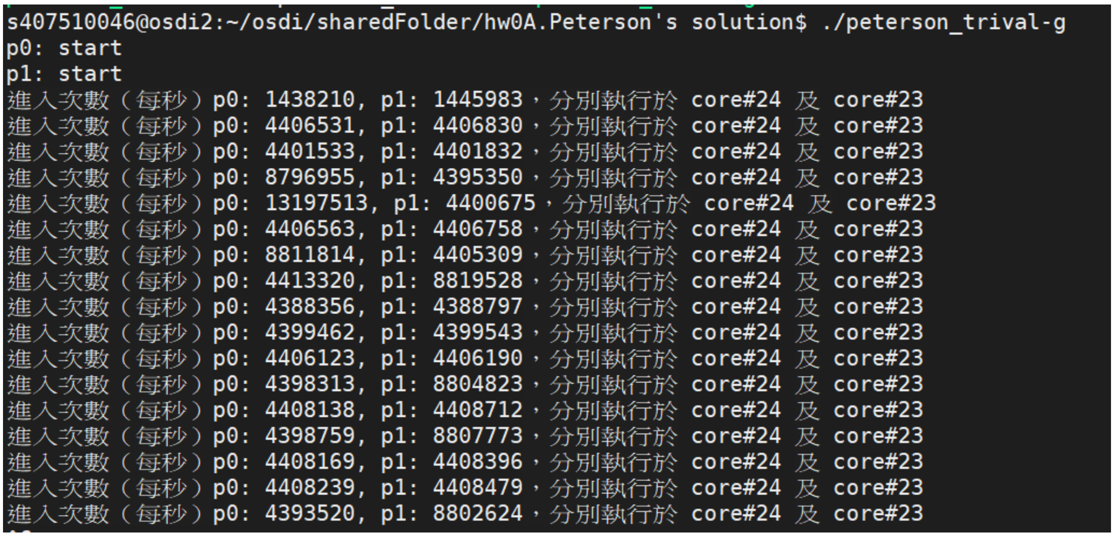
作業系統作業六

學號：407510046 姓名：黃X雯

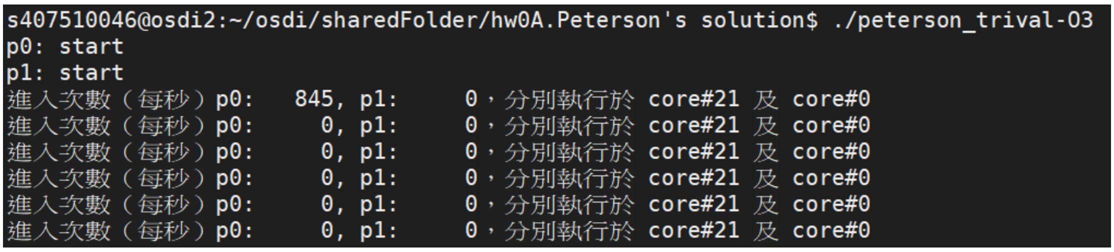
1.

