

Info I – „Übungsblatt 13“

Joachim Breitner

<http://www.joachim-breitner.de/wiki/Infotut>

13. Februar 2006





Unser Programm heute



- 1 Organisatorisches
- 2 Freie Software
 - Was ist falsch mit proprietärer Software
 - Was ist freie Software?
 - Toll, will ich haben!
- 3 Hackerlore
 - Murphy's Law
 - Hacker
 - Bugs
 - 42 und 23



1 Organisatorisches

2 Freie Software

- Was ist falsch mit proprietärer Software
- Was ist freie Software?
- Toll, will ich haben!

3 Hackerlore

- Murphy's Law
- Hacker
- Bugs
- 42 und 23



Übungsblatt-Rückblick



Statistik

- Schnitt: 29 von 40 Punkten
- Zwei müssen in der Rechnerübung noch Punkte holen

Häufige Fehler

- $\exists x$ obwohl x in der Aufgabenstellung gegeben ist.
- Vergessen, beim Bruder oder Onkel auf Ungleichheit zu überprüfen. Die Musterlösung verwendet \neq , also ist das anscheinend ok...
- $p \vee (p \wedge q) = p$



Übungsblatt-Rückblick



Statistik

- Schnitt: 29 von 40 Punkten
- Zwei müssen in der Rechnerübung noch Punkte holen

Häufige Fehler

- $\exists x$ obwohl x in der Aufgabenstellung gegeben ist.
- Vergessen, beim Bruder oder Onkel auf Ungleichheit zu überprüfen. Die Musterlösung verwendet \neq , also ist das anscheinend ok. . .
- $p \vee (p \wedge q) = p$



Übungsblatt-Zusammenfassung



Tutorium Benjamin Vogel:

`http://www.stud.uni-karlsruhe.de/~uxasq/info1tut9/
index.php?downloads`



1 Organisatorisches

2 Freie Software

- Was ist falsch mit proprietärer Software
- Was ist freie Software?
- Toll, will ich haben!

3 Hackerlore

- Murphy's Law
- Hacker
- Bugs
- 42 und 23



Sony verteilt Rootkits



Im Jahr 2005 war Sony BMG das erste Unternehmen, welches Rootkits in Form der DRM-Software XCP für das digitale Rechtemanagement einsetzte. Rootkits werden von Programmierern von Mal- und Spyware verwendet, um ihre illegalen Aktivitäten zu verbergen. Sony nutzt damit die Technologien von Kriminellen, um seine Profite zu sichern, und macht sich somit wegen Datenveränderung (§303a StGB) und Computersabotage (§303b StGB) strafbar. Der nur auf Nachfrage angebotene Deinstaller verursacht weitere Sicherheitslücken im System des Anwenders. Sony hat inzwischen entschieden, die CDs aus dem Handel zu nehmen und die etwa zwei Millionen verkauften CDs umzutauschen. Die entsprechende Software für den Mac, MediaMax von SunnComm, mit ähnlichen Rootkit-Eigenschaften, soll weiterhin verkauft werden.

Wikipedia, Artikel „Sony BMG“





Was lief da schief?



- Sony verteilte Software, die die Kunden verwenden mussten
- Niemand hatte die Möglichkeit zu prüfen, was diese Software genau macht
- Wenn man Sony nicht traut, gibt es keine Alternative zu ihrer Software
- Man weiß auch nicht, ob man seine Musik in fünf Jahren noch hören kann

... und das war nicht der einzige Fall. (NSA-Cryptoschlüssel in Windows NT, Upgrade-Zwang durch geheime Dateiformate, etc.)



Was fordern wir also?

Unser Leben wird mehr und mehr von Software diktiert. Daher fordern wir:

- Freiheit 0: Das Programm zu jedem Zweck auszuführen.
- Freiheit 1: Das Programm zu studieren und zu verändern.
- Freiheit 2: Das Programm zu kopieren.
- Freiheit 3: Das Programm zu verbessern und zu verbreiten, um damit einen Nutzen für die Gemeinschaft zu erzeugen.



Und wann brauch ich das?



