

Institut für Bedeutungen

Universität Bunkenstedt

Die Bedeutung der Bedeutungen unter besonderer Berücksichtigung der Bedeutungen

Diplomarbeit

Max Mustermann

Matrikel-Nummer 12345678

Betreuer	Maxima Musterfrau
Erstprüfer	Prof. Musterus
Zweitprüfer	Prof. Mustera

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Symbolverzeichnis	VI
1 Einleitung	1
2 weiteres Kapitel	2
2.1 eine Sektion	2
2.1.1 jetzt geht es noch tiefer	3
3 Zusammenfassung	5
A Anhang	6
A.1 Quelltexte	6

Abbildungsverzeichnis

2.1	Test-Bild	3
-----	---------------------	---

Tabellenverzeichnis

2.1	eine sinnlose Tabelle	3
2.2	eine kompliziertere Tabelle	4

Symbolverzeichnis

Allgemeine Symbole

Symbol	Bedeutung
a	der Skalar a
\underline{x}	der Vektor \underline{x}
$\underline{\mathbf{A}}$	die Matrix $\underline{\mathbf{A}}$

1 Einleitung

Er hörte leise Schritte hinter sich. Das bedeutete nichts Gutes. Wer würde ihm schon folgen, spät in der Nacht und dazu noch in dieser engen Gasse mitten im übel beleumundeten Hafenviertel? Gerade jetzt, wo er das Ding seines Lebens gedreht hatte und mit der Beute verschwinden wollte! Hatte einer seiner zahllosen Kollegen dieselbe Idee gehabt, ihn beobachtet und abgewartet, um ihn nun um die Früchte seiner Arbeit zu erleichtern? Oder gehörten die Schritte hinter ihm zu einem der unzähligen Gesetzeshüter dieser Stadt, und die stählerne Acht um seine Handgelenke würde gleich zuschnappen? Er konnte die Aufforderung stehen zu bleiben schon hören. Gehetzt sah er sich um. Plötzlich erblickte er den schmalen Durchgang. Blitzartig drehte er sich nach rechts und verschwand zwischen den beiden Gebäuden. Beinahe wäre er dabei über den umgestürzten Mülleimer gefallen, der mitten im Weg lag. Er versuchte, sich in der Dunkelheit seinen Weg zu ertasten und erstarrte: Anscheinend gab es keinen anderen Ausweg aus diesem kleinen Hof als den Durchgang, durch den er gekommen war. Die Schritte wurden lauter und lauter, er sah eine dunkle Gestalt um die Ecke biegen. Fieberhaft irrten seine Augen durch die nächtliche Dunkelheit und suchten einen Ausweg. War jetzt wirklich alles vorbei, waren alle Mühe und alle Vorbereitungen umsonst? Er presste sich ganz eng an die Wand hinter ihm und hoffte, der Verfolger würde ihn übersehen, als plötzlich neben ihm mit kaum wahrnehmbarem Quietschen eine Tür im nächtlichen Wind hin und her schwang. Könnte dieses der flehentlich herbeigesehnte Ausweg aus seinem Dilemma sein? Langsam bewegte er sich auf die offene Tür zu, immer dicht an die Mauer gepresst. Würde diese Tür seine Rettung werden?

2 weiteres Kapitel

In diesem Kapitel wird einiges gemacht¹ Vor allem in Unterabschnitt 2.1.1 wird einiges gezeigt, was noch nie jemand gesehen hat. Es lohnt sich also, dranzubleiben.