peterson_trival -g



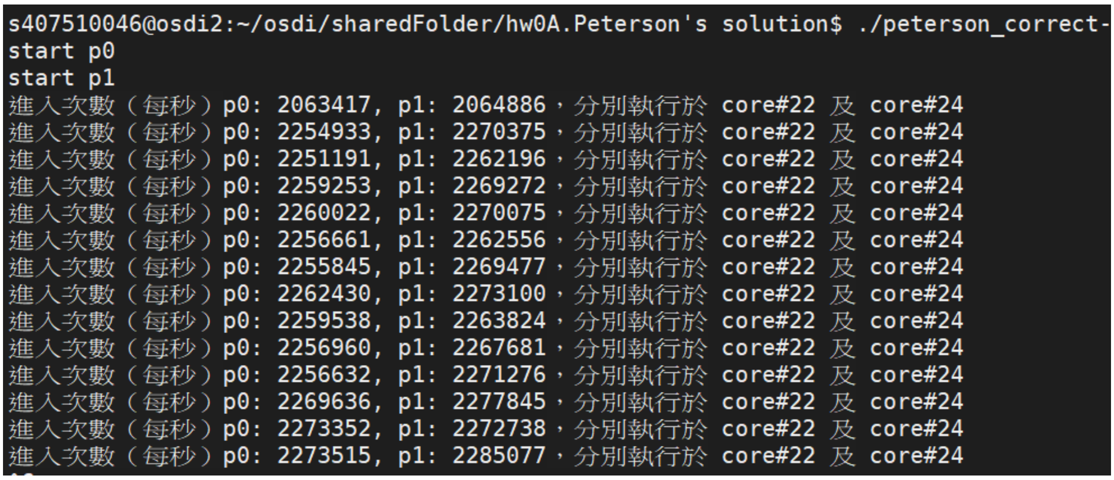
```
s407510046@osdi2:~/osdi/sharedFolder/hw0A.Peterson's solution$ ./peterson_trival-g
p0: start
p1: start
進入次數（每秒）p0: 1438210, p1: 1445983，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4406531, p1: 4406830，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4401533, p1: 4401832，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 8796955, p1: 4395350，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 13197513, p1: 4400675，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4406563, p1: 4406758，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 8811814, p1: 4405309，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4413320, p1: 8819528，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4388356, p1: 4388797，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4399462, p1: 4399543，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4406123, p1: 4406190，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4398313, p1: 8804823，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4408138, p1: 4408712，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4398759, p1: 8807773，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4408169, p1: 4408396，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4408239, p1: 4408479，分別執行於 core#24 及 core#23
進入次數（每秒）p0: 4393520, p1: 8802624，分別執行於 core#24 及 core#23
```

peterson_trival -O3



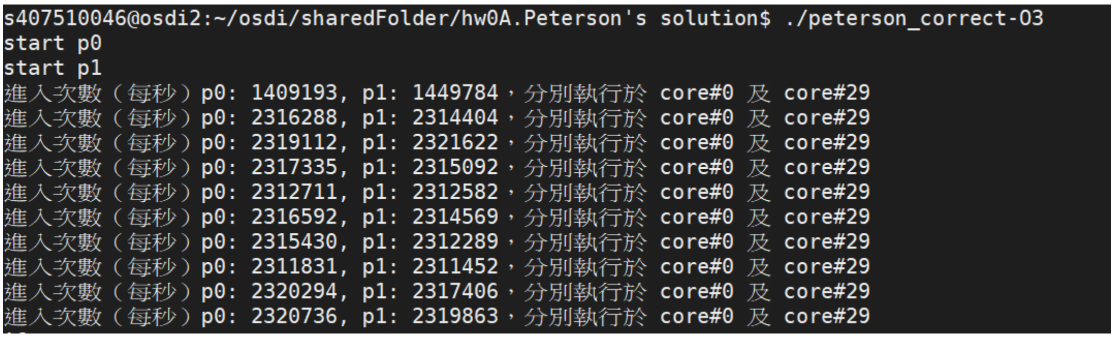
```
s407510046@osdi2:~/osdi/sharedFolder/hw0A.Peterson's solution$ ./peterson_trival-O3
p0: start
p1: start
進入次數（每秒）p0:   845, p1:     0，分別執行於 core#21 及 core#0
進入次數（每秒）p0:     0, p1:     0，分別執行於 core#21 及 core#0
進入次數（每秒）p0:     0, p1:     0，分別執行於 core#21 及 core#0
進入次數（每秒）p0:     0, p1:     0，分別執行於 core#21 及 core#0
進入次數（每秒）p0:     0, p1:     0，分別執行於 core#21 及 core#0
進入次數（每秒）p0:     0, p1:     0，分別執行於 core#21 及 core#0
```

peterson_correct -g



```
s407510046@osdi2:~/osdi/sharedFolder/hw0A.Peterson's solution$ ./peterson_correct-
start p0
start p1
進入次數（每秒）p0: 2063417, p1: 2064886，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2254933, p1: 2270375，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2251191, p1: 2262196，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2259253, p1: 2269272，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2260022, p1: 2270075，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2256661, p1: 2262556，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2255845, p1: 2269477，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2262430, p1: 2273100，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2259538, p1: 2263824，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2256960, p1: 2267681，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2256632, p1: 2271276，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2269636, p1: 2277845，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2273352, p1: 2272738，分別執行於 core#22 及 core#24
進入次數（每秒）p0: 2273515, p1: 2285077，分別執行於 core#22 及 core#24
```

