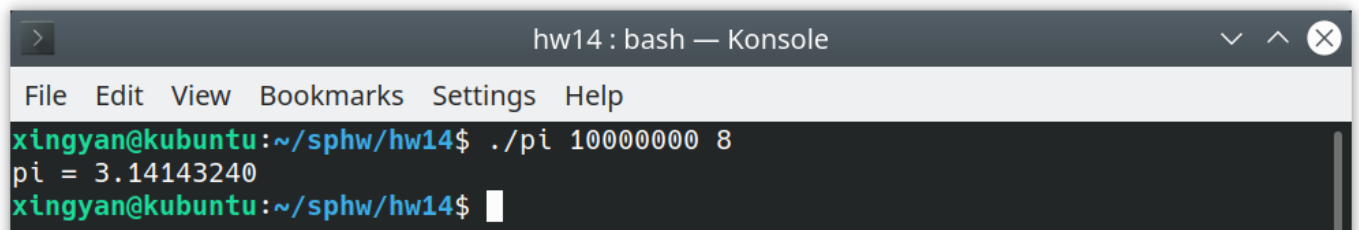


System programming homework #14

王興彥 408410113

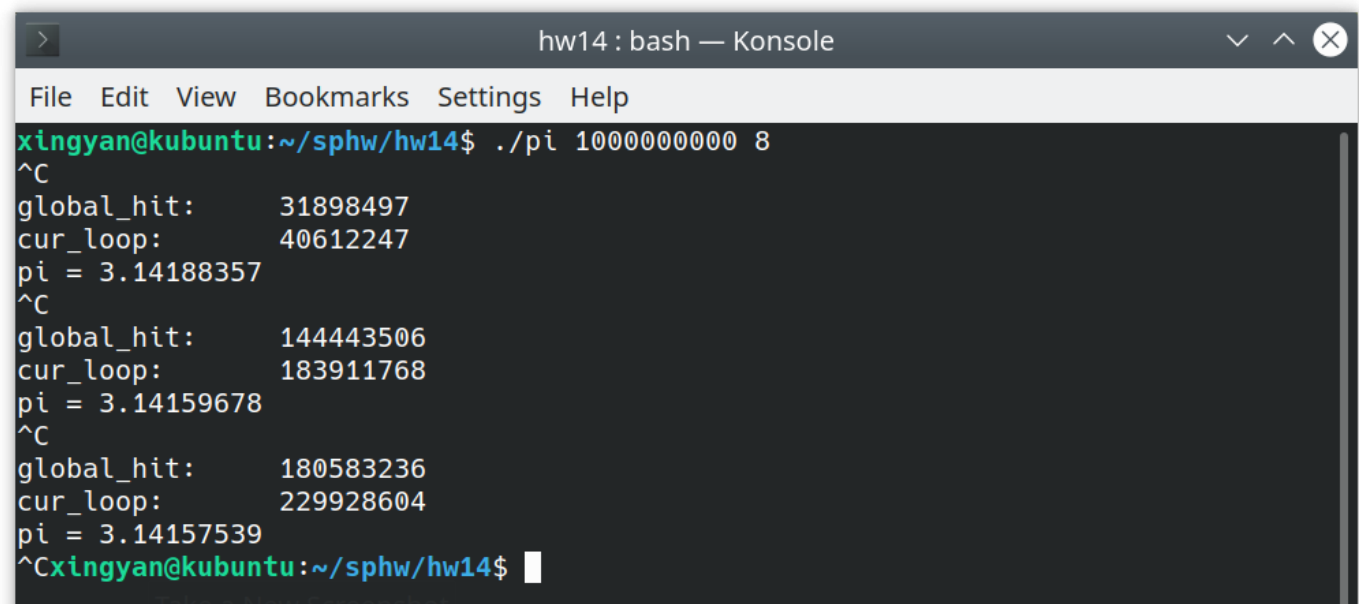
0. 完成 1234

1. 請撰寫應用程式，使用蒙特卡羅法計算 π ，應用程式可以有二個參數，第一個參數是總共打多少個點，第二個參數是使用多少 thread 做運算。



```
hw14: bash — Konsole
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/sphw/hw14$ ./pi 10000000 8
pi = 3.14143240
xingyan@kubuntu:~/sphw/hw14$
```

