

[illegible]

### 3. 執行『./peterson\_correct-g』

```
chunchi@chunchi-TravelMate-P243: ~/文件/os作業6
peterson_correct-g peterson_correct-03
chunchi@chunchi-TravelMate-P243:~/文件/os作業6$ ./peterson_correct-g
start p0
start p1
進入次數 (每秒) p0: 4861097, p1: 4860124, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 4703689, p1: 4700927, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 4725820, p1: 4725639, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 4726695, p1: 4726001, 分別執行於 core#3 及 core#1
進入次數 (每秒) p0: 4708691, p1: 4706508, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4698721, p1: 4691247, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4730633, p1: 4731410, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4717846, p1: 4719944, 分別執行於 core#3 及 core#1
進入次數 (每秒) p0: 4720426, p1: 4717425, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4724373, p1: 4723124, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4709845, p1: 4708275, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4728819, p1: 4730008, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4717616, p1: 4716839, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4725399, p1: 4724204, 分別執行於 core#3 及 core#1
進入次數 (每秒) p0: 4722749, p1: 4721346, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4722444, p1: 4721596, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4726968, p1: 4727627, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4729258, p1: 4733932, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4730631, p1: 4732068, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4725600, p1: 4726418, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4686269, p1: 4682772, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4705232, p1: 4704856, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4720978, p1: 4721885, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4719066, p1: 4719086, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4715789, p1: 4715158, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 4725342, p1: 4725418, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4729582, p1: 4730214, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4727419, p1: 4727615, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4719858, p1: 4718985, 分別執行於 core#2 及 core#1
進入次數 (每秒) p0: 4721347, p1: 4722171, 分別執行於 core#3 及 core#1
進入次數 (每秒) p0: 4688263, p1: 4682585, 分別執行於 core#3 及 core#1
^C
chunchi@chunchi-TravelMate-P243:~/文件/os作業6$ ^C
chunchi@chunchi-TravelMate-P243:~/文件/os作業6$
```

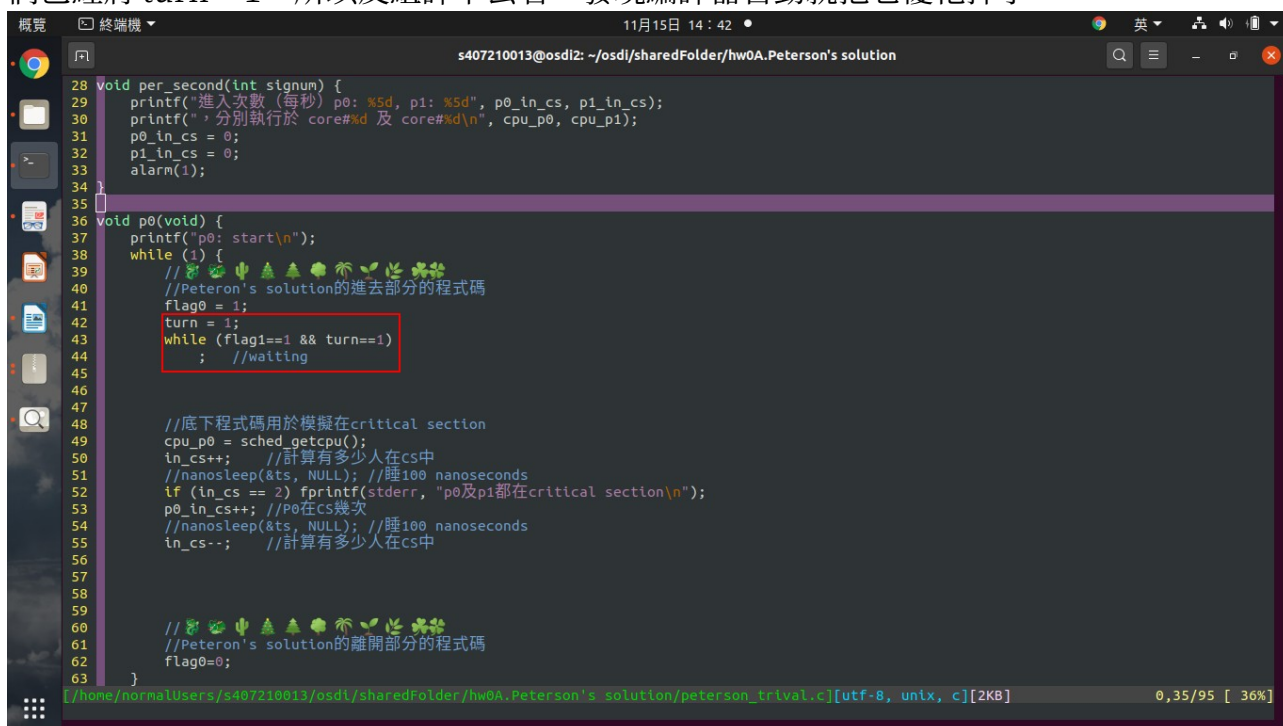
### 4. 執行『./peterson\_correct-O3』

```
chunchi@chunchi-TravelMate-P243: ~/文件/os作業6
./peterson_correct-O3
start p0
start p1
進入次數 (每秒) p0: 3556061, p1: 3553888, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3524743, p1: 3524804, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3501220, p1: 3504570, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3495863, p1: 3496160, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3498112, p1: 3498427, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3491225, p1: 3488937, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3489184, p1: 3487397, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3495223, p1: 3495116, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3491839, p1: 3492870, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3492529, p1: 3493283, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3497782, p1: 3496408, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3496678, p1: 3497023, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3498529, p1: 3496110, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3491022, p1: 3489781, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3491661, p1: 3492966, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3494048, p1: 3492800, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3499859, p1: 3498928, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3493407, p1: 3492420, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3494087, p1: 3499801, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3487200, p1: 3487127, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3492934, p1: 3490675, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3509853, p1: 3508766, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3494114, p1: 3497215, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3497477, p1: 3498563, 分別執行於 core#2 及 core#0
進入次數 (每秒) p0: 3497355, p1: 3496305, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3492786, p1: 3493005, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3495241, p1: 3494687, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3494898, p1: 3494979, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3493674, p1: 3492627, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3495372, p1: 3495285, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3499675, p1: 3499368, 分別執行於 core#3 及 core#0
進入次數 (每秒) p0: 3496967, p1: 3494711, 分別執行於 core#3 及 core#1
進入次數 (每秒) p0: 3495807, p1: 3495059, 分別執行於 core#2 及 core#1
^C
chunchi@chunchi-TravelMate-P243:~/文件/os作業6$
```

