



Informatik I Debugging

G. Zachmann
Clausthal University, Germany
zach@in.tu-clausthal.de



Der erste Computer-Bug

Photo # NH 96566-KN First Computer "Bug", 1945

9/9

0800 onban started
1000 stopped - onban ✓
1300 030 AP-AC ✓
030 PRO ✓ 2.13042045
check 2.13042045
Relays in 032 fault speed speed test
in relay 10.000 test.

1100 Started Cosine Tap (Sine cheat)
1525 Started Multi-Adder Test.

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

1700 changed stack.
1700 closed down.

First actual case of bug being found.



Grace Hopper
Admiral, US Navy



Folgen von Programmfehlern

- 1962 führte ein fehlender Bindestrich in einem Fortran-Programm zum Verlust der Venus-Sonde Mariner 1, welche über 80 Millionen US-Dollar gekostet hatte.
- Zwischen 1985 und 1987 gab es mehrere Unfälle mit dem medizinischen Bestrahlungsgerät Therac-25. Infolge einer Überdosis, die durch fehlerhafte Programmierung und fehlende Sicherungsmaßnahmen verursacht wurde, mussten Organe entfernt werden, ein Patient verstarb drei Wochen nach der Bestrahlung.
- 1999: NASA-Sonde *Climate Orbiter* verpasst den Landeanflug auf den Mars, weil die Programmierer das falsche Maßsystem verwendeten – Yard statt Meter. Die NASA verlor dadurch die mehrere hundert Millionen Dollar teure Sonde.



- 1996: Ariane-5 explodierte
 - Grund: Konvertierung einer zu großen Zahl von 64-bit float nach 16-bit int
- 1994: "Pentium-Bug"
 - Inkorrekte Division in manchen Fällen wegen unvollständiger Lookup-Tabellen
- Orkan "Lothar": keine Vorwarnung durch Wetterdienst
 - Grund: ignoriertes Ausreiser
- Millenium-Bridge in London:
 - Falsche Annahmen über Fußgänger-Verhalten → fehlerhafte Simulation



Programmfehler (oder Bugs)

- Ursache:
 - Programmierfehler
 - Fehler in der Laufzeitumgebung
 - Unvollständigkeit, Mehrdeutigkeiten, Fehler in der Spezifikation
- Eine Regel besagt: Ab einer bestimmten Größe enthalten Computerprogramme auch immer Programmfehler
- Arten von Programmfehlern:
 - Syntaxfehler
 - Laufzeitfehler
 - Designfehler

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 5

Syntaxfehler

- Syntaxfehler sind Verstöße gegen grammatische Regeln der Programmiersprache

```

N = 100
for i in range( 0, N )
    while (N % i) == 0:
        print i
        N = N / i
  
```

Doppelpunkt vergessen

- Solche Fehlern werden beim Versuch des Programmstarts von Python erkannt und gemeldet.

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 6

Laufzeitfehler (Semantische Fehler)

- Programme mit Syntaxfehlern lassen sich nicht starten.
- Semantische Fehler sind dagegen Fehler, die die Bedeutung des Programmcodes betreffen

```

a = "String"
print a[10]
  
```

a besteht nur aus 6 Zeichen

- Solche Fehler können das Programm zum Absturz bringen, das Programm friert ein (Endlosschleifen) oder verhält sich merkwürdig

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 7

Designfehler

- Designfehler sind Fehler im Grundkonzept einer Software

- Fehlerarten:
 - Fehler in der Anforderungsdefinition / Spezifikation
 - Das Programm tut nicht das, was der Kunde wollte
 - beruhen meist auf mangelnder Kenntnis des Fachgebietes, für das die Software entwickelt wird
 - Fehler im Softwaredesign
 - Das Programm läßt sich später nicht mehr verstehen / warten / verbessern
 - beruhen oft auf mangelnder Erfahrung des Softwareentwicklers

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 8

Vermeidung von Fehlern

- Je früher in einem Entwicklungsprozeß ein Fehler auftritt und je später er entdeckt wird, desto aufwendiger ist es, ihn zu beheben.
- Designfehler lassen sich oft durch gute Planung und Wahl eines geeigneten Softwaredesigns weitestgehend vermeiden
- Syntaktische Fehler erkennt meist der Compiler oder der Interpreter
- Beim Auffinden semantischer Fehler helfen (für bestimmte Fälle / Klassen von Fehlern) spezielle Programme

