



Grundkurs C++

Debugging

Berühmte Software-Bugs

9/9


0800 Antan started
1000 " stopped - antan ✓

13⁰⁰ (033) MP-MC $\left\{ \begin{array}{l} 1.2700 \quad 9.037847025 \\ 1.98249000 \quad 9.037846995 \text{ correct} \\ 2.130476415 \end{array} \right.$
(033) PRO 2 2.130476415
correct 2.130676415

Relays 6-2 in 033 failed special speed test
in Relay " 11.000 test.

Relays changed

1100 Started Cosine Tapc (Sine check)
1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

1630 Antan started.
1700 closed down.

Relay 3375
Relay 3376

1947: Fehlfunktion des Mark II Relay Calculator aufgrund einer Motte.

Berühmte Software-Fehler

- 1996: Ariane 5 muss 40 Sekunden nach dem Start gesprengt werden.
 - Umwandlung der Geschwindigkeit von double nach signed integer -> Überlauf
 - Software wurde von Ariane 4 übernommen, die diese Geschwindigkeit nicht erreichen konnte
 - Schaden: ca. 370 Millionen USD

Berühmte Software-Fehler

- 1994:
 - Fehler in der Divisionsroutine des Pentium-Prozessors
 - $x = 4195835.0$
 - $y = 3145727.0$
 - $z = x - (x/y) * y$
 - Bugs in CPUs sind aufgrund der Komplexität der Designs auch heute üblich.

Fehler zur Compile-Zeit

- Syntaxfehler (Vergessener Strichpunkt...)
- Inkompatible Datentypen
- Fehlende Dereferenzierung (Objekt statt Zeiger auf Objekt benutzt...)
- usw.
- Aber: Compiler können nicht alle Fehler erkennen.

Fehler zur Laufzeit

- Zugriff außerhalb der Grenzen eines Arrays.
- Dynamisch allozierten Speicher nicht wieder freigegeben (Memory-Leak).
- Dynamisch allozierten Speicher mehrfach freigegeben.
- Dereferenzierung nicht-initialisierter Zeiger.
- Signed Integer Overflow
- usw.

gcc Debug-Optionen

- Parameter `-g`
 - Erzeugt Debugging-Symbole in der ausführbaren Datei.
- Parameter `-Wall`
 - Aktiviert einen Satz von Warnungen, die i.d.R. leicht zu beheben sind.
- Parameter `-W`
 - Aktiviert weitere Warnungen, die nicht von `-Wall` eingeschlossen werden.
- Parameter `-Werror`
 - Warnungen werden wie Fehler behandelt und können so nicht mehr übersehen werden.
- Parameter `-O2`
 - Aktiviert Optimierungen (wichtig, da sonst manche Fehler nicht entdeckt werden).

gcc Debug-Symbole

```
GNU gdb (GDB) 7.0-ubuntu
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show
copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/thabigt/Desktop/valgrind/test...done.
(gdb) run
Starting program: /home/thabigt/Desktop/valgrind/test

Program received signal SIGBUS, Bus error.
0x00000000040061a in main ()
```


gcc Debug-Symbole

```
GNU gdb (GDB) 7.0-ubuntu
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show
copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/thabigt/Desktop/valgrind/test...done.
(gdb) run
Starting program: /home/thabigt/Desktop/valgrind/test

Program received signal SIGBUS, Bus error.
main () at test.c:23
23      *(&x + 100000) = 1;
```

gcc Debug-Symbole

- Debug-Symbole in externen Libraries:
 - Installieren der -dbg Pakete
 - Kompilieren der Libraries mit -g Option

gdb

- Benutzung von gdb am einfachsten durch Integration in Qt Creator.

```
44 }
45
46 void Notepad::on_actionSave_triggered()
47 {
48     QString fileName = QFileDialog::getSaveFileName(this, tr("Save File")
49     tr("Text Files (*.txt);;C++ Files (*.cpp *.h)"));
50
51     if (!fileName.isEmpty()) {
52         QFile file(fileName);
53         if (!file.open(QIODevice::WriteOnly)) {
54             // error message
55         } else {
56             QTextStream stream(&file);
57             stream << ui->textEdit->toPlainText();
58             stream.flush();
59             file.close();
60         }
61     }
62 }
63
64 void Notepad::createMessage()
65 {
66     QMessageBox::critical(this, tr("Statusleiste"), tr("Nachricht wird ge
67 }
68
```

Name	Wert	Typ
this	"Notepad"	Notepad
[QMain...	"Notepad"	QMainWindow
staticM...	@0x606b60	QObject
ui	@0x630ec0	Ui::Notepad

