

# OS 第二章编程作业

191220008 陈南瞳

## 1、运行机器的CPU核数

2核（虚拟机中为2核，实际机器为4核）

## 2、不同n值的情况下，程序的运行时间开销及计算结果，并分析

# 直接使用标准库自带的伪随机数生成函数会导致系统调用开销急剧增加!!!

因有该提示，所以决定对比一下在分别使用 `rand()` 函数和 `rand_r()` 函数的情况下，时间开销有何不同。

此外，我将总采样数修改为了一千万： `NUM = 10000000`

① 使用 `rand()` 函数

程序运行结果如下：

```
cnt@ubuntu:~/os2021$ time ./calc_pi -t 1
PI = 3.141850

real    0m0.700s
user    0m0.693s
sys     0m0.004s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 2
PI = 3.141748

real    0m2.632s
user    0m3.365s
sys     0m1.821s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 10
PI = 3.142137

real    0m2.049s
user    0m2.427s
sys     0m1.354s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 100
PI = 3.141932

real    0m2.851s
user    0m3.536s
sys     0m2.052s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 1000
PI = 3.141919

real    0m2.916s
user    0m3.446s
sys     0m2.331s
```

```
cnt@ubuntu:~/os2021$ time ./calc_pi -t 10000
PI = 3.141632

real    0m2.718s
user    0m2.814s
sys     0m2.461s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 100000
PI = 1.029128

real    0m0.887s
user    0m0.234s
sys     0m0.988s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 1000000
PI = 0.102892

real    0m2.956s
user    0m0.795s
sys     0m2.445s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 10000000
PI = 0.010306

real    0m21.829s
user    0m6.894s
sys     0m15.150s
```

分析:

在n=1和n=2时, real time, user time和sys time有一个激增。原因可能是因为rand()函数中使用了静态变量,多线程时可能会引起数据竞争,导致系统开销急剧增大,所以会导致激增。

当n=100000时发生突变,PI结果也发生突变。

## ② 使用 rand\_r() 函数

程序运行结果如下:

```
cnt@ubuntu:~/os2021$ time ./calc_pi -t 1
PI = 3.142097

real    0m0.303s
user    0m0.297s
sys     0m0.004s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 2
PI = 3.140154

real    0m0.183s
user    0m0.335s
sys     0m0.008s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 10
PI = 3.142840

real    0m0.179s
user    0m0.296s
sys     0m0.043s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 100
PI = 3.143720

real    0m0.185s
user    0m0.285s
sys     0m0.066s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 1000
PI = 3.149578

real    0m0.176s
user    0m0.256s
sys     0m0.065s
```

```
cnt@ubuntu:~/os2021$ time ./calc_pi -t 10000
PI = 3.156000

real    0m0.354s
user    0m0.280s
sys     0m0.302s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 100000
PI = 1.009067

real    0m0.886s
user    0m0.120s
sys     0m1.085s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 1000000
PI = 0.099808

real    0m2.779s
user    0m0.807s
sys     0m2.253s
cnt@ubuntu:~/os2021$ time ./calc_pi -t 10000000
PI = 0.013102

real    0m23.245s
user    0m7.280s
sys     0m16.205s
```

分析：

使用rand\_r()函数时，整体时间小于rand()函数的情况，这是由于rand\_r()是线程安全的函数，不会出现数据竞争的问题。

预期结果：随着n的增大，real time不断减小，user time 基本不变，sys time不断增大。这是因为real time计算的是进程从开始到结束的总时间，由于虚拟机使用了2个处理器，采用多线程可以并行地处理任务，加快运算速度。user time是每隔线程占用CPU的总的时间，与线程数无关，应当保持不变。当线程数增大时，系统调用的时间也会相应增大。

经过多次的测试，发现规律：

- real time在n=1和n=2时有一个比较明显的减小，当n>2时real time变化不大，但当n接近总采样数的量级时，real time会急剧增大。（因为是2个处理器，所以n: 1→2会有突变）
- user time一直变化不大，但当n接近总采样数的量级时，会明显增大。
- sys time总体呈递增趋势。
- 当n过大时，PI的结果会突变。

整体上实验结果与预期结果相符。