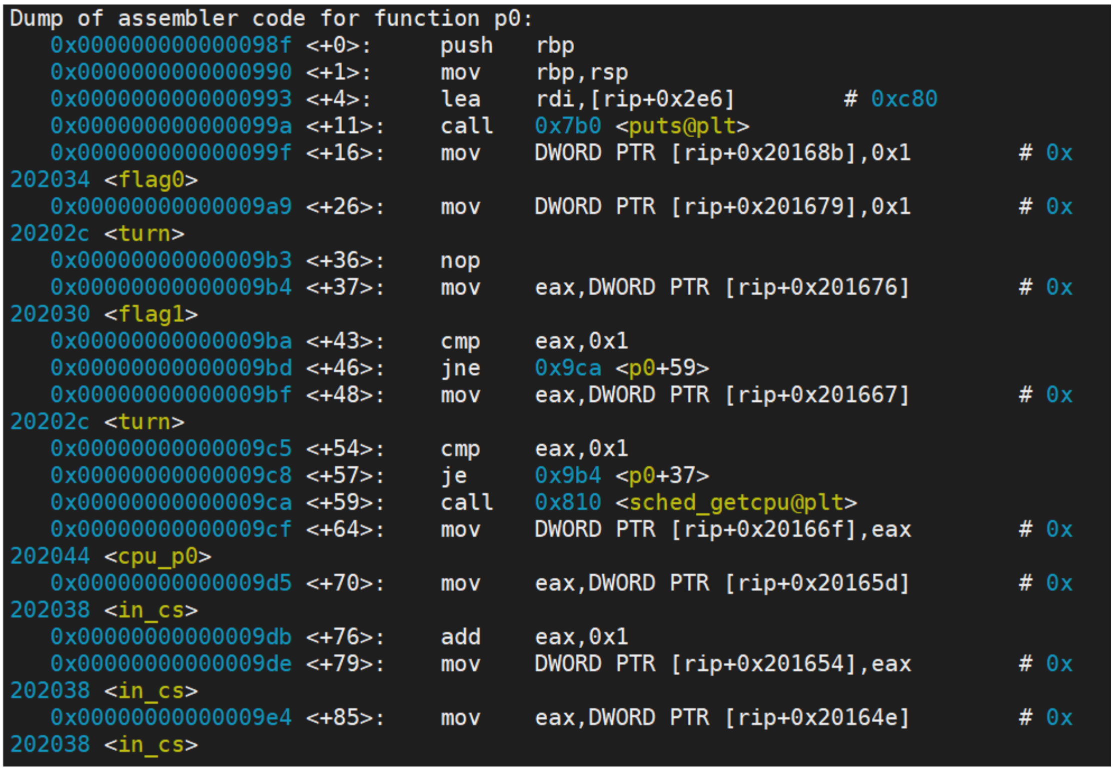
peterson_correct -O3



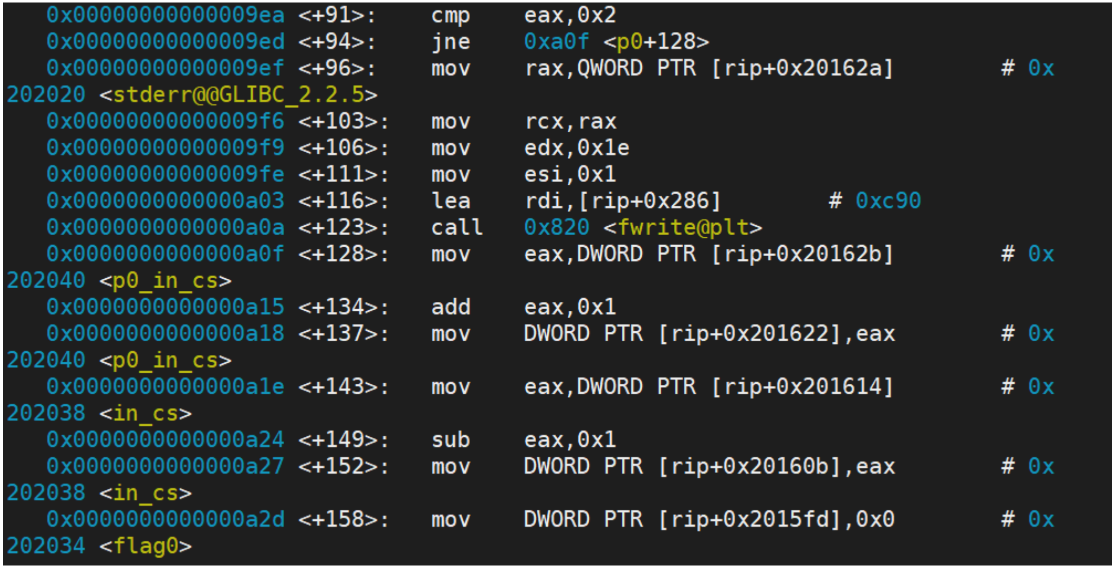
```
s407510046@osdi2:~/osdi/sharedFolder/hw0A.Peterson's solution$ ./peterson_correct-O3
start p0
start p1
進入次數（每秒）p0: 1409193, p1: 1449784，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2316288, p1: 2314404，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2319112, p1: 2321622，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2317335, p1: 2315092，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2312711, p1: 2312582，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2316592, p1: 2314569，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2315430, p1: 2312289，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2311831, p1: 2311452，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2320294, p1: 2317406，分別執行於 core#0 及 core#29
進入次數（每秒）p0: 2320736, p1: 2319863，分別執行於 core#0 及 core#29
```

2.