- Mir passt was am Programm nicht.
Beispiel: ContextFree nutzt den Bildschirmraum nicht aus.
- Mir fehlt eine Funktion am Programm.
Beispiel: ICQ soll bei Nachrichten mit dem ThinkLight blinken.
- Ich traue einem Programm nicht.
Beispiel: Ist die e-Mail-Verschlüsselung fehlerfrei.
- Ich will wissen wie das funktioniert.
Beispiel: Welcher Algorithmus steckt im Bildschirmschoner.
- Ich will das Programm neu kompilieren.
Beispiel: Denn ich habe keinen i386-Prozessor.
- Meine Freunde sollen das auch benutzen dürfen.
Beispiel: Ich kopiere es ihnen einfach
- Ich will die Wahl haben.
Beispiel: KDE vs. GNOME, vim vs. EMACS, Linux vs. BSD. ...



Und wann brauch ich das?



- Mir passt was am Programm nicht.
Beispiel: ContextFree nutzt den Bildschirmraum nicht aus.
- Mir fehlt eine Funktion am Programm.
Beispiel: ICQ soll bei Nachrichten mit dem ThinkLight blinken.
- Ich traue einem Programm nicht.
Beispiel: Ist die e-Mail-Verschlüsselung fehlerfrei.
- Ich will wissen wie das funktioniert.
Beispiel: Welcher Algorithmus steckt im Bildschirmschoner.
- Ich will das Programm neu kompilieren.
Beispiel: Denn ich habe keinen i386-Prozessor.
- Meine Freunde sollen das auch benutzen dürfen.
Beispiel: Ich kopiere es ihnen einfach
- Ich will die Wahl haben.
Beispiel: KDE vs. GNOME, vim vs. EMACS, Linux vs. BSD...



Und wann brauch ich das?



- Mir passt was am Programm nicht.
Beispiel: ContextFree nutzt den Bildschirmraum nicht aus.
- Mir fehlt eine Funktion am Programm.
Beispiel: ICQ soll bei Nachrichten mit dem ThinkLight blinken.
- Ich traue einem Programm nicht.
Beispiel: Ist die e-Mail-Verschlüsselung fehlerfrei.
- Ich will wissen wie das funktioniert.
Beispiel: Welcher Algorithmus steckt im Bildschirmschoner.
- Ich will das Programm neu kompilieren.
Beispiel: Denn ich habe keinen i386-Prozessor.
- Meine Freunde sollen das auch benutzen dürfen.
Beispiel: Ich kopiere es ihnen einfach
- Ich will die Wahl haben.
Beispiel: KDE vs. GNOME, vim vs. EMACS, Linux vs. BSD. ...



Und wann brauch ich das?



- Mir passt was am Programm nicht.
Beispiel: ContextFree nutzt den Bildschirmraum nicht aus.
- Mir fehlt eine Funktion am Programm.
Beispiel: ICQ soll bei Nachrichten mit dem ThinkLight blinken.
- Ich traue einem Programm nicht.
Beispiel: Ist die e-Mail-Verschlüsselung fehlerfrei.
- Ich will wissen wie das funktioniert.
Beispiel: Welcher Algorithmus steckt im Bildschirmschoner.
- Ich will das Programm neu kompilieren.
Beispiel: Denn ich habe keinen i386-Prozessor.
- Meine Freunde sollen das auch benutzen dürfen.
Beispiel: Ich kopiere es ihnen einfach
- Ich will die Wahl haben.
Beispiel: KDE vs. GNOME, vim vs. EMACS, Linux vs. BSD...



Und wann brauch ich das?



- Mir passt was am Programm nicht.
Beispiel: ContextFree nutzt den Bildschirmraum nicht aus.
- Mir fehlt eine Funktion am Programm.
Beispiel: ICQ soll bei Nachrichten mit dem ThinkLight blinken.
- Ich traue einem Programm nicht.
Beispiel: Ist die e-Mail-Verschlüsselung fehlerfrei.
- Ich will wissen wie das funktioniert.
Beispiel: Welcher Algorithmus steckt im Bildschirmschoner.
- Ich will das Programm neu kompilieren.
Beispiel: Denn ich habe keinen i386-Prozessor.
- Meine Freunde sollen das auch benutzen dürfen.
Beispiel: Ich kopiere es ihnen einfach
- Ich will die Wahl haben.
Beispiel: KDE vs. GNOME, vim vs. EMACS, Linux vs. BSD. . .