2.1 eine Sektion

Er hörte leise Schritte hinter sich. Das bedeutete nichts Gutes. Wer würde ihm schon folgen, spät in der Nacht und dazu noch in dieser engen Gasse mitten im übel beleumundeten Hafenviertel? Gerade jetzt, wo er das Ding seines Lebens gedreht hatte und mit der Beute verschwinden wollte! Hatte einer seiner zahllosen Kollegen dieselbe Idee gehabt, ihn beobachtet und abgewartet, um ihn nun um die Früchte seiner Arbeit zu erleichtern? **TODO: das muss ich noch verfeinern, weil ich erst zur Hälfte verstanden habe** Oder gehörten die Schritte hinter ihm zu einem der unzähligen Gesetzeshüter dieser Stadt, und die stählerne Acht um seine Handgelenke würde gleich zuschnappen? Er konnte die Aufforderung stehen zu bleiben schon hören. Gehetzt sah er sich um. Plötzlich erblickte er den schmalen Durchgang. Blitzartig drehte er sich nach rechts und verschwand zwischen den beiden Gebäuden. Beinahe wäre er dabei über den umgestürzten Mülleimer gefallen, der mitten im Weg lag. Er versuchte, sich in der Dunkelheit seinen Weg zu ertasten und erstarrte[?]: Anscheinend gab es keinen anderen Ausweg aus diesem kleinen Hof als den Durchgang, durch den er gekommen war. Die Schritte wurden lauter und lauter, er sah eine dunkle Gestalt um die Ecke biegen. Fieberhaft irrten seine Augen durch die nächtliche Dunkelheit und suchten einen Ausweg. War jetzt wirklich alles vorbei, waren alle Mühe und alle Vorbereitungen umsonst? Er presste sich ganz eng an die Wand hinter ihm und hoffte, der Verfolger würde ihn übersehen, als plötzlich neben ihm mit kaum wahrnehmbarem Quietschen eine Tür im nächtlichen Wind hin und her schwang. Könnte dieses der flehentlich herbeigesehnte Ausweg aus seinem Dilemma sein? Langsam bewegte er sich auf die offene Tür zu, immer dicht an die Mauer gepresst. Würde diese Tür seine Rettung werden?

Die Gleichung

$$a^2 + b^2 = c^2 \tag{2.1}$$

ist allseits bekannt und bedarf wohl keiner weiteren Erläuterung.

¹ wobei einiges nicht vieles heißt, ich möchte hier also keine falschen Hoffnungen wecken.

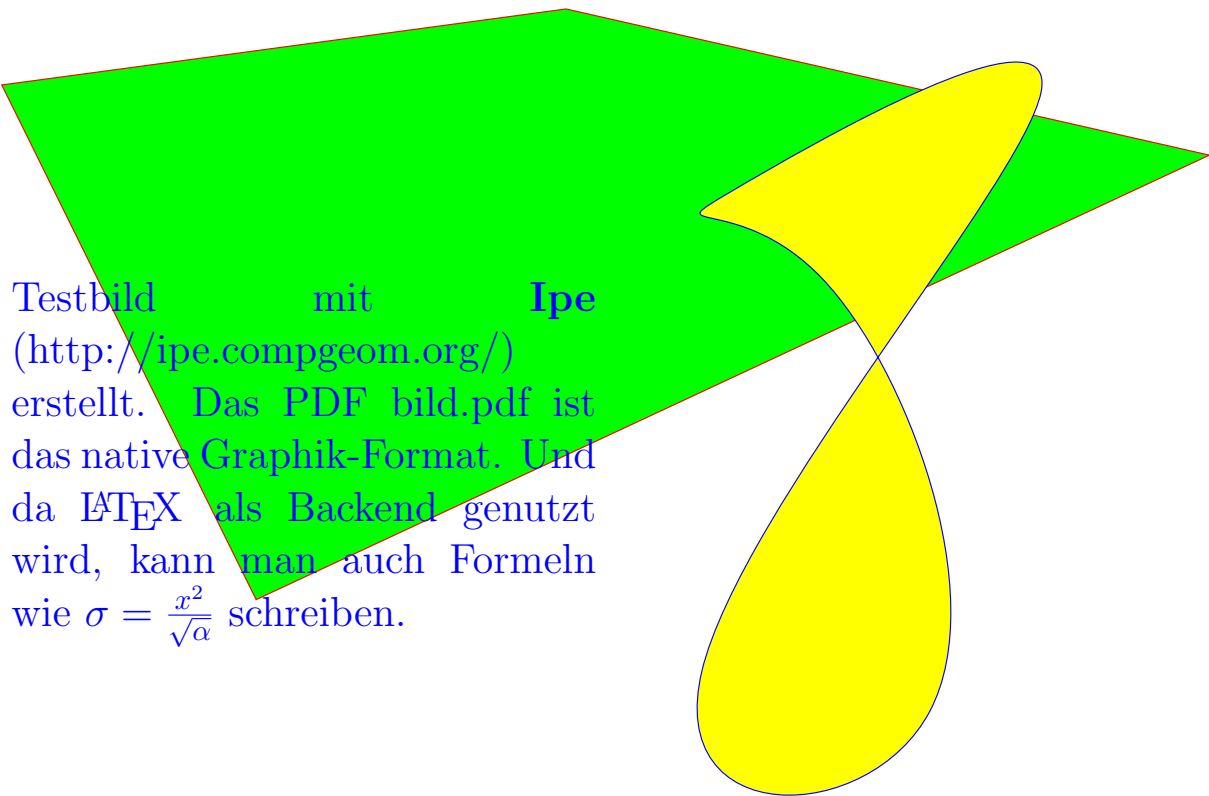


Abbildung 2.1: Test-Bild

Auch nicht schlecht ist Abbildung 2.1. Aber überhaupt keinen Sinn macht Tabelle 2.1. Hieran sieht man den Vorteil des autoref-Befehls und das so Links erstellt werden.

