

Computer-Code aktuell halten, automatisch und ohne Magie

Michele MARTONE

23.03.2025

Chemnitzer Linux Tage 2025

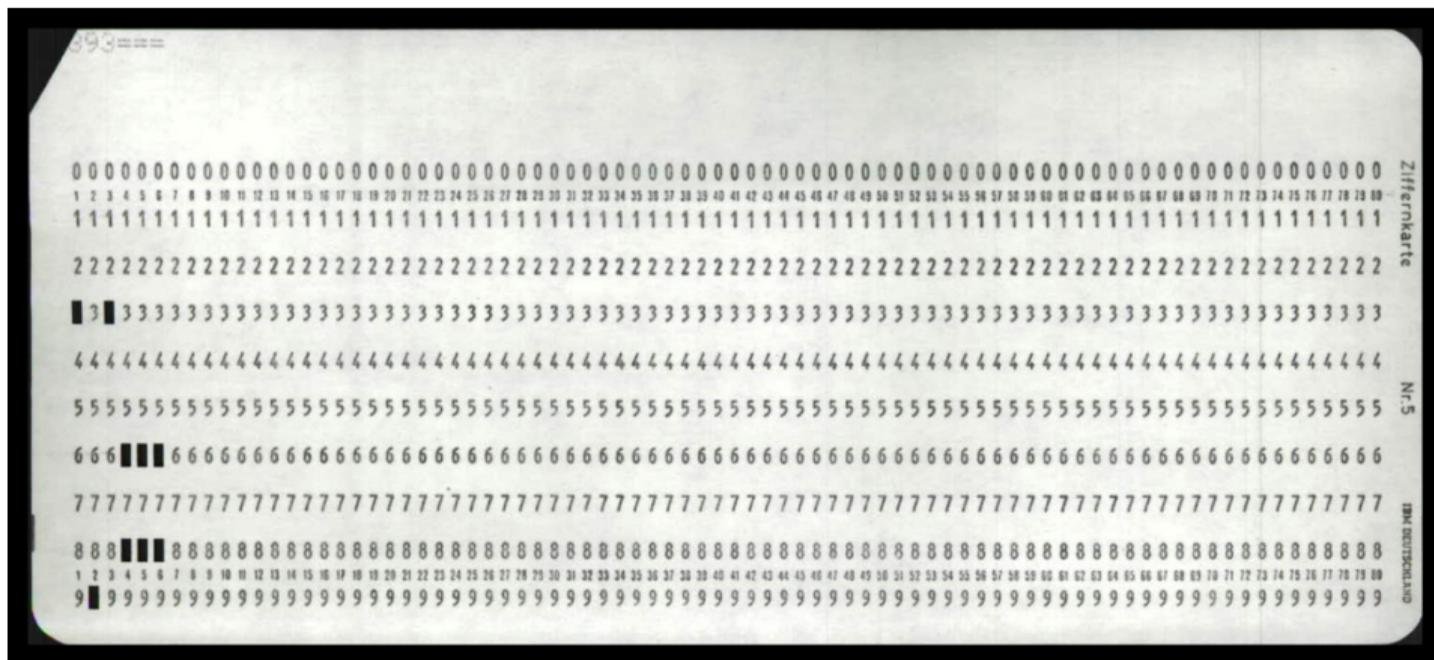


Lochkarte



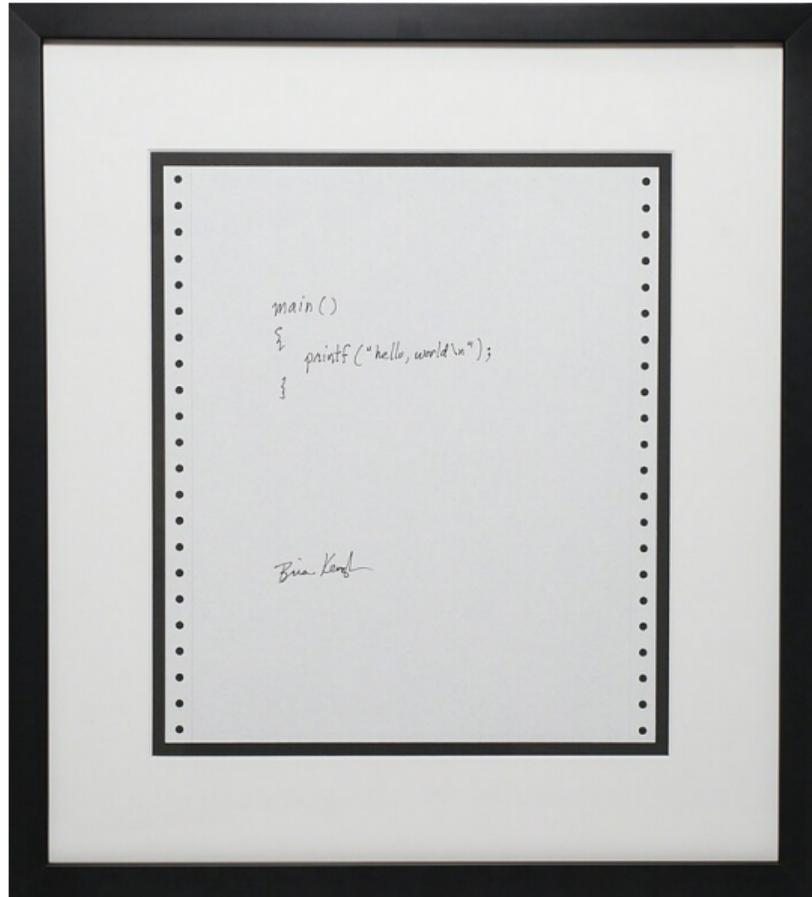
<https://www.punctum.com/interest/punch/html/c0224.de.html>

Lochkarte



<https://www.punctum.com/interest/punch/html/c0030.de.html>

“Hello, World!” C Programm, von Brian Kernighan (1978)



<https://www.artsy.net/artwork/brian-kernighan-hello-world>



Mehrere Wege zum selben Zweck

Bytes kopieren, einzeln:

```
$ dd if=/dev/zero of=/dev/shm/zeros bs=1 count=1024  
1024+0 records in  
1024+0 records out  
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.00377247 s, 271 kB/s
```

Mehrere Wege zum selben Zweck

Bytes kopieren, einzeln:

```
$ dd if=/dev/zero of=/dev/shm/zeros bs=1 count=1024
1024+0 records in
1024+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.00377247 s, 271 kB/s
```

Bytes kopieren, löffelweise:

```
$ dd if=/dev/zero of=/dev/shm/zeros bs=1024 count=1
1+0 records in
1+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.000196283 s, 5.2 MB/s
