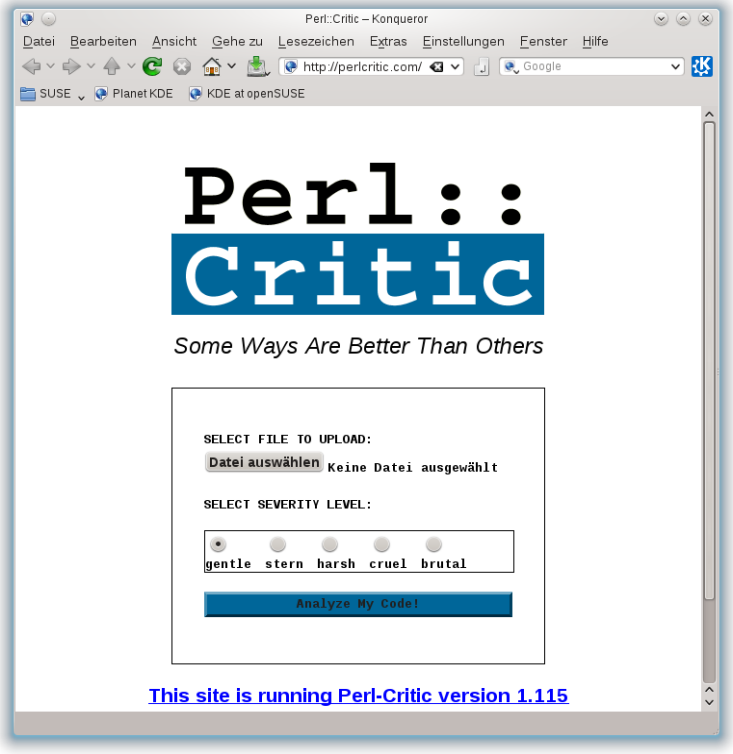
Weiterer C/C++ Codechecker: cppcheck

http://cppcheck.sourceforge.net/

- `cppcheck --enable=all main.c`
- Prüft ggf. auch alle möglichen `#define`-Kombinationen
- `cppcheck --enable=all` / `warning` / `style` / `performance` / `portability` / `information` / `unusedFunction` / `missingInclude`
- Report auch als XML möglich:
  `cppcheck --enable=all --xml [sources] 2>report.xml`
- ... und Webseite erzeugen:
  `cppcheck-htmlreport --file=report.xml --report=reportdir`

perlcritic

...als Webservice: http://perlcritic.com



perlcritic

...auf der Kommandozeile

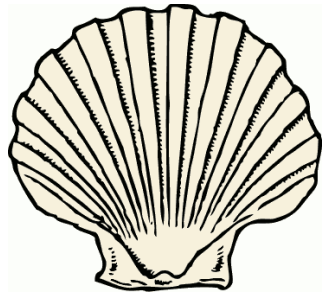
```
perlcritic [--brutal | --cruel | --harsh | --stern | --gentle]  test.pl
perlcritic --list-themes
perlcritic --theme=xxxx test.pl
```

spezielle Kommentare

```
perl-befehl;  ## no critic

## no critic
befehle...
## do critic
```

Shell



- Online Shellcheck: `http://www.shellcheck.net/`
  (auch als Open-Source-Tool downloadbar (geschrieben in Haskell)
- `checkbashisms`: Prüft auf Bash-spezialitäten[1] in `#!/bin/sh`-Skripten

[1] `http://mywiki.wooledge.org/Bashism`

Lint-Programme für Nicht-Programmiersprachen

- LaTeX[2]-Paket `nag`
- LaTeX: `chktex`
- RPM-Pakete: `rpmlint`
- DEB-Pakete: `lintian`

[2] Wobei man LaTeX durchaus auch als Programmiersprache, nicht nur als Textsatzsystem ansehen kann...

Web & Co

- Javascript: http://www.jslint.com/
- PHP Code sniffer (`phpcs`):
  http://pear.php.net/package/PHP_CodeSniffer/
  (`pear install PHP_CodeSniffer`)
- (W3-HTML-Validator http://validator.w3.org)
- (W3-CSS-Validator http://jigsaw.w3.org/css-validator/)

Vielen Dank

Fragen? (hoffentlich richtige...) Antworten!

- Literaturhinweis: Artikel in Linux User 11/2012:
  `http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2012/11/Splint-und-Co-Tools-zur-statischen-Code-Analyse`

Vielen Dank für Ihre Aufmerksamkeit

Wolfgang Dautermann
wolfgang.dautermann [AT] fh-joanneum.at