

ソフトウェアよもやま奇譚

飯田 浩志*

はじめに

2020年度から小学校では、読み書きソロバンならぬ、読み書きプログラミングになるとか [2]. 翻^{ひら}つて筆者はと云へば、大学卒業と同時に上京し、ソフトウェアの受注開発をメインとする企業^{おい}に於て9年の実務経験がある。此^この小品では、ソフトウェアに^{かん}して、筆者が先輩諸氏から見聞きした事柄や自身の経験に基づき、ソフトウェアを専門としない一般の方でも興味を持てさうな話題を、幾つか紹介したい。

緑のランプ

或^ある先輩から聞いた話。たしか、なんらかのプラントの制御パネル（壁に、スイッチ類、各種メータとかランプ等が集まったもの）を動作させるソフトの開発だつたと記憶してある。当該パネルの中に緑色のランプがあるのだが、困^そつたことに、其れがどうやつても光らない。正しいオペレーションをしてゐるにも^{かかは}拘らず光らない。探せども探せどもソフトにバグ（誤り）は見つからず、そんな^つ点かぬ^{はず}筈がない、なぜだ何故なんだ、と途方に暮れた挙句、よくよく調べてみたら、其のランプの電球が切れてゐたといふオチ。LEDなら好かつたのにね*¹。

*E-mail address: auau2.a.go.go@gmail.com

主記憶の怪

さるメーカーが新型プリンタの開発を進めてみた頃、其のプリンタの組込みOSを並行して開発してみた時の話。開発言語はCとMotorola MC68000のアセンブラだった。MC68Kの動作クロック周波数は12.5 MHzで、パイプラインなんてものも無く、ロジックアナライザなる機器で一命令ずつトレースしてdebugとか、平和な時代だった。ちなみに、実機（開発途中のプリンタ）上でのソフト開発は不可能につき、往年のDEC VAX11-785上でソースを作成し、クロスコンパイラが吐いたコードを実機にロードして実行してみた。

閑話休題、開発の途中、時折エラーを起こすプログラムに悩まされてみた。これが、再現性がまったく無く、突然、誤動作する。或る決まった条件下で必ずエラーになるのであれば比較的、原因の究明は容易なのだが、再現性がないとなると非常に厄介である*²。そんなある日、ハードウェア担当者から、主記憶がたまたまparity errorを起こすことが判明したと知らされた。なんだそうか。主記憶上で勝手にプログラムの0-1が引つ繰り返つて化けてゐたのか。そんなの分かるわけがない。信じるものは何もない、と此のとき悟つた。

数値計算を疑え

ときに、コンピュータの計算は精確だらうか？ ちつは、誤差がある*³。既に詳細は忘れてしまつたが、CADの仕事をしてゐた頃、FORTRANで、倍精度実数型（double precision）の変数a, bに対して $a-b+b$ （あるいは $a+b-b$ だったかも） $=a$ と信じていたが、たしか、さうはならぬ場面に遭遇した記憶がある。

*1 本題とはまるで無関係ながら是迄に、^{これまで}仕事帰りの電車の中でバグに気づくと云ふ人に数多く出会つた気がするけれども、これは『プログラマあるある』なんだらうか？

*2 だからglobal変数は嫌いなんだ。

*3 おまけに、扱へる数値の大きさ（及び小ささ）にも限界アリ。

実際、誤差に就て例へば、以下のPerlプログラムで*4:

```
$ cat third.pl
#!/usr/local/bin/perl

my $a = my $b = 1.0;
$b /= 3;
foreach (1..3) {
    $a -= $b
}
print +"1-(1/3)*3 = ", $a, "\n"
$ ./third.pl
1-(1/3)*3 = -5.42101086242752217e-20
$
```

と、零にならない。1/3の他1/10でも、同様の結果となる。無論、 $1/4=(0.01)_2$ なら綺麗に零になる。ちなみに⁵此処で用ゐたPerlのversionは5.26.1である*5。

コンパイラさえも疑え

今でこそ、パソコンでフルカラー（ほぼ1670万色）表示は普通*6だけれど、RGB合はせて $3+3+2=8$ ビットの時代には、 $2^8=256$ 階調の色しか出なかつた。兎に角、メモリが高価だった。石（CPU）が32ビットの時代に、主記憶が4 Gbyteあれば仮想記憶なぞ要らんのにね*7、などと冗談を云つてみたものだが、今や、手元のノートパソコンでも16 Gbyteもの主記憶がある。すごい時代になったものだ。

*4 Macのターミナル（Windowsで云へばDOS窓に相当）で実行した。

*5 ⁵Configure -dで作成にも拘らず、タイマ系のテストで一つエラーになる。5.26の儘にしておけば好かつた、失敗した。

*6 カーニハン先生が著書 [1] の多くを執筆されたパソコンも1600万色の表示が可能なもの（p. 22参照）。

*7 2^{32} は、 $2^{10}=1024\approx 10^3$ と考へれば、概ね $4 \times (10^3)^3$ 。

さて、とある企業から、新規に作成したフルカラーのボードを制御するソフトの開発を依頼された。開発は順調に進み、そろそろ完成かと思われた矢先、ひとつ問題が発生した。新規のボードを指した実機に加えてコンパイラも当該企業から提供されてゐたのだが、コンパイル時に最適化オプションを附加すると誤動作することが判明した。調べてみると、コンパイラのバグで、最適化した時の出力コードに誤りがあった。前述したやうに、MC68Kで開発したことがあり、今回のMC68030（懐かしい）のassemble codeに馴染みがあったことが幸いした。結局、最適化なしでコンパイルしたソフトを納品することに落ち着いた。

おわりに

以上が示すやうに、ソフトウェア開発の現場では、予期せぬことが起こり得る。万全を尽くしたかに思へても、意外な落とし穴に^{はま}嵌ることがままあり、thrillingであると同時に奥が深い。この^{つたな}拙い記事が契機となつてソフトウェアに興味を持ち、より多くの優秀な^{かたがた}方々が、其の開発に携わつていただけるやうになることを切に希望する。

参考文献

- [1] Brian W Kernighan, *Understanding the Digital World*. Princeton Univ Press 2017.
- [2] プログラミングが今なぜ注目されるのか?, *Mac Fan* 2017年11月号, マイナビ, pp. 236-9.