gdb ./peterson_trival -g   disassemble p0

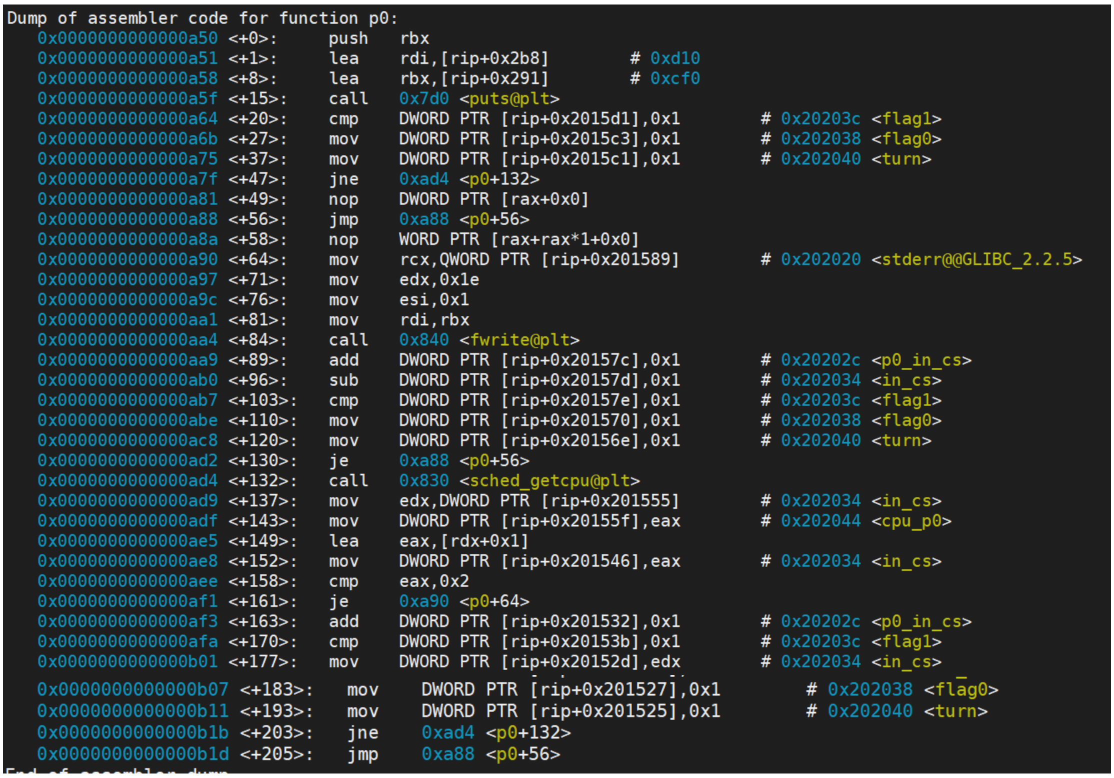


```
Dump of assembler code for function p0:
   0x000000000000098f <+0>:     push   rbp
   0x0000000000000990 <+1>:     mov    rbp,rsp
   0x0000000000000993 <+4>:     lea    rdi,[rip+0x2e6]        # 0xc80
   0x000000000000099a <+11>:    call   0x7b0 <puts@plt>
   0x000000000000099f <+16>:    mov    DWORD PTR [rip+0x20168b],0x1        # 0x
202034 <flag0>
   0x00000000000009a9 <+26>:    mov    DWORD PTR [rip+0x201679],0x1        # 0x
20202c <turn>
   0x00000000000009b3 <+36>:    nop
   0x00000000000009b4 <+37>:    mov    eax,DWORD PTR [rip+0x201676]        # 0x
202030 <flag1>
   0x00000000000009ba <+43>:    cmp    eax,0x1
   0x00000000000009bd <+46>:    jne    0x9ca <p0+59>
   0x00000000000009bf <+48>:    mov    eax,DWORD PTR [rip+0x201667]        # 0x
20202c <turn>
   0x00000000000009c5 <+54>:    cmp    eax,0x1
   0x00000000000009c8 <+57>:    je     0x9b4 <p0+37>
   0x00000000000009ca <+59>:    call   0x810 <sched_getcpu@plt>
   0x00000000000009cf <+64>:    mov    DWORD PTR [rip+0x20166f],eax        # 0x
202044 <cpu_p0>
   0x00000000000009d5 <+70>:    mov    eax,DWORD PTR [rip+0x20165d]        # 0x
202038 <in_cs>