```

Größenvorteile (*Economy of Scale*) auf CPU-Ebene!



as Schifflin fliegt, der Webstuhl kracht,
Wir weben eifrig Tag und Nacht.

Heinrich Heine. Die schlesischen Weber, 1844

*“Das Schiffllein fliegt, der Webstuhl kracht,
Wir weben emsig Tag und Nacht.”*

Heinrich Heine. *Die schlesischen Weber*, 1844

*“Das Schiffllein fliegt, der Webstuhl kracht,
Wir weben **emsig** Tag und Nacht.”*

Heinrich Heine. *Die schlesischen Weber*, 1844

Aufgabe: seltene Wörter mit gängigeren ersetzen

Das Schifflein fliegt, der Webstuhl kracht,
Wir weben emsig Tag und Nacht.

Aufgabe: seltene Wörter mit gängigeren ersetzen

Das Schiffflein fliegt, der Webstuhl kracht,
Wir weben emsig Tag und Nacht.

Laut <https://corpora.uni-leipzig.de/> wir können
emsig mit **eifrig** ersetzen.

Aufgabe: seltene Wörter mit gängigeren ersetzen

Das Schiffflein fliegt, der Webstuhl kracht,
Wir weben emsig Tag und Nacht.

Laut <https://corpora.uni-leipzig.de/> wir können
emsig mit eifrig ersetzen.

Das Schiffflein fliegt, der Webstuhl kracht,
Wir weben eifrig Tag und Nacht.

