

thomas@familiezimmermann.de

Von: thomas@familiezimmermann.de
Gesendet: Thursday, 14 January 2021 20:19
An: 'geocache.wizard@gmail.com'
Betreff: WG: Malbolge-Generator/Interpreter
Anlagen: stringout.c

Moin Mark,
nachfolgend mein Mailverkehr mit Matthias Lutter und dem C-Code.
Ich habe hier ein Problem mit der Umsetzung in Dart 😞

Und brauche mal einen Profi-Blick.

Der Code läuft unter C problemlos durch und ist generiert kompaktes Malbolge. Aber in Dart kriege ich das Verhalten bislang seltsam

- Laufzeit ätzend
- Kleinbuchstaben werden nicht generiert

Der PR folgt.

Gruß
Thomas

-----Ursprüngliche Nachricht-----

Von: Matthias Lutter <matthias@lutter.cc>
Gesendet: Tuesday, 12 January 2021 23:13
An: thomas@familiezimmermann.de
Betreff: Re: Malbolge-Generator/Interpreter

Hallo Thomas,

in Dart habe ich noch nie entwickelt, aber das sollte für die Diskussion ja keine große Rolle spielen. Auf jeden Fall ein schönes Projekt!

thomas@familiezimmermann.de schrieb:

> Wenn ich Deine Ausführung richtig verstanden habe, ist in Pseudo-Code der Ablauf wie folgt:
> code ist das zu erzeugende Malbolge-Programm
>
> code = 'b' => nach der Ausführung des Befehls erhalte ich d=0, c= 98, a=0; dann wird die Speicherstelle [98] verschlüsselt und anschließend werden c, und d inkrementiert; d.h c=99, d=1.
>

Ja.

> Damit kann ich in [1] das erste Zeichen so ablegen, das es mit einem Rot nach a kopiert wird, anschließend kommt die Ausgabe.

Nicht unbedingt, da du nur eines von 8 Zeichen dort ablegen darfst, es muss einem gültigen Malbolge-Befehl entsprechen.

In einer Vereinfachung habe ich mich darauf beschränkt, einfach überall zwischen [1] und [98] ein Nop abzulegen, da man in den Zellen ab [99] soieso Zeichen haben wird, die man aufgrund des generierten Codes vorgegeben hat und spätestens ab da keine Auswahl mehr treffen kann.

Sagen wir, man will den String $x[0] \dots x[n]$ erzeugen.

Um weiß jetzt nicht, wie man $x[0]$ am besten erzeugt. Z.B. wie folgt:

Rot y1

Opr y2

Anschließend wurde nicht zwingend $x[0]$ generiert, aber ein Wert, der bei der Ausgabe $x[0]$ erzeugt. Es gilt $A \bmod 256 == x[0]$.

Das heißt, man schreibt in die Zellen [99], [100], ... so lange ein Nop, bis d auf eine Zelle zeigt, die y1 enthält. Dann schreibt man ein Rot, anschließend wieder Nops bis d auf eine Zelle mit dem Inhalt y2 zeigt.

Nun kann es natürlich sein, dass man $x[1]$ wie folgt generieren muss:

Rot y3

Opr y4

Opr y5

Mehr als 2 Opr-Befehle wird man nicht benötigen. Auch hier ist wieder die Frage, wann stehen die richtigen Werte in [d], also der Adresse, auf die d zeigt.

Manchmal hat man Glück und es reicht bereits:

Opr y6

Weil im A-Register ein günstiger Wert steht. Oder man hat noch mehr Glück und das vorherige Zeichen entspricht dem aktuellen, $x[i] == x[i-1]$. Im letzteren Fall kann man natürlich direkt die Ausgabe noch einmal schreiben.

> Anschließend die Ausgabe.

> Dieser Schritt wird wiederholt, bis alle Zeichen ausgegeben sind.

>

> Da ich im Generator den Speicher mitschreiben würde, hätte ich jederzeit den Überblick, welche Werte in d und a vorliegen.

>

> Als letzten Befehl käme dann das v.

Das ist wieder richtig.

Du musst darauf achten, dass sich der Wert von [d], also der Zelle, auf die d zeigt, ab 98 verändert. Ich glaube das war klar, aber hier können schnell Fehler passieren.

Ich habe da mal vor Jahren, damals hatte ich noch einen anderen Nachnamen, einen Generator in C geschrieben. Der Code ist echt häßlich, aber vlt. hilft es dir ja trotzdem, darum habe ich den mal angehängt.

Als der Generator von zb3 kam, habe ich meinen Generator von der Webseite genommen und auf den von zb3 verlinkt.

Viele Grüße
Matthias