Und wann brauch ich das?



- Mir passt was am Programm nicht.
Beispiel: ContextFree nutzt den Bildschirmraum nicht aus.
- Mir fehlt eine Funktion am Programm.
Beispiel: ICQ soll bei Nachrichten mit dem ThinkLight blinken.
- Ich traue einem Programm nicht.
Beispiel: Ist die e-Mail-Verschlüsselung fehlerfrei.
- Ich will wissen wie das funktioniert.
Beispiel: Welcher Algorithmus steckt im Bildschirmschoner.
- Ich will das Programm neu kompilieren.
Beispiel: Denn ich habe keinen i386-Prozessor.
- Meine Freunde sollen das auch benutzen dürfen.
Beispiel: Ich kopiere es ihnen einfach
- Ich will die Wahl haben.
Beispiel: KDE vs. GNOME, vim vs. EMACS, Linux vs. BSD. . .



Und wann brauch ich das?



- Mir passt was am Programm nicht.
Beispiel: ContextFree nutzt den Bildschirmraum nicht aus.
- Mir fehlt eine Funktion am Programm.
Beispiel: ICQ soll bei Nachrichten mit dem ThinkLight blinken.
- Ich traue einem Programm nicht.
Beispiel: Ist die e-Mail-Verschlüsselung fehlerfrei.
- Ich will wissen wie das funktioniert.
Beispiel: Welcher Algorithmus steckt im Bildschirmschoner.
- Ich will das Programm neu kompilieren.
Beispiel: Denn ich habe keinen i386-Prozessor.
- Meine Freunde sollen das auch benutzen dürfen.
Beispiel: Ich kopiere es ihnen einfach
- Ich will die Wahl haben.
Beispiel: KDE vs. GNOME, vim vs. EMACS, Linux vs. BSD. . .



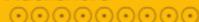
Community



Freie Software lebt von einer Entwickler-Gemeinder, der Community. Diese ist meist sehr offen und hört auf die Wünsche der Anwender. Es gibt oft

- Mailinglisten für User und Entwickler
- Offene Bug Tracking Systeme
- Zugriff auf die Quellen per CVS oder svn

Als Anwender hat man wesentlich eher die Chance, dass Wünsche erhöht werden, Bugreports sind willkommen und man kann mitmachen, mit wenig oder auch mit viel Einsatz.



Vorurteile über freie Software



- Freie Software ist nicht benutzerfreundlich.
- Freie Software gibt es nur für Linux.
- Mit Freier Software kann man kein Geld machen.
- Freie Software ist von Hackern geschrieben und gefährlich.
- Von Freier Software hat man nix, wenn man nicht programmieren kann.
- Freie Software ist zweitklassig.



Tolle Sache, wo bekomme ich das her?



- Die Einsteigervariante: Man verwendet Freie Software unter Windows:
 - Mozilla Firefox & Thunderbird (<http://getfirefox.com/>)
 - OpenOffice (<http://www.openoffice.org/>)
 - Miranda (<http://miranda-im.org/>)
- Die fortgeschrittene Variante: Man installiert sich ein Linux:
 - Für mehr Komfort: <http://www.ubuntu.com/>
 - Für mehr von allem: <http://www.debian.org/>
 - Für mehr Bastelei: <http://www.gentoo.org/>
- Für die, die alles anders machen wollen:
 - BSD, z.B. <http://www.freebsd.org/>
 - Linux From Scratch <http://www.linuxfromscratch.org/>
 - Hurd <http://www.gnu.org/software/hurd/hurd.html/>



Tolle Sache, wo bekomme ich das her?