Notation für Textunterschiede

Alter Text:

Das Schifflein fliegt, der Webstuhl kracht,
Wir weben emsig Tag und Nacht.

Das Schifflein fliegt, der Webstuhl kracht,
Wir weben eifrig Tag und Nacht.

Gibt es eine *knappe* Darstellung der Veränderung?

Notation für Textunterschiede

Alter Text:

```
Das Schifflein fliegt, der Webstuhl kracht,  
Wir weben emsig Tag und Nacht.
```

```
Das Schifflein fliegt, der Webstuhl kracht,  
Wir weben eifrig Tag und Nacht.
```

Vielleicht diese:

```
Das Schifflein fliegt, der Webstuhl kracht,  
Wir weben [-emsig-] {+eifrig+} Tag und Nacht.
```

Zusatz: das ist die wdiff Notation

```
$ wdiff alt.txt neu.txt  
...
```

Gut für

- ▶ zeigt Unterschied im Kontext
- ▶ die Korrektur einer langen Text offenkundig zu machen
- ▶ insbesondere mit Code, und aller Arten von Text
- ▶ Lesen des ganzen Text nicht mehr nötig
- ▶ Teil von GNU diffutils

Für mehr, siehe:

<https://www.gnu.org/software/wdiff/manual/wdiff.html>

Neues Gedankenspiel: fiktive Computersprache für Textkorrekturen z.B.

Alter Text:

```
Das Schifflein fliegt, der Webstuhl kracht,  
Wir weben emsig Tag und Nacht.
```

Textkorrekturbefehl:

```
[-emsig-] {+eifrig+}
```

Neues Gedankenspiel: fiktive Computersprache für Textkorrekturen z.B.

Alter Text:

```
Das Schifflein fliegt, der Webstuhl kracht,  
Wir weben emsig Tag und Nacht.
```

Textkorrekturbefehl:

```
[-emsig-] {+eifrig+}
```

Neuer Text:

```
Das Schifflein fliegt, der Webstuhl kracht,  
Wir weben eifrig Tag und Nacht.
```

Wort löschen?

“Lösche emsig”

`[-emsig-]`

Wort löschen, im Kontext

Nur nach “weben”

weben [-emsig-]

Wort ersetzen, im Kontext

weben [-emsig-] {+eifrig+}

Anzahl der Leerstellen ist unerheblich (so wollen wir)

Wir weben emsig Tag und Nacht

weben_ [-emsig-]_ {+eifrig+}

_ _ weben_ _ _ _ [-emsig-]_ _ _ _ {+eifrig+}

weben_ _ _ _ _ _ _ _ _ [-emsig-] {+eifrig+}

Wir weben [-emsig-] {+eifrig+} Tag und Nacht

Jedes Adverb löschen?

Vorausgesetzt dass:

- ▶ wir führen Schlüsselwörter ein: ADJEKTIV, ADVERB, ...
- ▶ jedes Adverb bekannt oder erkennbar ist

[- ADVERB -]

Jedes Adverb mit bestimmem Wort ersetzen?

“Ersetze jedes Adverb mit *eifrig*”

`[-ADVERB-] {+eifrig+}`

Selbe Sache, aber mit Kontext?

“Ersetze jedes Adverb nach *weben* mit *eifrig*”

weben [-ADVERB-] {+eifrig+}

Und hier?

Zwei Adverben vertauschen?

$[-ADVERB -] \{+ADVERB+\}$, $[-ADVERB -] \{+ADVERB+\}$

Und hier?

Zwei Adverben vertauschen?

$[-ADVERB -] \{+ADVERB+\}$, $[-ADVERB -] \{+ADVERB+\}$

Das ist problematisch, oder?

Wir führen Deklarationen von Variablen ein

“Vertausche aufeinander folgende Adverben”

@@

Sei A_1 ein Adverb.

Sei A_2 ein Adverb.

@@

$[-A_1 -]$ $\{+A_2+\}$,

$[-A_2 -]$ $\{+A_1+\}$

eifrig,

emsig

Wir führen Deklarationen von Variablen ein

“Vertausche aufeinander folgende Adverben”

@@

Sei $A1$ ein Adverb.

