# 1　標準入力によって命令実行回数が変えられるプログラム

```
#include <stdio.h>

void main(){
  long n, i;
  printf("Input n(Million): ");
  scanf("%d", &n);
  n *= 1000000;

  for(i = 1; i <= n; i++) i+1;
}
```

# 2　$n$ と $N$ について

例えば上のプログラムでは、処理数は

| 命令 | 処理数 |
|---|---|
| printf | 1 |
| scanf | 1 |
| n *= 1000000 | 2 |
| for ～ | $4n+1$ |

である。for 文は一回ループするごとに、i <= n の比較で一回と、i++、つまり i+1 という演算と i = ... という代入で二回の処理を行う。それと for 文で囲まれた式が一処理であるので一回の処理を行う。
初期設定式は $n$ の大きさに関わらず一回の処理であるので、$4n+1$ という式が成り立つ。
従って、上のプログラムの処理数は $4n+5$、$n$ の大きさに比例する[1]。

# 3　CPU 時間の測定

測定に使ったツールは time コマンド。$n = 100$(Million) から始め、550(Million) まで十回に分けて CPU 時間を測定した。

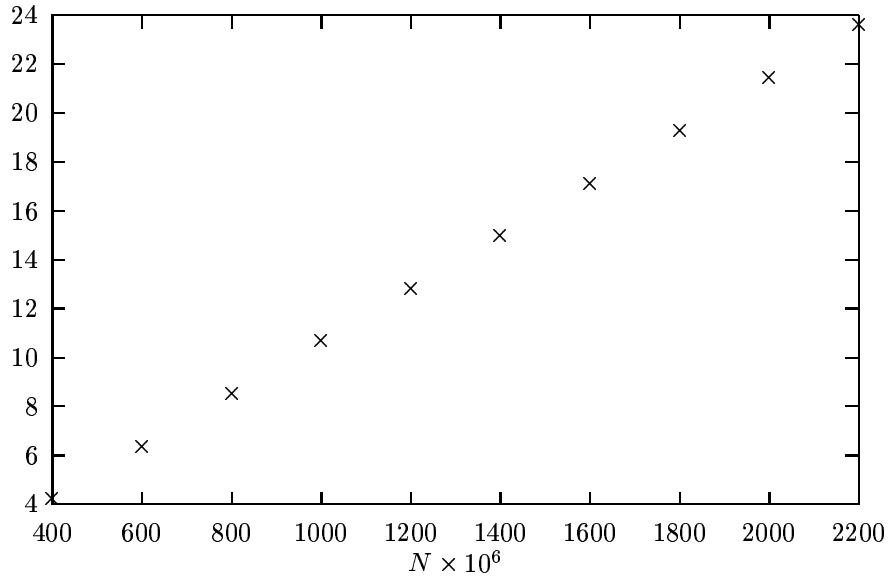
| $n$(Million) | 時間 (sec) | 実処理数 (Million) |
|---|---|---|
| 100 | 4.32 | 400 |
| 150 | 6.46 | 600 |
| 200 | 8.61 | 800 |
| 250 | 10.78 | 1000 |
| 300 | 12.93 | 1200 |

| $n$(Million) | 時間 (sec) | 実処理数 (Million) |
|---|---|---|
| 350 | 15.08 | 1400 |
| 400 | 17.23 | 1600 |
| 450 | 19.39 | 1800 |
| 500 | 21.54 | 2000 |
| 550 | 23.69 | 2200 |

[1]ただし、C コンパイラの最適化処理によってはこの限りではない。

# 4　最小二乗法による近似計算式

先ほど測定した値をグラフにしてみると、以下のようになった。




これを、最小二乗法で近似計算式を求めるために $x, y$ の関係の表を作ると以下のようになる。

| | | | | | | | | | | | 横計 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x$ | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 | 2200 | 13000 |
| $y$ | 4.32 | 6.46 | 8.61 | 10.78 | 12.93 | 15.08 | 17.23 | 19.39 | 21.54 | 23.69 | 140.03 |
| $xy$ | 1728 | 3876 | 6888 | 10780 | 15516 | 21112 | 27568 | 34902 | 43080 | 52118 | 217568 |
| $x^2$ | 160000 | 360000 | 640000 | 1000000 | 1440000 | 1960000 | 2560000 | 3240000 | 4000000 | 4840000 | 20200000 |

最小二乗法による近似計算式の係数を求めるためには

$$
\begin{aligned}
a &= \frac{Z - \frac{XY}{n}}{W - \frac{X^2}{n}} \\
b &= \frac{Y}{n} - \frac{X}{n} \times a
\end{aligned}
$$

