

# 目次

フィボナッチ数列でベンチマーク ..... 1

**テスト** ..... 1

Python ..... 1

Perl ..... 1

LuaJIT ..... 2

node.js ..... 2

Ruby2.0 ..... 3

Ruby1.9 ..... 3

PHP5 ..... 4

Java SE Embedded 8 ..... 4

Java (OpenJDK 11) ..... 5

GCC ..... 5

Nim ..... 6



# フィボナッチ数列でベンチマーク

MA-E3xxには、スクリプト言語として

- Python3.4
- Python2.7 (v2.2.0 以降追加)
- Perl5.8
- LuaJIT-2.0.2
- nodejs-0.10 (v2.2.0 以降削除予定)

を導入してあります。

せっかくなので、[satosystemsの日記 - フィボナッチで各種言語をベンチマーク](#)で行っているベンチマークテストを、MA-E3xx上で各種言語でテストしてみました。

## テスト

### Python

コードは下記になります。

```
def fib(n):  
    if n < 2: return n  
    return fib(n - 2) + fib(n - 1)  
  
print(fib(38))
```

実行結果はこのようになりました。

```
user1@plum:~$ time python3 fib.py  
39088169  
  
real    2m35.226s  
user    2m33.116s  
sys    0m0.072s
```

### Perl

```
sub fib($) {
```

```
    return $_[0] if ($_[0] < 2);
    return fib($_[0] - 2) + fib($_[0] - 1);
}

print fib(38), "\n";
```

実行結果はこのようになりました。

```
user1@plum:~$ time perl fib.pl
39088169

real    4m5.286s
user    4m1.284s
sys     0m0.572s
```

## LuajIT

fib.lua

```
function fib(n)
    if n < 2 then return n end
    return fib(n - 2) + fib(n - 1)
end

print(fib(38))
```

実行結果はこのようになりました。

```
user1@plum:~$ time luajit fib.lua
39088169

real    0m9.108s
user    0m8.592s
sys     0m0.396s
```

JITコンパイラによりネイティブコードに変換されるだけあって、桁違いに速いです。

## node.js

```
function fib(n) {
```

```
    if (n < 2) return n;
    return fib(n - 2) + fib(n - 1);
}

console.log(fib(38));
```

実行結果はこのようになりました。

```
user1@plum:~$ time nodejs fib.js
39088169

real    0m7.108s
user    0m6.884s
sys 0m0.084s
```

さすがは V8 ☐LuaJIT よりも高速でした。

## Ruby2.0

Ruby2.0 を追加で導入し、テストしてみました。

```
def fib(n)
  return n if (n < 2)
  return fib(n - 2) + fib(n - 1)
end

puts fib(38)
```

実行結果はこのようになりました。

```
user1@plum:~$ time ruby2.0 fib.rb
39088169

real    0m58.870s
user    0m56.992s
sys 0m0.952s
```

## Ruby1.9

Ruby2.0 と同じコードを、Ruby1.9 で実行してみます。

```
user1@plum:~$ time ruby1.9.1 fib.rb
```

39088169

```
real    1m20.224s
user    1m18.072s
sys     0m0.944s
```

## PHP5

```
<?php
function fib($n) {
    if ($n < 2) return $n;
    return fib($n - 2) + fib($n - 1);
}

print fib(38);
print "\n";
?>
```

```
user1@plum:~$ time php5-cgi fib.php
X-Powered-By: PHP/5.5.9-1ubuntu4.3
Content-type: text/html
```

39088169

```
real    2m39.913s
user    2m38.097s
sys     0m0.067s
```

Python3 と同程度、こんなところでしょうか。

## Java SE Embedded 8

コンパイルする言語なので、速いと思いますが一応。

```
public class fib {
    static int fib(int n) {
        if (n < 2) return n;
        return fib(n - 2) + fib(n - 1);
    }
}
```

```
public static void main(String[] args) {  
    System.out.println(fib(38));  
}  
}
```

実行結果は下記のようにになりました。

```
user1@plum:/tmp$ time java fib  
39088169  
  
real    0m5.021s  
user    0m2.024s  
sys 0m2.796s
```

さすがは HotSpot による JIT コンパイルが効くだけあって、スクリプト言語よりも高速です。VM の起動に時間がかかるせいかな “user” は短いですが “sys” で時間がかかっています。

## Java (OpenJDK 11)

OpenJDK 11 でも試してみます。

```
user1@plum:~$ time java fib  
39088169  
  
real    0m7.083s  
user    0m2.139s  
sys 0m4.703s  
user1@plum:~$ time java fib  
39088169  
  
real    0m2.284s  
user    0m2.157s  
sys 0m0.114s
```

```
user1@plum:~$ java -version  
openjdk version "11.0.3" 2019-04-16  
OpenJDK Runtime Environment (build 11.0.3+7-Ubuntu-1ubuntu218.04.1)  
OpenJDK Server VM (build 11.0.3+7-Ubuntu-1ubuntu218.04.1, mixed mode)
```

初回はキャッシュされてないので遅いですが “user” の時間は Java SE embedded 8 とあまり遜色なさそうな感じです。

## GCC

ネイティブコードにコンパイルするC言語でも試してみます。

```
#include <stdio.h>

int fib(int n) {
    if (n < 2) return n;
    return fib(n - 2) + fib(n - 1);
}

int main(int argc, char *argv[]) {
    printf("%d\n", fib(38));
    return 0;
}
```

実行結果は下記のようにになりました。

```
user1@plum:/tmp$ time ./fib
39088169

real    0m1.376s
user    0m1.332s
sys     0m0.012s
```

スクリプト言語やVM系とは比較にならないほど高速でした。

## Nim

いま注目の Nim では？

```
proc fib(n: int): int =
  if n < 2:
    return n
  else:
    return fib(n - 1) + fib(n - 2)

echo(fib(38))
```

```
user1@plum:~$ time ./fib
39088169

real    0m1.324s
user    0m1.313s
```



```
sys 0m0.004s
```

C よりも速いとは。



From:

<https://wiki.centurysys.net/> - MA-X/MA-S/MA-E/IP-K Developers' WiKi

Permanent link:

[https://wiki.centurysys.net/doku.php?id=mae3xx\\_tips:test\\_fib\\_benchmark:start](https://wiki.centurysys.net/doku.php?id=mae3xx_tips:test_fib_benchmark:start)

Last update: **2023/06/23 07:46**