



**BERGISCHE
UNIVERSITÄT
WUPPERTAL**

Prof. Dr. Hans-Jürgen Buhl
Praktische Informatik/Numerik

Fakultät für
Mathematik und Naturwissenschaften,
Mathematik und Informatik

E-MAIL buhl@math.uni-wuppertal.de

WWW www.math.uni-wuppertal.de/~buhl

DATUM 11. Januar 2017

Softwarequalität

WS 2016/2017 – Übungsblatt 8

Ausgabe: 21. Dezember 2016

Abgabe bis 11. Januar 2017 an: <mailto:1449250@uni-wuppertal.de>

Aufgabe 1. *Installationshinweise Savannah Nana*

Laden Sie von

<http://savannah.gnu.org/projects/nana/>

die Dateien `nana-2.5.tar.gz` und `nana-2.5.tar.gz.sig` herunter, besorgen Sie sich den öffentlichen gpg-Schlüssel von Phil Maker (über <http://savannah.gnu.org/project/memberlist-gpgkeys.php?group=nana> oder einen der gpg-Keyserver mittels `seahorse`, `gpg2`, `gpa`, `kgpg`, `gpgv2`, ...) und überprüfen Sie die Authentizität der von Ihnen heruntergeladenen Dateien:

```
p> gpg2 --search-keys Phil Maker
gpg: suche nach "Phil Maker" auf hkp-Server wwwkeys.eu.pgp.net
(1) Phil Maker <pjm@gnu.org>
    1024 bit DSA key AD58EA42, erzeugt: 2005-04-13
(2) Phil Maker <pjm@gnu.org>
    2048 bit RSA key 7D25F017, erzeugt: 1998-11-03
Keys 1-2 of 2 for "Phil Maker".  Eingabe von Nummern, Nächste (N) oder Abbrechen (Q) >
1
gpg: fordere Schlüssel AD58EA42 von hkp-Server wwwkeys.eu.pgp.net an
gpg: Schlüssel AD58EA42: Öffentlicher Schlüssel "Phil Maker <pjm@gnu.org>" importiert
gpg: Anzahl insgesamt bearbeiteter Schlüssel: 1
gpg:                                importiert: 1

> gpg2 nana-2.5.tar.gz.sig
gpg: Unterschrift vom Mi 13 Apr 2005 05:48:25 CEST mittels DSA-Schlüssel ID AD58EA42
gpg: Korrekte Unterschrift von "Phil Maker <pjm@gnu.org>"
gpg: WARNUNG: Dieser Schlüssel trägt keine vertrauenswürdige Signatur!
gpg:           Es gibt keinen Hinweis, daß die Signatur wirklich dem vorgeblichen Besitzer gehört.
Haupt-Fingerabdruck = 1806 4A1E F48D 8372 3BE8 156E A266 78A3 AD58 EA42
```

(Warum ist das erforderlich?)

Entpacken Sie das Archiv:

```
tar xzf nana-2.5.tar.gz
cd nana-2.5
```

Führen Sie ein paar wegen inzwischen in der Programmiersprache C durchgeführter Änderungen nötige Modifikationen durch

```

Zeile 87 von src/I.h:
void _I_default_handler(const char *expr, const char *file, int line);

Zeile 48 von src/I.c:
void _I_default_handler(const char *exprn, const char *file, int line) {

statt Zeile 84 von src/nana-config.h.in:
typedef void*(FKTxy)(unsigned int);
static FKTxy *_nana_p = malloc; /* this costs us storage space */

Zeil 82 von src/nana-clg.in:
${NANACC-@CC@} $CPPFLAGS $* -L$HOME/lib -lnana -o a.out &&

Zeile 73 von src/nana.in:
@CPP@ -I$HOME/include -D_NANA_FILTER_ $* 2>/dev/null | ${NANAEXECDIR-@libexecdir@}/nanafilter

Zeile 63 in shortform/nana-sfdir.in:
    ${NANABIN-@bindir@}/nana-sfg $f >$TARGET/$f

neue Zeile 58 in shortform/nana-sfdir.in einfügen:
    -name \*.cpp -o \

Zeile 33 von src/nana_error.c:
#include <stdlib.h>

Zeile 36 von src/nanafilter.c:
#include <stdlib.h>

Zeile 31 von src/I.c:
#include <stdlib.h>