の式を使う。ここでは、$x$ の横計が $X$、$y$ の横計が $Y$、$xy$ の横計が $Z$、$x^2$ の横計が $W$ である。
上表の合計値をこの式に当てはめると、以下のようになる。

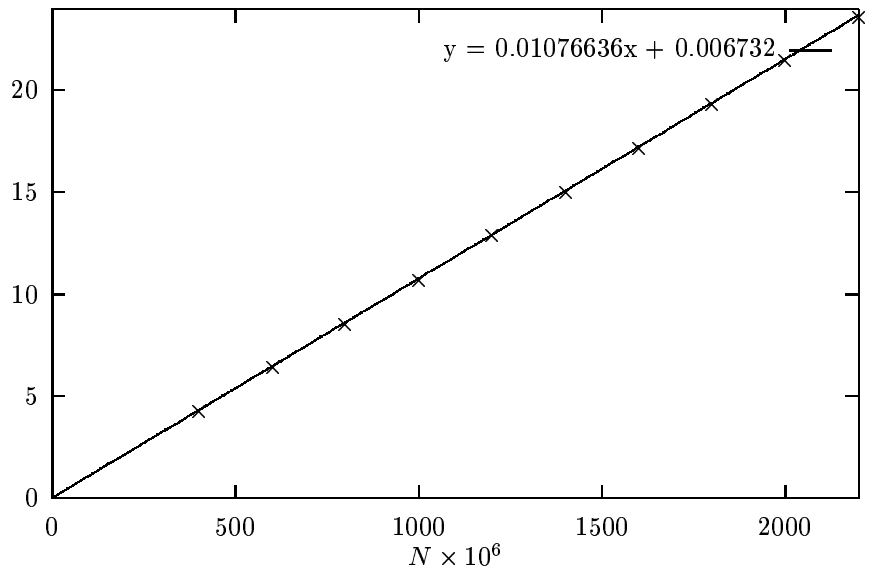
$$
\begin{aligned}
a &= \frac{217568 - \frac{13000 \times 140.03}{10}}{20200000 - \frac{13000^2}{10}} \\
&= 0.01076636 \\
b &= \frac{140.03}{10} - \frac{13000}{10} \times 0.01076636 \\
&= 0.006732
\end{aligned}
$$

従って、近似計算式は

$$y = 0.01076636x + 0.006732$$

となる。

ちなみに、この式のグラフと先ほどのグラフを重ねてみると




となり、正確に求められていることが視覚的にも良く分かる。

## 5 コンピュータのMIPS値

最小二乗法で求める場合は、先ほど求めた $x$ の係数 $a$ を使う。

$$\begin{aligned} MIPS_1 &= \frac{1}{a} \\ &= \frac{1}{0.01076636} \\ &= 92.8819025 \end{aligned}$$

一方、概算（ここでは 10 個の測定結果の平均を使った）で求めると

$$\begin{aligned} MIPS_2 &= \frac{X}{Y} \\ &= \frac{13000}{140.03} \\ &= 92.8372492 \end{aligned}$$

多少の誤差こそあるが、大体は同じである。このコンピュータは一秒間に 9300 万回程度の整数演算を行うことが出来る。

## 6 一回の命令実行に要する平均クロックサイクル数が 2 であると仮定したときのクロック周波数

平均クロックサイクル数 CPI、命令実行回数 $I \times N$、CPU 時間 $T_{cpu}$ の 3 つが分かれば、

$$T_{cpu} = \sum_i (I_i \times CPI_i) \times Ck \tag{1}$$

$$Ck = \frac{1}{CR} \tag{2}$$

の二つの式から逆算出来る。まず (1) の式から Ck を求める為に、式を以下のように移項する。

$$Ck = \frac{T_{cpu}}{\sum_i (I_i \times CPI_i)} \tag{3}$$

CPU 時間は 140.03、命令実行回数は $13000 \times 10^6$、平均クロックサイクル数は 2 であるので、(3) より

$$\begin{aligned} Ck &= \frac{140.03}{13000 \times 10^6 \times 2} \\ &= \frac{0.00538577}{10^6} \\ &= 0.00538577 \times 10^{-6} \end{aligned}$$

また、(2) より、

$$\begin{aligned} 0.00538577 \times 10^{-6} &= \frac{1}{CR} \\ CR &= \frac{1}{0.00538577 \times 10^{-6}} \\ &= 185.674472 \times 10^6 \end{aligned}$$

一回の命令実行に要する平均クロックサイクル数が 2 であると仮定したとき、クロック周波数は 186MHz 程度と推測される。