Formen	Städte
Quadrat	Bunkenstedt
Dreieck	Laggenbeck
Kreis	Peine
Raute	Wakaluba

Tabelle 2.1: eine sinnlose Tabelle

2.1.1 jetzt geht es noch tiefer

Er hörte leise Schritte hinter sich. Das bedeutete nichts Gutes. Wer würde ihm schon folgen, spät in der Nacht und dazu noch in dieser engen Gasse mitten im übel beleumundeten Hafenviertel? Gerade jetzt, wo er das Ding seines Lebens gedreht hatte und mit der Beute verschwinden wollte! Hatte einer seiner zahllosen Kollegen dieselbe Idee gehabt, ihn beobachtet und abgewartet, um ihn nun um die Früchte seiner Arbeit zu erleichtern? Oder gehörten die Schritte hinter ihm zu einem der

unzähligen Gesetzeshüter dieser Stadt, und die stählerne Acht um seine Handgelenke würde gleich zuschnappen? Er konnte die Aufforderung stehen zu bleiben schon hören. Gehetzt sah er sich um. Plötzlich erblickte er den schmalen Durchgang. Blitzartig drehte er sich nach rechts und verschwand zwischen den beiden Gebäuden. Beinahe wäre er dabei über den umgestürzten Mülleimer gefallen, der mitten im Weg lag. Er versuchte, sich in der Dunkelheit seinen Weg zu ertasten und erstarrte: Anscheinend gab es keinen anderen Ausweg aus diesem kleinen Hof als den Durchgang, durch den er gekommen war. Die Schritte wurden lauter und lauter, er sah eine dunkle Gestalt um die Ecke biegen. Fieberhaft irrten seine Augen durch die nächtliche Dunkelheit und suchten einen Ausweg. War jetzt wirklich alles vorbei, waren alle Mühe und alle Vorbereitungen umsonst? Er presste sich ganz eng an die Wand hinter ihm und hoffte, der Verfolger würde ihn übersehen, als plötzlich neben ihm mit kaum wahrnehmbarem Quietschen eine Tür im nächtlichen Wind hin und her schwang. Könnte dieses der flehentlich herbeigesehnte Ausweg aus seinem Dilemma sein? Langsam bewegte er sich auf die offene Tür zu, immer dicht an die Mauer gepresst. Würde diese Tür seine Rettung werden?

- Erstens ist das soundso,
- dann darf man natürlich nicht vergessen und
- das ist auch noch wichtig.

Komplexe Tabellen sind nicht sehr einfach:

		dies			
		von dort	und dort	über hier	zu Los
das	hier	bla	bla	bla	bla
	dort	bla	bla	bla	bla
	da	bla	bla	bla	bla

Tabelle 2.2: eine kompliziertere Tabelle mit viel Beschreibungstext, der aber nicht im Tabellenverzeichnis auftauschen soll

3 Zusammenfassung

Er hörte leise Schritte hinter sich. Das bedeutete nichts Gutes. Wer würde ihm schon folgen, spät in der Nacht und dazu noch in dieser engen Gasse mitten im übel beleumundeten Hafenviertel? Gerade jetzt, wo er das Ding seines Lebens gedreht hatte und mit der Beute verschwinden wollte! Hatte einer seiner zahllosen Kollegen dieselbe Idee gehabt, ihn beobachtet und abgewartet, um ihn nun um die Früchte seiner Arbeit zu erleichtern? Oder gehörten die Schritte hinter ihm zu einem der unzähligen Gesetzeshüter dieser Stadt, und die stählerne Acht um seine Handgelenke würde gleich zuschnappen? Er konnte die Aufforderung stehen zu bleiben schon hören. Gehetzt sah er sich um. Plötzlich erblickte er den schmalen Durchgang. Blitzartig drehte er sich nach rechts und verschwand zwischen den beiden Gebäuden. Beinahe wäre er dabei über den umgestürzten Mülleimer gefallen, der mitten im Weg lag. Er versuchte, sich in der Dunkelheit seinen Weg zu ertasten und erstarrte: Anscheinend gab es keinen anderen Ausweg aus diesem kleinen Hof als den Durchgang, durch den er gekommen war. Die Schritte wurden lauter und lauter, er sah eine dunkle Gestalt um die Ecke biegen. Fieberhaft irrten seine Augen durch die nächtliche Dunkelheit und suchten einen Ausweg. War jetzt wirklich alles vorbei, waren alle Mühe und alle Vorbereitungen umsonst? Er presste sich ganz eng an die Wand hinter ihm und hoffte, der Verfolger würde ihn übersehen, als plötzlich neben ihm mit kaum wahrnehmbarem Quietschen eine Tür im nächtlichen Wind hin und her schwang. Könnte dieses der flehentlich herbeigesehnte Ausweg aus seinem Dilemma sein? Langsam bewegte er sich auf die offene Tür zu, immer dicht an die Mauer gepresst. Würde diese Tür seine Rettung werden?

