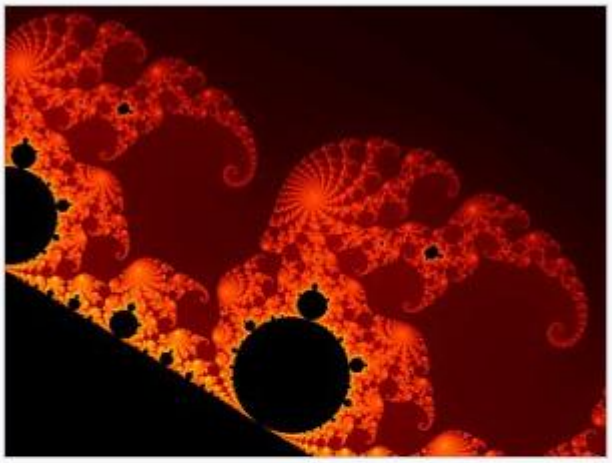
プログラムはどこまで小さくできるか？

2025 年 10 月 25 日(土)　オープンソースカンファレンス 2025 東京秋　OSASK 計画

[1] 究極まで小さくしたい！

プログラムはどこまで小さくできるのか？実行に際して不要な情報をすべて取り除いたら何が残るのか？私はその探求がやめられない。ロマンにあふれている。



この画像はフラクタル図形であるマンデルブロ集合の一部である。このJPEGファイルのサイズは17KB以上（約140,000ビット）ある。ところが、これと同じファイルは140,000ビットよりも遥かに小さいコンピュータ・プログラムによって作成することが出来る。従って、このJPEGファイルのコルモゴロフ複雑性は140,000よりも遥かに小さい。

[2] 祝！世界記録更新

Wikipedia で「コルモゴルフ複雑性」を調べてみると、右のような説明のついた画像が出てきます。

17KB よりも遥かに小さいプログラムで作成できると書いてありますが、実際にどのくらいになるのか全く示されていません。それで 10 年以上前に **109 バイト**という記録を私が打ち立てました（自作の機械語で）。

今回この問題に再チャレンジして、**75 バイト**にできました。つまり世界記録を塗り替えました！（ちょっと試せばわかりますが、109 バイトはすでにとんでもない記録です。そして 75 バイトはもっととんでもないです）。

[3] 基本的な仕組み

変数名は i, j, k, …などにしてしまっても実行には差し支えないです。それなら var0, var1, var2, …みたいに番号にしてしまいましょう（出現頻度が高いものを var0 や var1 にする）。演算子も適当に番号を決めてすべて数値にしましょう。for や if もすべて数値で表します。あとはそれを並べるだけです。

```
s = 0;
for (i = 1; i <= 100; i++) {
  s += i;
}
→ 4-c65-0; 1-a0-1; 5;
→ 4c 65 01 a0 15 (5 バイト)
```

1～100 までの和を求める for ループは右の例のように 5 バイトになります。かなりコンパクトですね！x86 では何バイトですか？

4 は for を意味しています。1 は+=を意味しています。5 は}を意味しています。ここで最初の s=0; に対応するバイトコードがないことに気づきます。このバイトコード体系では変数はすべて 0 で初期化されているので、省略しているのです（展示ブースにはさらに詳しい資料もあります）。

4c6501a0…というバイト列があった時に、これを 4, c65, 0, 1, a0, …と正しく切り分ける技術が必要になります。実は 0～6 で始まる 16 進数は 1 桁、8～b で始まる 16 進数は 2 桁、c～d で始まる 16 進数は 3 桁、e で始まる 16 進数は 4 桁、7 で始まる 16 進数は 5 桁以上と決めているので、切り分けは簡単にできるのです。・・・このサンプルコードや上記の 75 バイトを C 言語に戻すためのプログラムは 390 行くらいで書けました。それはつまり変換規則がそれだけ単純だということです。

[4] 旅はまだ終わらない～♪

109 バイトという記録を出した 10 数年前、私は限界に到達したかも？という感触がありました。しかし今回さらに改良できてしまったので、実はまだまだだったと分かりました。今は 75 バイトが限界だろうと思っていますが、10 年後にはさらにうまいコード体系を発明しているかもしれません・・・。