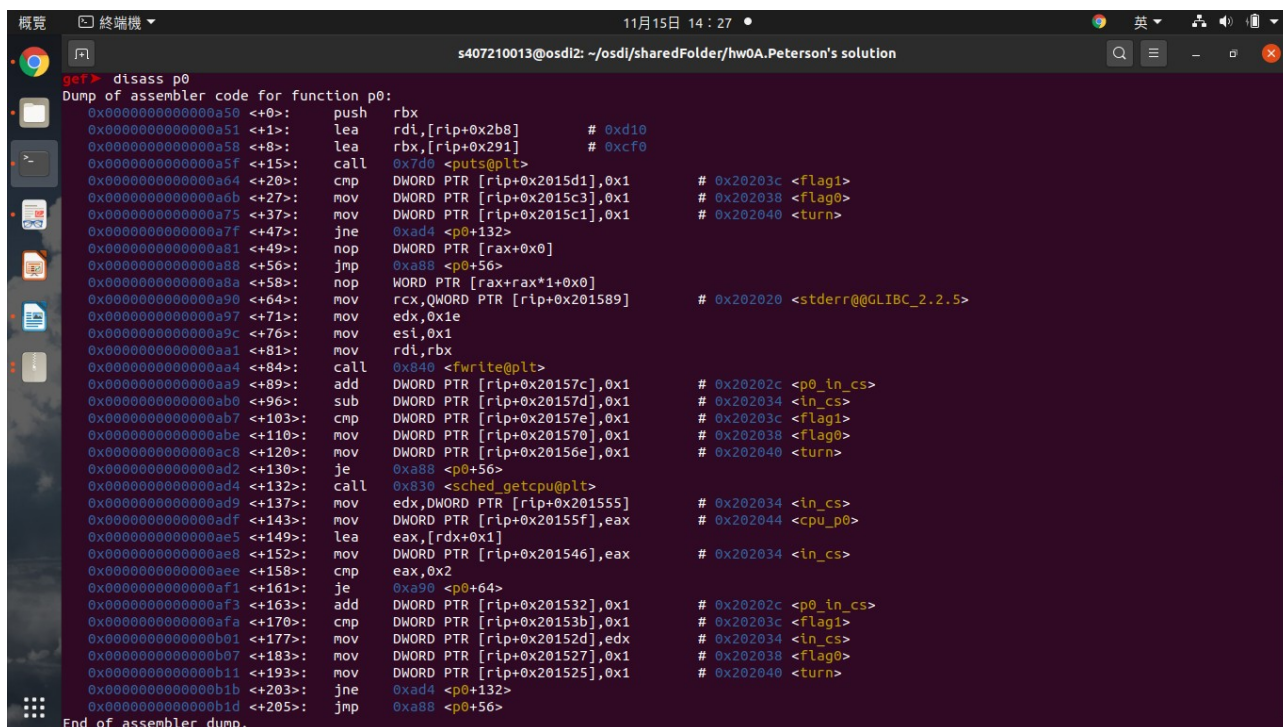


二. 解釋「為什麼」 peterson\_trival-O3 的執行結果是錯的?

在『while(flag == 1 && turn == 1)』的判斷式裡面我們使用 turn == 1，而因為在前面我們已經將 turn = 1，所以反組譯下去看，發現編譯器自動就把它優化掉了。



```
28 void per_second(int signum) {
29     printf("進入次數 (每秒) p0: %5d, p1: %5d", p0_in_cs, p1_in_cs);
30     printf(", 分別執行於 core#%d 及 core#%d\n", cpu_p0, cpu_p1);
31     p0_in_cs = 0;
32     p1_in_cs = 0;
33     alarm(1);
34 }
35
36 void p0(void) {
37     printf("p0: start\n");
38     while (1) {
39         // 🌳 🌲 🌴 🌵 🌶 🌱 🌿 🍀
40         // Peterson's solution的進去部分的程式碼
41         flag0 = 1;
42         turn = 1;
43         while (flag1==1 && turn==1)
44             ; //waiting
45
46         // 底下程式碼用於模擬在critical section
47         cpu_p0 = sched_getcpu();
48         in_cs++; // 計算有多少人在cs中
49         // nanosleep(&ts, NULL); // 睡100 nanoseconds
50         if (in_cs == 2) fprintf(stderr, "p0及p1都在critical section\n");
51         p0_in_cs++; // p0在cs幾次
52         // nanosleep(&ts, NULL); // 睡100 nanoseconds
53         in_cs--; // 計算有多少人在cs中
54
55         // 🌳 🌲 🌴 🌵 🌶 🌱 🌿 🍀
56         // Peterson's solution的離開部分的程式碼
57         flag0=0;
58     }
59 }
60
61 // [home/normalUsers/s407210013/osdl/sharedFolder/hw0A.Peterson's solution/peterson_trival.c][utf-8, unix, c][2KB] 0,35/95 [ 36%]
```



```
gef> disass p0
Dump of assembler code for function p0:
0x000000000000a50 <+0>: push    rbx
0x000000000000a51 <+1>: lea     rdi,[rip+0x2b8]          # 0xd10
0x000000000000a59 <+8>: lea     rbx,[rip+0x291]          # 0xcf0
0x000000000000a5f <+15>: call    0x7d0 <puts@plt>
0x000000000000a64 <+20>: cmp     DWORD PTR [rip+0x2015d1],0x1 # 0x20203c <flag1>
0x000000000000a6b <+27>: mov     DWORD PTR [rip+0x2015c3],0x1 # 0x202038 <flag0>
0x000000000000a75 <+37>: mov     DWORD PTR [rip+0x2015c1],0x1 # 0x202040 <turn>
0x000000000000a7f <+47>: jne     0xad4 <p0+132>