A Anhang

A.1 Quelltexte

cpu.c aus Linux 2.6.16

```
1  /* CPU control.
2  * (C) 2001, 2002, 2003, 2004 Rusty Russell
3  *
4  * This code is licenced under the GPL.
5  */
6  #include <linux/proc_fs.h>
7  #include <linux/smp.h>
8  #include <linux/init.h>
9  #include <linux/notifier.h>
10 #include <linux/sched.h>
11 #include <linux/unistd.h>
12 #include <linux/cpu.h>
13 #include <linux/module.h>
14 #include <linux/kthread.h>
15 #include <linux/stop_machine.h>
16 #include <asm/semaphore.h>
17
18 /* This protects CPUs going up and down... */
19 static DECLARE_MUTEX(cpucontrol);
20
21 static struct notifier_block *cpu_chain;
22
23 #ifdef CONFIG_HOTPLUG_CPU
24 static struct task_struct *lock_cpu_hotplug_owner;
25 static int lock_cpu_hotplug_depth;
26
27 static int __lock_cpu_hotplug(int interruptible)
28 {
29     int ret = 0;
30
31     if (lock_cpu_hotplug_owner != current) {
32         if (interruptible)
33             ret = down_interruptible(&cpucontrol);
```

```
34     else
35         down(&cpucontrol);
36     }
37
38     /*
39     * Set only if we succeed in locking
40     */
41     if (!ret) {
42         lock_cpu_hotplug_depth++;
43         lock_cpu_hotplug_owner = current;
44     }
45
46     return ret;
47 }
48
49 void lock_cpu_hotplug(void)
50 {
51     __lock_cpu_hotplug(0);
52 }
53 EXPORT_SYMBOL_GPL(lock_cpu_hotplug);
54
55 void unlock_cpu_hotplug(void)
56 {
57     if (--lock_cpu_hotplug_depth == 0) {
58         lock_cpu_hotplug_owner = NULL;
59         up(&cpucontrol);
60     }
61 }
62 EXPORT_SYMBOL_GPL(unlock_cpu_hotplug);
63
64 int lock_cpu_hotplug_interruptible(void)
65 {
66     return __lock_cpu_hotplug(1);
67 }
68 EXPORT_SYMBOL_GPL(lock_cpu_hotplug_interruptible);
69 #endif /* CONFIG_HOTPLUG_CPU */
70
71 /* Need to know about CPUs going up/down? */
72 int register_cpu_notifier(struct notifier_block *nb)
73 {
74     int ret;
75
76     if ((ret = lock_cpu_hotplug_interruptible()) != 0)
77         return ret;
78     ret = notifier_chain_register(&cpu_chain, nb);
79     unlock_cpu_hotplug();
80     return ret;
81 }
82 EXPORT_SYMBOL(register_cpu_notifier);
83
84 void unregister_cpu_notifier(struct notifier_block *nb)
85 {
86     lock_cpu_hotplug();
87     notifier_chain_unregister(&cpu_chain, nb);
88     unlock_cpu_hotplug();
```

```

89 }
90 EXPORT_SYMBOL(unregister_cpu_notifier);
91
92 #ifndef CONFIG_HOTPLUG_CPU
93 static inline void check_for_tasks(int cpu)
94 {
95     struct task_struct *p;
96
97     write_lock_irq(&tasklist_lock);
98     for_each_process(p) {
99         if (task_cpu(p) == cpu &&
100             (!cputime_eq(p->utime, cputime_zero) ||
101              !cputime_eq(p->stime, cputime_zero)))
102             printk(KERN_WARNING "Task_%s_(pid=%d)_is_on_cpu_%d\n",
103                    (state_="%ld, flags_=%lx)\n",
104                     p->comm, p->pid, cpu, p->state, p->flags);
105     }
106     write_unlock_irq(&tasklist_lock);
107 }
108
109 /* Take this CPU down. */
110 static int take_cpu_down(void *unused)
111 {
112     int err;
113
114     /* Ensure this CPU doesn't handle any more interrupts. */
115     err = __cpu_disable();
116     if (err < 0)
117         return err;
118
119     /* Force idle task to run as soon as we yield: it should
120        immediately notice cpu is offline and die quickly. */
121     sched_idle_next();
122     return 0;
123 }
124
125 int cpu_down(unsigned int cpu)
126 {
127     int err;
128     struct task_struct *p;
129     cpumask_t old_allowed, tmp;
130
131     if ((err = lock_cpu_hotplug_interruptible()) != 0)
132         return err;
133
134     if (num_online_cpus() == 1) {
135         err = -EBUSY;
136         goto out;
137     }
138
139     if (!cpu_online(cpu)) {
140         err = -EINVAL;
141         goto out;
142     }
143