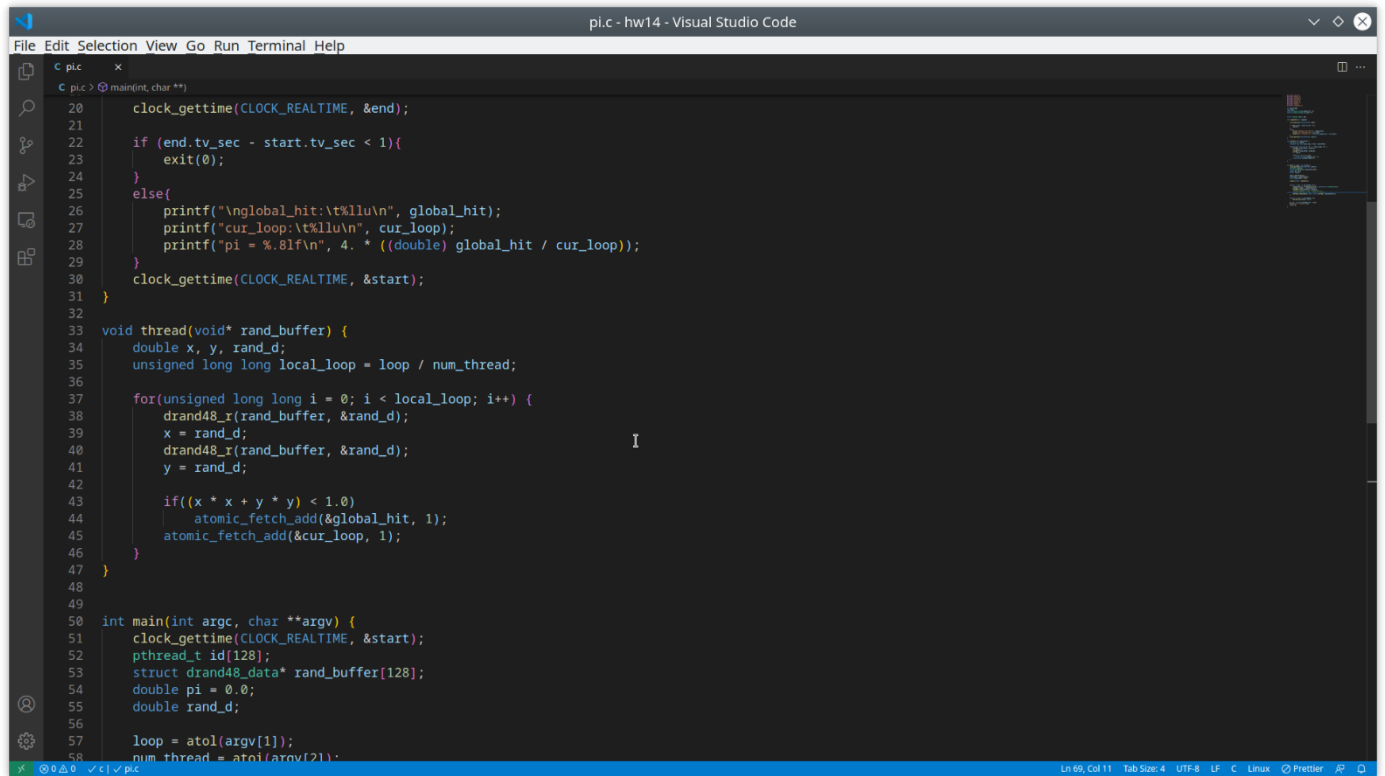
2. 當按下 `ctr-c` 的時候，顯示截至目前計算出來的 π 是多少。
3. 當在一秒內連續按下二次 `ctr-c` 的時候，顯示截至目前計算出來的 π 是多少，並結束程式。



```
hw14: bash — Konsole
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/sphw/hw14$ ./pi 1000000000 8
^C
global_hit:      31898497
cur_loop:        40612247
pi = 3.14188357
^C
global_hit:      144443506
cur_loop:        183911768
pi = 3.14159678
^C
global_hit:      180583236
cur_loop:        229928604
pi = 3.14157539
^C xingyan@kubuntu:~/sphw/hw14$
```

4. 請說明你的應用程式比授課老師所給的範例程式快或者是慢，並『具體』說明變快或者變慢的原因。

程式執行的比範例程式慢，因為所需變數更動時，都會直接用 atomic 直接更新，因此比範例程式慢，因為範例程式是 thread 算完之後再一次更新，消耗的時間較少。



```
pi.c - hw14 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C pi.c x
C pi.c > main(int, char **)
20 clock_gettime(CLOCK_REALTIME, &end);
21
22 if (end.tv_sec - start.tv_sec < 1){
23     exit(0);
24 }
25 else{
26     printf("\nglobal_hit:\t%llu\n", global_hit);
27     printf("cur_loop:\t%llu\n", cur_loop);
28     printf("pi = %.8lf\n", 4. * ((double) global_hit / cur_loop));
29 }
30 clock_gettime(CLOCK_REALTIME, &start);
31 }
32
33 void thread(void* rand_buffer) {
34     double x, y, rand_d;
35     unsigned long long local_loop = loop / num_thread;
36
37     for(unsigned long long i = 0; i < local_loop; i++) {
38         drand48_r(rand_buffer, &rand_d);
39         x = rand_d;
40         drand48_r(rand_buffer, &rand_d);
41         y = rand_d;
42
43         if((x * x + y * y) < 1.0)
44             atomic_fetch_add(&global_hit, 1);
45         atomic_fetch_add(&cur_loop, 1);
46     }
47 }
48
49
50 int main(int argc, char **argv) {
51     clock_gettime(CLOCK_REALTIME, &start);
52     pthread_t id[128];
53     struct drand48_data* rand_buffer[128];
54     double pi = 0.0;
55     double rand_d;
56
57     loop = atol(argv[1]);
58     num_thread = atoi(argv[2]);
```