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 9

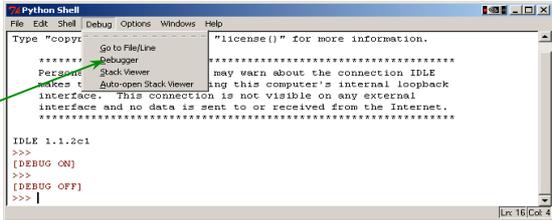
Debugger

- Nicht immer sind semantische Fehler so einfach zu finden wie in dem Beispiel oben
- Deswegen hilft ein spezielles Programm, der sogenannte Debugger, dem Programmierer bei der Fehlersuche
- Ein Debugger ermöglicht die Ablaufverfolgung des Programms in einzelnen Schritten
- Mit Python wird ein Debugger mitgeliefert

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 10

Python-Debugger

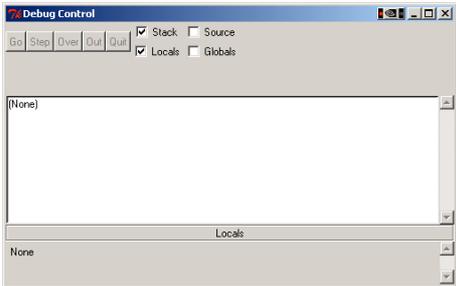
- Die Python-IDE Idle starten
- Das Menu „Debugger“ wählen
- Auf Menüpunkt „Debugger“ drücken



Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 11

Python-Debugger

- Das Debugger-Fenster wird geöffnet



Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 12

Python-Debugger

- Das betreffende Programm mit Idle öffnen

The screenshot shows the Python Shell application with the 'File' menu open. A green arrow points to the 'Open Module...' option, which is used to open a Python script.

- Und starten

The screenshot shows the Python Shell application with a script named 'debuggingexample.py' open. A green arrow points to the 'Run Module' button, which is used to execute the script.

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 13

Python-Debugger

- Im Debugger-Fenster sieht man den Stack, sowie die Werte der lokalen und globalen Variablen

The screenshot shows the 'Debug Control' window in the Python debugger. It displays the current execution state, including the stack and the values of local and global variables. The stack shows the current function call, and the locals and globals sections show the current state of the variables.

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 14

Python-Debugger

- Im Sourcecode-Fenster wird die aktuelle Position, an der der Debugger das Programm gerade hält, angezeigt

The screenshot shows the 'Sourcecode' window in the Python debugger. It displays the source code of the script being debugged. A blue highlight indicates the current position of the debugger, which is at the start of the 'for i in range(1, 20):' loop.

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 15

Python-Debugger

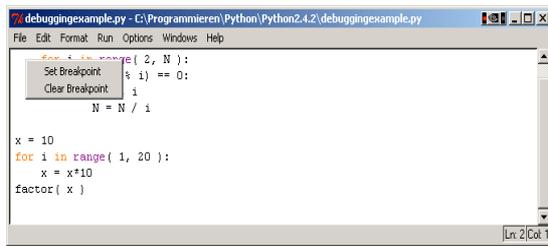
- Mit dem Step-Button kann man das Programm schrittweise weiter ausführen

The screenshot shows the 'Debug Control' window and the 'Sourcecode' window. A green arrow points to the 'Step' button in the 'Debug Control' window, which is used to execute the program one line at a time. Another green arrow points to the source code in the 'Sourcecode' window, showing the current position of the debugger.

Prof. Dr. G. Zachmann Informatik 1 - WS 05/06 Debugging 16

Python-Debugger

- Wenn man nur an bestimmten Stellen des Source interessiert ist kann man per Rechtsklick im Sourcecode-Fenster einen **Breakpoint** setzen und mittels **F5**-Taste im Debugger-Fenster bis zu dieser Stelle laufen lassen



The screenshot shows a Python IDE window titled 'debuggingexample.py - C:\Programmieren\Python\Python2.4.2\debuggingexample.py'. The code editor contains the following Python code:

```
for i in range( 2, N ):
    Set Breakpoint ( i ) == 0:
    Clear Breakpoint
    i
    N = N / i

x = 10
for i in range( 1, 20 ):
    x = x*10
factor( x )
```

A context menu is open over the first line of the code, with 'Set Breakpoint' selected. The status bar at the bottom right of the editor shows 'Ln: 2 | Col: 1'.