```

```

144     err = notifier_call_chain(&cpu_chain, CPU_DOWN_PREPARE,
145                              (void *) (long)cpu);
146     if (err == NOTIFY_BAD) {
147         printk("%s:_attempt_to_take_down_CPU_%u_failed\n",
148                __FUNCTION__, cpu);
149         err = -EINVAL;
150         goto out;
151     }
152
153     /* Ensure that we are not runnable on dying cpu */
154     old_allowed = current->cpus_allowed;
155     tmp = CPU_MASK_ALL;
156     cpu_clear(cpu, tmp);
157     set_cpus_allowed(current, tmp);
158
159     p = __stop_machine_run(take_cpu_down, NULL, cpu);
160     if (IS_ERR(p)) {
161         /* CPU didn't die: tell everyone. Can't complain. */
162         if (notifier_call_chain(&cpu_chain, CPU_DOWN_FAILED,
163                                (void *) (long)cpu) == NOTIFY_BAD)
164             BUG();
165
166         err = PTR_ERR(p);
167         goto out_allowed;
168     }
169
170     if (cpu_online(cpu))
171         goto out_thread;
172
173     /* Wait for it to sleep (leaving idle task). */
174     while (!idle_cpu(cpu))
175         yield();
176
177     /* This actually kills the CPU. */
178     __cpu_die(cpu);
179
180     /* Move it here so it can run. */
181     kthread_bind(p, get_cpu());
182     put_cpu();
183
184     /* CPU is completely dead: tell everyone. Too late to complain. */
185     if (notifier_call_chain(&cpu_chain, CPU_DEAD, (void *) (long)cpu)
186         == NOTIFY_BAD)
187         BUG();
188
189     check_for_tasks(cpu);
190
191 out_thread:
192     err = kthread_stop(p);
193 out_allowed:
194     set_cpus_allowed(current, old_allowed);
195 out:
196     unlock_cpu_hotplug();
197     return err;
198 }

```

```

199 #endif /*CONFIG_HOTPLUG_CPU*/
200
201 int __devinit cpu_up(unsigned int cpu)
202 {
203     int ret;
204     void *hcpu = (void *) (long)cpu;
205
206     if ((ret = lock_cpu_hotplug_interruptible()) != 0)
207         return ret;
208
209     if (cpu_online(cpu) || !cpu_present(cpu)) {
210         ret = -EINVAL;
211         goto out;
212     }
213
214     ret = notifier_call_chain(&cpu_chain, CPU_UP_PREPARE, hcpu);
215     if (ret == NOTIFY_BAD) {
216         printk("%s:_attempt_to_bring_up_CPU_%u_failed\n",
217             __FUNCTION__, cpu);
218         ret = -EINVAL;
219         goto out_notify;

```

```

220     }
221
222     /* Arch-specific enabling code. */
223     ret = __cpu_up(cpu);
224     if (ret != 0)
225         goto out_notify;
226     if (!cpu_online(cpu))
227         BUG();
228
229     /* Now call notifier in preparation. */
230     notifier_call_chain(&cpu_chain, CPU_ONLINE, hcpu);
231
232 out_notify:
233     if (ret != 0)
234         notifier_call_chain(&cpu_chain, CPU_UP_CANCELED, hcpu);
235 out:
236     unlock_cpu_hotplug();
237     return ret;
238 }

```

Literaturverzeichnis

[Wer99] WERANDERS, Hans: Der Titel ist seine Allegorie seiner selbst. In: *Bücher über dies und das* (1999), Februar, S. 257–286

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Ort, Datum

Unterschrift