Sei $A2$ ein Adverb.

@@

$[-A1-]$ $\{+A2+\}$,

$[-A2-]$ $\{+A1+\}$

$[-eifrig-]$ $\{+emsig+\}$,

$[-emsig-]$ $\{+eifrig+\}$

Wir führen Deklarationen von Variablen ein

“Vertausche diese drei Worte mit jenen drei”

@@

Sei A1 ein Adverb.

Sei A2 ein Adverb.

@@

$[-A1-]$ $[-,-]$ $[-A2-]$

$\{+A2+\}$ $\{+,+\}$ $\{+A1+\}$

$[-eifrig-]$ $[-,-]$ $[-emsig-]$

$\{+emsig+\}$ $\{+,+\}$ $\{+eifrig+\}$

Wir führen Deklarationen von Variablen ein

“Vertausche diese Wortgruppe mit jener” – besser?

@@

Sei A1 ein Adverb.

Sei A2 ein Adverb.

@@

[-A1 , A2-]

{+A2 , A1+}

[-eifrig , emsig-]

{+emsig , eifrig+}

Vereinfachen wir die Notation: führen wir $-$ und $+$ Zeilen ein
“Lösche Ausdrücke aus dieser Zeile, und füge diese andere ein”

@@

Sei $A1$ ein Adverb.

Sei $A2$ ein Adverb.

@@

$-$ $A1$, $A2$

$+$ $A2$, $A1$

$-$ eifrig , emsig

$+$ emsig , eifrig

Eine Variable für einen ganzen Hauptsatz

“Lösche den letzten Hauptsatz”

```
@@  
Sei H ein Hauptsatz.  
@@  
- , H  
.
```

```
Das Schifflein fliegt  
, der Webstuhl kracht  
, Wir weben emsig Tag und Nacht  
.
```

Eine Variable für einen ganzen Hauptsatz

“Lösche den letzten Hauptsatz”

@@

Sei H ein Hauptsatz.

@@

- , H

.

Das Schifflein fliegt

, der Webstuhl kracht

- , Wir weben emsig Tag und Nacht

.

Zeit für eine kleine Prüfung

“Lösche... welchen Hauptsatz?”

@@

Sei **H** ein Hauptsatz.

@@

- , H

,

Das Schifflein fliegt
, der Webstuhl kracht
, Wir weben emsig Tag und Nacht
.

Zeit für eine kleine Prüfung

“Lösche... welchen Hauptsatz?”

@@

Sei H ein Hauptsatz.

@@

- , H

,

Das Schifflein fliegt

- , der Webstuhl kracht

, Wir weben emsig Tag und Nacht

.

Zusatz: Unterschied von zwei Textdateien aka *Diff*

```
$ diff -u alt.txt neu.txt
```

```
...
```

```
@@ -1,2 +1,2 @@
```

```
Das Schiffllein fliegt, der Webstuhl kracht,
```

```
-Wir weben emsig Tag und Nacht.
```

```
+Wir weben eifrig Tag und Nacht.
```