0x000000000000a81 <+49>: nop     DWORD PTR [rax+0x0]
0x000000000000a88 <+56>: jmp     0xa88 <p0+56>
0x000000000000a8a <+58>: nop     WORD PTR [rax+rax*1+0x0]
0x000000000000a90 <+64>: mov     rcx,QWORD PTR [rip+0x201589] # 0x202020 <stderr@@GLIBC_2.2.5>
0x000000000000a97 <+71>: mov     edx,0x1e
0x000000000000a9c <+76>: mov     esi,0x1
0x000000000000aa1 <+81>: mov     rdi,rbx
0x000000000000aa4 <+84>: call    0xb40 <fwrite@plt>
0x000000000000aa9 <+89>: add     DWORD PTR [rip+0x20157c],0x1 # 0x20202c <p0_in_cs>
0x000000000000aab <+96>: sub     DWORD PTR [rip+0x20157d],0x1 # 0x202034 <in_cs>
0x000000000000ab7 <+103>: cmp     DWORD PTR [rip+0x20157e],0x1 # 0x20203c <flag1>
0x000000000000abe <+110>: mov     DWORD PTR [rip+0x201570],0x1 # 0x202038 <flag0>
0x000000000000ac8 <+120>: mov     DWORD PTR [rip+0x20156e],0x1 # 0x202040 <turn>
0x000000000000ad2 <+130>: je      0xa88 <p0+56>
0x000000000000ad4 <+132>: call    0x830 <sched_getcpu@plt>
0x000000000000ad9 <+137>: mov     edx,DWORD PTR [rip+0x201555] # 0x202034 <in_cs>
0x000000000000adf <+143>: mov     DWORD PTR [rip+0x20155f],eax # 0x202044 <cpu_p0>
0x000000000000ae5 <+149>: lea     eax,[rdi+0x1]
0x000000000000aeb <+152>: mov     DWORD PTR [rip+0x201546],eax # 0x202034 <in_cs>
0x000000000000aee <+158>: cmp     eax,0x2
0x000000000000af1 <+161>: je      0xa98 <p0+64>
0x000000000000af3 <+163>: add     DWORD PTR [rip+0x201532],0x1 # 0x20202c <p0_in_cs>
0x000000000000afa <+170>: cmp     DWORD PTR [rip+0x20153b],0x1 # 0x20203c <flag1>
0x000000000000b01 <+177>: mov     DWORD PTR [rip+0x20152d],edx # 0x202034 <in_cs>
0x000000000000b07 <+183>: mov     DWORD PTR [rip+0x201527],0x1 # 0x202038 <flag0>
0x000000000000b11 <+193>: mov     DWORD PTR [rip+0x201525],0x1 # 0x202040 <turn>
0x000000000000b1b <+203>: jne     0xad4 <p0+132>
0x000000000000b1d <+205>: jmp     0xa88 <p0+56>
End of assembler dump.
```

