

Kompressionsprogramm

Ein einfaches Huffman-Coding benutzendes Kompressionsprogramm. Das Programm kann sowohl mit 8-Bits lange als auch mit 16-Bits lange Aufteilung funktionieren. 16-Bit Kompression braucht mehr Zeit um durchgeführt zu werden, aber ergibt ein kompakter File am Ende.

Anforderungen

- Linux oder BSD basiertes System
- GMP
- AMD64 Architektur (empfohlen)

Installation

```
git clone --recursive https://github.com/mondokm/kompressionsprogramm.git
cd kompressionsprogramm
make install
```

Verwendung

```
komp -k[ompression] -d[ekompression] -8[-bit] -16[-bit] file
```

Default: 8-Bit Kompression

Kompression

Das Programm liest erstens das File Byte zu Byte durch und zählt, wie viel es von den verschiedenen Variationen gibt. Mit der gesammelten Information wird es dann ein binären Baum bauen, mithilfe von denen es die Länge der Codes berechnen wird. Mit den Längen können wir die kanonische Huffman-Kodes unseren Wörterbuches generieren, und die Kompression durchführen.

Der Ausgang wird der Filename <originalfilename>.komp haben.

Dekompression

Der Anfang der komprimiertes file enthält Header-Information (z.B. die benutzte Aufteilung, die Kodelängen, die Leftover-Byte), die wir einlesen, und benutzen, um ein Suchbaum zu bauen. Mit dem Suchbaum führen wir danach die Dekompression durch.

Beispiele

Die folgende Befehl komprimiert abc.txt in 8-Bit mode:

```
komp abc.txt
```

Um die generierte abc.txt.komp zu dekomprimieren, sollen wir die folgende Befehl benutzen:

```
komp -d abc.txt.komp
```

Die folgende Befehl komprimiert abc.txt in 16-Bit mode (die Flagge k ist optionell):

```
komp -k16 abc.txt
```

Optimalisation

8-Bit Kompression von Files kleiner als 295 Bytes wird sicherlich in einem File größer als die originelle resultieren. 16-Bit Kompression ist nur für files größer als 70 KBytes empfohlen.