Zusatz: Unterschied von zwei Textdateien aka *Diff*

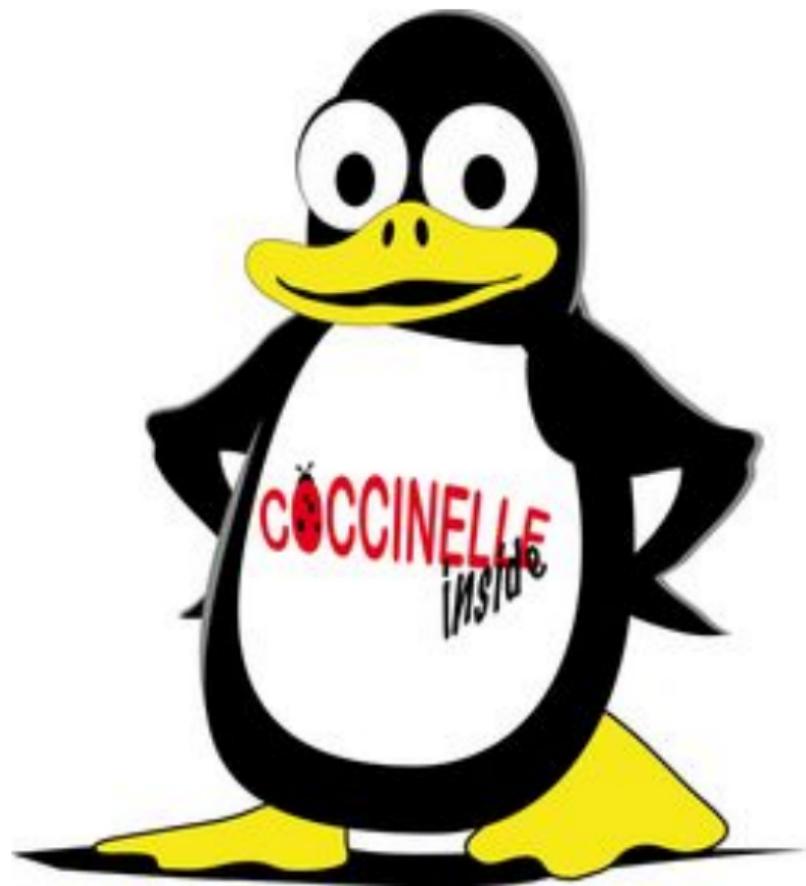
```
$ diff -u alt.txt neu.txt
...
@@ -1,2 +1,2 @@
  Das Schiffllein fliegt, der Webstuhl kracht,
- Wir weben emsig Tag und Nacht.
+ Wir weben eifrig Tag und Nacht.
```

Ausführbar als *Patches* (*Flicken*)!

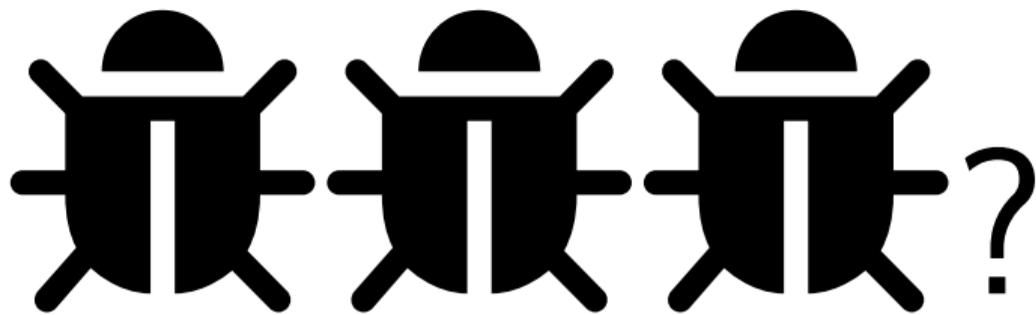
```
$ diff -u alt.txt neu.txt | patch alt.txt
patching alt.txt
```

Wozu dienen Patches?