- Die Einsteigervariante: Man verwendet Freie Software unter Windows:
 - Mozilla Firefox & Thunderbird (<http://getfirefox.com/>)
 - OpenOffice (<http://www.openoffice.org/>)
 - Miranda (<http://miranda-im.org/>)
- Die fortgeschrittene Variante: Man installiert sich ein Linux:
 - Für mehr Komfort: <http://www.ubuntu.com/>
 - Für mehr von allem: <http://www.debian.org/>
 - Für mehr Bastelei: <http://www.gentoo.org/>
- Für die, die alles anders machen wollen:
 - BSD, z.B. <http://www.freebsd.org/>
 - Linux From Scratch <http://www.linuxfromscratch.org/>
 - Hurd <http://www.gnu.org/software/hurd/hurd.html/>



Tolle Sache, wo bekomme ich das her?



- Die Einsteigervariante: Man verwendet Freie Software unter Windows:
 - Mozilla Firefox & Thunderbird (<http://getfirefox.com/>)
 - OpenOffice (<http://www.openoffice.org/>)
 - Miranda (<http://miranda-im.org/>)
- Die fortgeschrittene Variante: Man installiert sich ein Linux:
 - Für mehr Komfort: <http://www.ubuntu.com/>
 - Für mehr von allem: <http://www.debian.org/>
 - Für mehr Bastelei: <http://www.gentoo.org/>
- Für die, die alles anders machen wollen:
 - BSD, z.B. <http://www.freebsd.org/>
 - Linux From Scratch <http://www.linuxfromscratch.org/>
 - Hurd <http://www.gnu.org/software/hurd/hurd.html/>



1 Organisatorisches

2 Freie Software

- Was ist falsch mit proprietärer Software
- Was ist freie Software?
- Toll, will ich haben!

3 Hackerlore

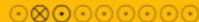
- Murphy's Law
- Hacker
- Bugs
- 42 und 23



Murphy's Law



„Anything that can go wrong will go wrong.“

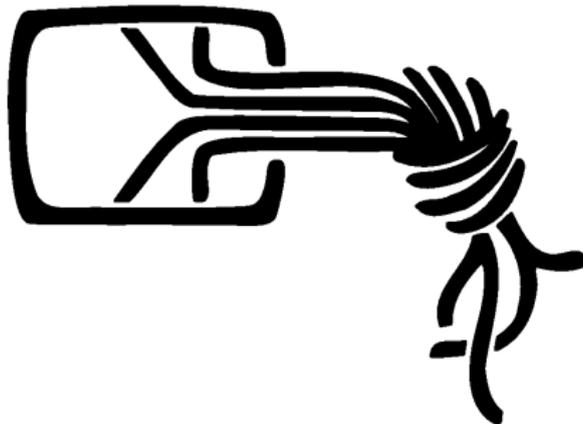


Was ist ein Hacker?



„Ein Hacker ist ein Mensch, der mit seiner Kaffeemaschine auch eine Suppe kochen kann.“

Wau Holland, CCC-Gründer





Hackerethik (CCC-Version)



- Der Zugang zu Computern und allem, was einem zeigen kann, wie diese Welt funktioniert, sollte unbegrenzt und vollständig sein.
- Alle Informationen müssen frei sein.
- Misstrauere Autoritäten – fördere Dezentralisierung
- Beurteile einen Hacker nach dem, was er tut und nicht nach üblichen Kriterien wie Aussehen, Alter, Abstammung, Geschlecht oder gesellschaftlicher Stellung.
- Man kann mit einem Computer Kunst und Schönheit schaffen.
- Computer können dein Leben zum Besseren verändern
- Mülle nicht in den Daten anderer Leute
- Öffentliche Daten nützen, private Daten schützen

Der erste Computer-Bugs



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

9/2

9/9

0800 Antam started
 1000 " stopped - antam ✓
 13⁰⁰ MC (032) MP-MC ~~1.582649000~~ 1.582649000
 (033) PRO 2 2.130476415
 conch 2.130676415

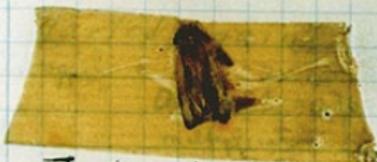
{ 1.2700 9.037 847 025
 9.037 846 995 conch
 4.615925059(-2)

Relays 6-2 in 033 failed special speed test
 in relay " 11.00 test "

Relay
 2145
 Relay 3376

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

1545



Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.