Tiefe	Funktion	Datei	Zeile	Zahl	Funktion	Datei	Zeile	Adresse	Bedingung	Anhalten nach
0	Notepad::createMessage	notepad.cpp	66	1	Notepad::cre...	/home/...	66	0x40442b		
1	Notepad::qt_static_metacall	moc_notepad...	81							
2	QObject::activate(QObject*, int, int, void**)	/opt/Qt/5.2.1/...								
3	QAbstractButton::clicked(bool)	/opt/Qt/5.2.1/...								
4	??	/opt/Qt/5.2.1/...								
5	??	/opt/Qt/5.2.1/...								
6	QAbstractButton::mouseReleaseEvent(QMouseEvent*)	/opt/Qt/5.2.1/...								
7	QWidget::event(QEvent*)	/opt/Qt/5.2.1/...								

gdb

- Wichtige Befehle zur Ausführung in der Kommandozeile:
 - **gdb ./executable**
 - **run** [Übergabeparameter]
 - **bt** ← zeigt Stack an
 - **print** Variablenname ← gibt Inhalt der Variable aus
 - **quit**

Valgrind

- **Memcheck**
 - Entdeckt unerlaubte Speicherzugriffe, Memory-Leaks, doppelte Speicherfreigaben...
- **Cachegrind**
 - Cache-Profiler
- **Callgrind**
 - Cachegrind mit zusätzlichen Aufrufstatistiken
- **Massif**
 - Heap-Profiler
- **Helgrind**
 - Thread-Debugger

Valgrind Memcheck

```
void f(void)
{
    int* x = new int[10];
    x[10] = 0;
}

int main(void)
{
    f();
    return 0;
}
```

```
valgrind --leak-check=yes ./main
```

Valgrind Memcheck

```
==13211== Memcheck, a memory error detector
==13211== Copyright (C) 2002-2009, and GNU GPL'd, by Julian
Seward et al.
==13211== Using Valgrind-3.6.0.SVN-Debian and LibVEX; rerun with
-h for copyright info
==13211== Command: ./exec
==13211==
==13211== Invalid write of size 4
==13211==    at 0x40088E: f() (main.cpp:4)
==13211==    by 0x4008A8: main (main.cpp:9)
==13211== Address 0x595e068 is 0 bytes after a block of size 40
alloc'd
==13211==    at 0x4C27939: operator new[](unsigned long)
(vg_replace_malloc.c:305)
==13211==    by 0x40088D: f() (main.cpp:3)
==13211==    by 0x4008A8: main (main.cpp:9)
```

Valgrind Memcheck

```
==13211== HEAP SUMMARY:
==13211==      in use at exit: 40 bytes in 1 blocks
==13211==    total heap usage: 1 allocs, 0 frees, 40 bytes
allocated
==13211==
==13211== 40 bytes in 1 blocks are definitely lost in loss record
1 of 1
==13211==    at 0x4C27939: operator new[](unsigned long)
(vg_replace_malloc.c:305)
==13211==    by 0x40088D: f() (main.cpp:3)
==13211==    by 0x4008A8: main (main.cpp:9)
==13211==
==13211== LEAK SUMMARY:
==13211==    definitely lost: 40 bytes in 1 blocks
==13211==    indirectly lost: 0 bytes in 0 blocks
==13211==    possibly lost: 0 bytes in 0 blocks
==13211==    still reachable: 0 bytes in 0 blocks
==13211==    suppressed: 0 bytes in 0 blocks
```


Valgrind Integration in Qt Creator

```
7 #include <QTextStream>
8
9 Notepad::Notepad(QWidget *parent) :
10     QMainWindow(parent),
11     ui(new Ui::Notepad)
12 {
13     ui->setupUi(this);
14     connect(ui->pushButton, SIGNAL(clicked()), ui->statusBar, SLOT(clearMessage()));
15     connect(ui->pushButton, SIGNAL(clicked()), this, SLOT(createMessage()));
16     ui->statusBar->showMessage("Applikation geladen.");
17     int* leak = new int(5);
18 }
19
20 Notepad::~Notepad()
21 {
22     delete ui;
23 }
24
25
```

Speicheranalyse mit Valgrind

4 bytes in 1 blocks are definitely lost in loss record 5 of 212 in Notepad::Notepad(QWidget*) in notepad.cpp:17

- 1: operator new(unsigned long) in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so
- 2: Notepad::Notepad(QWidget*) in notepad.cpp:17
- 3: main in main.cpp:7

in 0x0

Valgrind und gdb Integration in Qt Creator

- Anleitung unter:

<http://doc.qt.io/qtcreator/creator-debugging.html>

Valgrind Callgrind

```
valgrind --tool=callgrind ./main
```

- Erzeugt Datei callgrind.out.[Prozess-ID]
- Kann mit kcachegrind visualisiert werden.