- ▶ Bugfixing
- ▶ vertraute, datensparsame Verbreitung von Aktualisierungen



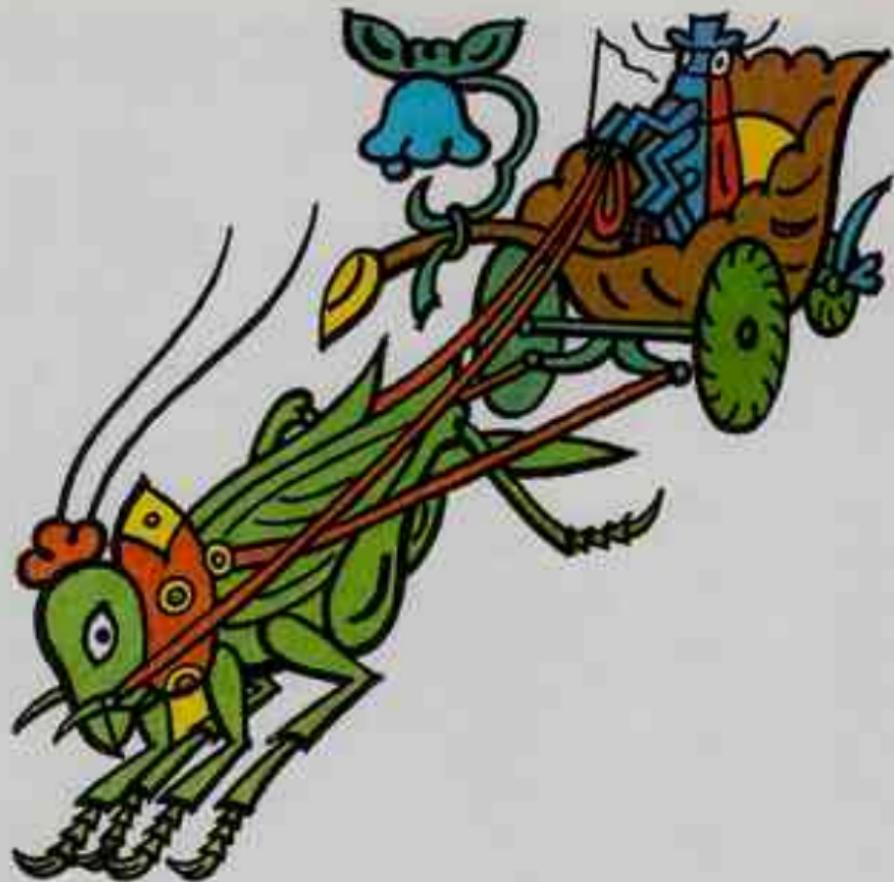
Warte mal, kannst Du mehr erzählen?



























Tolle Bilder oder?

- ▶ Das waren Zeichnungen von Josef Lada (1887–1957), aus dem Kinderbuch *Was eine Krähe der anderen erzählte*, 1978, Artia, Prag



Josef Lada.cz

- ▶ Zeichnungen benutzt mit Erlaubnis von der Nachkommen von Josef Lada

All together is better!!!



COCCINELLE
inside

Zeit für echtes Coccinelle: semantisches Patching 1

```
1 @@
2 type T;
3 identifier M = {X,Y};
4 fresh identifier G="g_"##M;
5 @@
6     struct ptcl_t {
7 -         T M;
8     };
9 ++ T G[N];
```

```
1 @@ -1,10 +1,12 @@
2     #define N 3
3     struct ptcl_t {
4 -     double X,Y,Z;
5 +     double Z;
6     };
7 +double g_X[N];
8 +double g_Y[N];
9
10    int main() {
11        struct ptcl_t aos[N];
12        // ...
13    }
```

Zeit für echtes Coccinelle: semantisches Patching 2

```
1 @r@
2 type T;
3 identifier M = {X,Y};
4 fresh identifier G="g_"##M;
5 @@
6     struct ptcl_t {
7 -     T M;
8     };
9 ++ T G[N];
10
11 @@
12 identifier P, r.M, r.G;
13 typedef ptcl_t;
14 expression E;
15 @@
16     struct ptcl_t P[N];
17     ...
18 - P[E].M
19 + G[E]
```

```
1 @@ -1,11 +1,13 @@
2     #define N 3
3     struct ptcl_t {
4 -     double X,Y,Z;
5 +     double Z;
6     };
7 +double g_X[N];
8 +double g_Y[N];
9
10 int main() {
11     struct ptcl_t aos[N];
12 - aos[0].X = aos[0].Y
13 + g_X[0] = g_Y[0]
14         + aos[0].Z;
15 }
```

Mehr darüber

<https://michelemartone.org> ← Meine Webseite mit all diesen Links ↓

▶ COCCINELLE Webpräsenz

<https://coccinelle.gitlabpages.inria.fr/website/>

▶ 6-seitiger Kurzaufsatz (Martone & Lawall 2021)

<https://hal.inria.fr/hal-03266521>

▶ Folien der 1-tägigen Schulung vom 22.01.2025

<https://doi.org/10.5281/zenodo.14728519>