1630 Antam started.
 1700 closed down.



Aber dann:



It has been just so in all of my inventions. The first step is an intuition, and comes with a burst, then difficulties arise—this thing gives out and [it is] then that „Bugs“—as such little faults and difficulties are called—show themselves and months of intense watching, study and labor are requisite before commercial success or failure is certainly reached.

Source: Edison to Puskas, 13 November 1878, Edison papers, Edison National Laboratory, U.S. National Park Service, West Orange, N.J., cited in Thomas P. Hughes, American Genesis: A History of the American Genius for Invention, Penguin Books, 1989, on page 75.



Und auch noch gut zu wissen:



Heisenbugs

Frei nach der Unschärferelation von Werner Heisenberg: Bugs die nur dann auftreten, wenn man keinen Debugger laufen lässt (siehe auch Murphys Gesetz).

Donald Knuths Bugs

„Beware of bugs in the above code; I have only proved it correct, not tried it.“



Die Antwort



42 ist die Antwort auf die Frage nach dem Leben, dem Universum
und dem ganzen Rest.

Das Problem ist, wir kennen die Frage eigentlich garnicht.



Die Antwort



42 ist die Antwort auf die Frage nach dem Leben, dem Universum
und dem ganzen Rest.

Das Problem ist, wir kennen die Frage eigentlich garnicht.



Die Frage



Arthur Dent, als letztes lebendes Element des Großrechners „Erde“, welcher die Frage finden sollte, legt beim Scrabble diese Frage aus:

„Wieviel ist neun multipliziert mit sechs.“

W·T·F

Vielleicht: $9_{13} \cdot 6_{13} = 54_{10} = 4 \cdot 13 + 2 \cdot 1 = 42_{13}$

„Die Antwort darauf ist ganz einfach. Es war ein Scherz. Es musste eine Zahl sein, eine normale, kleine Zahl, und ich wählte diese. Binäre Darstellungen, Basis 13, Tibetanische Mönche, das ist alles kompletter Unsinn. Ich saß an meinem Schreibtisch, sah in den Garten und dachte ‚42 geht‘. Ich schrieb es. Ende der Geschichte.“

Douglas Adams, 1993



Die Frage



Arthur Dent, als letztes lebendes Element des Großrechners „Erde“, welcher die Frage finden sollte, legt beim Scrabble diese Frage aus:

„Wieviel ist neun multipliziert mit sechs.“

W.T.F

Vielleicht: $9_{13} \cdot 6_{13} = 54_{10} = 4 \cdot 13 + 2 \cdot 1 = 42_{13}$

„Die Antwort darauf ist ganz einfach. Es war ein Scherz. Es musste eine Zahl sein, eine normale, kleine Zahl, und ich wählte diese. Binäre Darstellungen, Basis 13, Tibetanische Mönche, das ist alles kompletter Unsinn. Ich saß an meinem Schreibtisch, sah in den Garten und dachte ‚42 geht‘. Ich schrieb es. Ende der Geschichte.“

Douglas Adams, 1993



Die Frage



Arthur Dent, als letztes lebendes Element des Großrechners „Erde“, welcher die Frage finden sollte, legt beim Scrabble diese Frage aus:

„Wieviel ist neun multipliziert mit sechs.“

W·T·F

Vielleicht: $9_{13} \cdot 6_{13} = 54_{10} = 4 \cdot 13 + 2 \cdot 1 = 42_{13}$

„Die Antwort darauf ist ganz einfach. Es war ein Scherz. Es musste eine Zahl sein, eine normale, kleine Zahl, und ich wählte diese. Binäre Darstellungen, Basis 13, Tibetanische Mönche, das ist alles kompletter Unsinn. Ich saß an meinem Schreibtisch, sah in den Garten und dachte ‚42 geht‘. Ich schrieb es. Ende der Geschichte.“

Douglas Adams, 1993



Die Frage



Arthur Dent, als letztes lebendes Element des Großrechners „Erde“, welcher die Frage finden sollte, legt beim Scrabble diese Frage aus:

„Wieviel ist neun multipliziert mit sechs.“

W·T·F

Vielleicht: $9_{13} \cdot 6_{13} = 54_{10} = 4 \cdot 13 + 2 \cdot 1 = 42_{13}$

„Die Antwort darauf ist ganz einfach. Es war ein Scherz. Es musste eine Zahl sein, eine normale, kleine Zahl, und ich wählte diese. Binäre Darstellungen, Basis 13, Tibetanische Mönche, das ist alles kompletter Unsinn. Ich saß an meinem Schreibtisch, sah in den Garten und dachte ‚42 geht‘. Ich schrieb es. Ende der Geschichte.“

Douglas Adams, 1993



Dreiunszwanzig



Karl Koch war ein 19-jähriger Hacker in den 80er Jahren. Berühmt wurde er durch den sogenannten „KGB-Hack“. Karl gründete den Chaos-Computer-Club-Ableger in Hannover.

Karl war begeistert von dem Roman „Illuminatus!“ von Robert Anton Wilson und war überzeugt von einer Verschwörung der Illuminaten und ihrer heiligen Zahl 23 sowie deren Quersumme 5. Er sah sich in der Rolle des Helden „Hagbard Celine“.

Die Geschichte von Karl Koch und gewisse Ungereimtheiten darin üben eine große Faszination auf die Hackerkultur aus. Der Film „23 – nichts ist so wie es scheint“ sollte daher gesehen werden. Karl starb unter ungeklärten Umständen am 23.5.1989 im Alter von 23 Jahren.



Dreiunszwanzig



Karl Koch war ein 19-jähriger Hacker in den 80er Jahren. Berühmt wurde er durch den sogenannten „KGB-Hack“. Karl gründete den Chaos-Computer-Club-Ableger in Hannover.

Karl war begeistert von dem Roman „Illuminatus!“ von Robert Anton Wilson und war überzeugt von einer Verschwörung der Illuminaten und ihrer heiligen Zahl 23 sowie deren Quersumme 5. Er sah sich in der Rolle des Helden „Hagbard Celine“.

Die Geschichte von Karl Koch und gewisse Ungereimtheiten darin üben eine große Faszination auf die Hackerkultur aus. Der Film „23 – nichts ist so wie es scheint“ sollte daher gesehen werden. Karl starb unter ungeklärten Umständen am 23.5.1989 im Alter von 23 Jahren.



Dreiunszwanzig



Karl Koch war ein 19-jähriger Hacker in den 80er Jahren. Berühmt wurde er durch den sogenannten „KGB-Hack“. Karl gründete den Chaos-Computer-Club-Ableger in Hannover.

Karl war begeistert von dem Roman „Illuminatus!“ von Robert Anton Wilson und war überzeugt von einer Verschwörung der Illuminaten und ihrer heiligen Zahl 23 sowie deren Quersumme 5. Er sah sich in der Rolle des Helden „Hagbard Celine“.

Die Geschichte von Karl Koch und gewisse Ungereimtheiten darin üben eine große Faszination auf die Hackerkultur aus. Der Film „23 – nichts ist so wie es scheint“ sollte daher gesehen werden.

Karl starb unter ungeklärten Umständen am 23.5.1989 im Alter von 23 Jahren.



Dreiunszwanzig



Karl Koch war ein 19-jähriger Hacker in den 80er Jahren. Berühmt wurde er durch den sogenannten „KGB-Hack“. Karl gründete den Chaos-Computer-Club-Ableger in Hannover.

Karl war begeistert von dem Roman „Illuminatus!“ von Robert Anton Wilson und war überzeugt von einer Verschwörung der Illuminaten und ihrer heiligen Zahl 23 sowie deren Quersumme 5. Er sah sich in der Rolle des Helden „Hagbard Celine“.

Die Geschichte von Karl Koch und gewisse Ungereimtheiten darin üben eine große Faszination auf die Hackerkultur aus. Der Film „23 – nichts ist so wie es scheint“ sollte daher gesehen werden. Karl starb unter ungeklärten Umständen am 23.5.1989 im Alter von 23 Jahren.

Viel Erfolg bei der Klausur
und
bis zum Sommersemester!