三. 「peterson\_trival-g」速度比「peterson\_correct-O3」快或是慢？程式的正確與否？

「peterson\_trival-g」的速度比「peterson\_correct-O3」快。  
「peterson\_trival-g」是錯的，「peterson\_correct-O3」是對的。

```
s407210013@osdi2:~/osdi/sharedFolder/hw0A.Peterson's solution$ ./peterson_trival-g
p0: start
p1: start
進入次數 (每秒) p0: 4103941, p1: 4103203, 分別執行於 core#40 及 core#0
進入次數 (每秒) p0: 4110828, p1: 4110929, 分別執行於 core#40 及 core#0
進入次數 (每秒) p0: 4116468, p1: 4116509, 分別執行於 core#40 及 core#0
進入次數 (每秒) p0: 4124849, p1: 4124959, 分別執行於 core#40 及 core#0
進入次數 (每秒) p0: 8246525, p1: 4121573, 分別執行於 core#40 及 core#0
^C
s407210013@osdi2:~/osdi/sharedFolder/hw0A.Peterson's solution$ ./peterson_correct-O3
start p0
start p1
進入次數 (每秒) p0: 2039554, p1: 1944407, 分別執行於 core#0 及 core#40
進入次數 (每秒) p0: 2337706, p1: 2189681, 分別執行於 core#0 及 core#40
進入次數 (每秒) p0: 2333938, p1: 2189117, 分別執行於 core#0 及 core#40
進入次數 (每秒) p0: 2343611, p1: 2205768, 分別執行於 core#0 及 core#40
進入次數 (每秒) p0: 2338497, p1: 2187995, 分別執行於 core#0 及 core#40
進入次數 (每秒) p0: 2335519, p1: 2193213, 分別執行於 core#0 及 core#40
^C
s407210013@osdi2:~/osdi/sharedFolder/hw0A.Peterson's solution$ □
```

四. 解釋「peterson\_trival-g」的速度比「peterson\_correct-O3」快的理由。  
因為「peterson\_trival-g」每秒進入的次數大約是 400 萬次上下，而每秒進入的次數大約是 200 萬次上下，因此「peterson\_trival-g」比「peterson\_correct-O3」速度還快。