Valgrind Callgrind

The screenshot displays the Valgrind Callgrind interface for a program named 'callgrind.out.5065'. The interface is divided into several panes:

- Kostenprofil (Cost Profile):** A table showing the cost of various functions. The columns are 'Inkl.', 'Self', 'Aufgerufen', 'Funktion', and 'Ort'. The total cost for the entire run is 342,534,701 instructions.
- Dialog::Dialog():** A pane showing the source code of the function being analyzed. The code includes headers, function definition, and initialization of variables like 'jackcon', 'scene', and 'view'.
- Call Graph:** A hierarchical diagram showing the call flow. The root node is 'main' (23.68% cost), which calls 'Dialog::Dialog()' (23.68% cost). 'Dialog::Dialog()' then calls 'JackConnection::JackConnection()' and 'QGraphicsScene::addText(QString const&, QFont const&)'.

Inkl.	Self	Aufgerufen	Funktion	Ort
11.88	11.87	991 392	getc	libc-2.10.1.so: getc.c, lib
16.67	4.79	1 033 858	0x000000000006d520	libX11.so.6.2.0
4.01	4.01	1 933 773	strcmp'2	libc-2.10.1.so: strcmp.S
36.51	3.96	2	_XimParseStringFile	libX11.so.6.2.0
20.63	3.95	137 760	0x000000000006d590	libX11.so.6.2.0
3.32	3.30	54 217	_XrmInternalStringToQuark	libX11.so.6.2.0
6.47	2.96	11 051	0x0000000000014720	libfontconfig.so.1.3.0
3.48	2.70	8 177	do_lookup_x	ld-2.10.1.so: do-lookup.h
2.85	2.68	110 923	int_malloc	libc-2.10.1.so: malloc.c
3.27	2.29	258 201	0x00000000000144c0	libfontconfig.so.1.3.0
4.55	1.92	42 376	XStringToKeysym	libX11.so.6.2.0
4.05	1.48	105 528	malloc	libc-2.10.1.so: malloc.c
2.75	0.84	43 842	0x0000000000008e60	libfontconfig.so.1.3.0
2.95	0.67	1 924	dl_map_object	ld-2.10.1.so: dl-load.c
4.14	0.65	7 261	dl_lookup_symbol_x	ld-2.10.1.so: dl-lookup.c
2.60	0.42	92	dl_relocate_object	ld-2.10.1.so: dl-reloc.c, c
4.40	0.39	190 557	strcmp	libc-2.10.1.so: strcmp.S
3.36	0.36	20 083	XOpenLC	libX11.so.6.2.0
6.77	0.32	1 428	0x0000000000014ab0	libfontconfig.so.1.3.0
3.59	0.24	1	FcFontSetList	libfontconfig.so.1.3.0
3.26	0.20	10	FcConfigSubstituteWithPat	libfontconfig.so.1.3.0
3.36	0.19	20 091	XlcOpenConverter	libX11.so.6.2.0
3.60	0.18	1 222	dcigettext	libc-2.10.1.so: dcigettext
3.62	0.16	10 036	Xlcbmbstowcs	libX11.so.6.2.0
3.48	0.16	10 036	Xlcbmbstoutf8	libX11.so.6.2.0
6.81	0.15	1	0x0000000000038e6a0	libQtGui.so.4.5.2
3.31	0.04	1 118	0x0000000000014d30'2 <...>	libfftw3.so.3.2.3
6.88	0.03	3	FcFontSetMatch	libfontconfig.so.1.3.0
2.94	0.03	155	QLibrary::load() <cycle 14>	libQtCore.so.4.5.2
3.64	0.01	10 036	Xmbstowcs	libX11.so.6.2.0

```

13  // #include <QPushButton>
14
15
16  Dialog::Dialog() {
0.02  1 call(s) 'QMainWindow::QMainWindow(QWidget*, QFlags<Qt::WindowType>);' (libQtGui.so.4.5.2)
0.01  1 call(s) 'QPushButton::QPushButton(QWidget*)' (libQtGui.so.4.5.2)
0.00  2 call(s) '_dl_runtime_resolve' (ld-2.10.1.so: dl-trampoline.S)
0.00  1 call(s) 'QFlags<Qt::WindowType>::QFlags(void**)' (vad: qglobal.h)
0.00  1 call(s) 'mcount' (libc-2.10.1.so: _mcount.S)
17  0.00  jackcon = NULL;
18
19  0.00  scene = NULL;
20  0.00  view = NULL;
21
    
```

```

graph TD
    main["main  
23.68%"] -- 1x --> Dialog["Dialog::Dialog()  
23.68%"]
    Dialog -- 1x --> JackConn["JackConnection::JackConnection()"]
    Dialog -- 1x --> AddText["QGraphicsScene::addText(QString const&, QFont const&)"]
    
```

Valgrind - Probleme

- Programme laufen bis zu 100-mal langsamer ab.
- Das kann besonders bei echtzeitfähiger Software zu neuen Problemen führen.
- Überwältigende Zahl von Ausgaben vor allem durch externe Bibliotheken. Diese können durch sog. Suppression-Dateien unterdrückt werden.