```

und installieren Sie nana in Ihrem Home-Verzeichnis:

```

nana-2.5> ./configure --prefix=~
creating cache ./config.cache
checking host system type... i686-pc-linux-gnu
...
nana-2.5> make
Making all in src
make[1]: Entering directory `/home/User/Desktop/nana-2.5/src'
gcc -DPACKAGE=\"nana\" -DVERSION=\"2.5\" -DHAVE_VSPRINTF=1 -DHAVE_VSNPRINTF=1 -DHAVE_GETTIMEOFDAY=1 -I. -I. -g -I. \
-I/home/User/include -g -I. -I/home/User/includude -c I.c
...
nana-2.5> make install
Making install in src
make[1]: Entering directory `/home/User/Desktop/nana-2.5/src'
make[2]: Entering directory `/home/User/Desktop/nana-2.5/src'
/bin/sh ../mkinstalldirs ~/lib
...

```

(Lesen Sie bei Bedarf die Datei `INSTALL` beziehungsweise `doc/nana.ps`)

Ändern Sie Ihr Linux-Environment (Datei `~/.bashrc`) durch Hinzufügen von:

```

export NANAPKGDIR=$HOME/share/nana
# export NANARUN_GDB_OPTIONS=
export NANABIN=~/.bin
export NANACC=gcc

export CPPFLAGS="-g -I. -I$HOME/include"
export CFLAGS="-g -I. -I$HOME/include"
export LDFLAGS="-L $HOME/lib"
export LDLIBS="-lnana"

```

Um diese Änderung für Sie funktionsfähig zu machen, starten Sie eine neue `bash` oder loggen Sie sich aus und danach wieder ein.

Aufgabe 2. Testläufe unter gdb-Kontrolle

```
emacs testpgm.c

#include <nana.h>

main() {
    int i = 10;

    DL("Hello\n");
    DI(i == 1);
}

-----

nana testpgm.c > testpgm.gdb
make testpgm

gdb testpgm -x testpgm.gdb
Breakpoint 1 at 0x4004d3: file testpgm.c, line 6.
Breakpoint 2 at 0x4004dd: file testpgm.c, line 7.
(gdb) run
Starting program: /home/buhl/nana/testpgm
Hello
"DI("i == 1")" has failed at f:1 with
#0 main () at testpgm.c:7
(gdb) cont
Continuing.

Program exited with code 0320.
(gdb) quit
```

Welche Vorteile hat die Benutzung der D-Varianten (DL(), DI(),..., debugger based) gegenüber den C-basierten Varianten von nana-Kommandos? (Lesen Sie dazu die [Dokumentation](#))

Aufgabe 3. SdV/DbC mit nana

Laden Sie das Beispielprojekt [Wuerfel-nana-src.tar.gz](#) herunter, entpacken Sie es, übersetzen Sie es und testen Sie

- ohne DbC-Vertragsverletzung,
- mit DbC-Vertragsverletzung: je einmal Vorbedingung, Nachbedingung, Invariante nicht erfüllt.

Schreiben Sie eine Benutzeranleitung zum Single-Stepping durch ein mit nana-Contracts versehenes C++-Programm. ([Debug Step Into, ...](#))

Wie kann nana innerhalb von eclipse benutzt werden?

Aufgabe 4. nana-sfg

Laden Sie die modifizierte Version von [nana-sfg](#) herunter und erzeugen Sie damit eine für Dokumentationszwecke geeignete Version von `Wuerfel.cpp`.

Aufgabe 5. nana-Einordnung

Ergänzen Sie SdV-Tabelle in Abschnitt 1.3.3 um eine Spalte für nana-Codeverträge (erwerben Sie die notwendigen nana-Kenntnisse durch Lektüre der [Dokumentation](#)).