   0x00000000000009db <+76>:    add    eax,0x1
   0x00000000000009de <+79>:    mov    DWORD PTR [rip+0x201654],eax        # 0x
202038 <in_cs>
   0x00000000000009e4 <+85>:    mov    eax,DWORD PTR [rip+0x20164e]        # 0x
202038 <in_cs>
```



```
   0x00000000000009ea <+91>:    cmp    eax,0x2
   0x00000000000009ed <+94>:    jne    0xa0f <p0+128>
   0x00000000000009ef <+96>:    mov    rax,QWORD PTR [rip+0x20162a]        # 0x
202020 <stderr@@GLIBC_2.2.5>
   0x00000000000009f6 <+103>:   mov    rcx,rax
   0x00000000000009f9 <+106>:   mov    edx,0x1e
   0x00000000000009fe <+111>:   mov    esi,0x1
   0x0000000000000a03 <+116>:   lea    rdi,[rip+0x286]        # 0xc90
   0x0000000000000a0a <+123>:   call   0x820 <fwrite@plt>
   0x0000000000000a0f <+128>:   mov    eax,DWORD PTR [rip+0x20162b]        # 0x
202040 <p0_in_cs>
   0x0000000000000a15 <+134>:   add    eax,0x1
   0x0000000000000a18 <+137>:   mov    DWORD PTR [rip+0x201622],eax        # 0x
202040 <p0_in_cs>
   0x0000000000000a1e <+143>:   mov    eax,DWORD PTR [rip+0x201614]        # 0x
202038 <in_cs>
   0x0000000000000a24 <+149>:   sub    eax,0x1
   0x0000000000000a27 <+152>:   mov    DWORD PTR [rip+0x20160b],eax        # 0x
202038 <in_cs>
   0x0000000000000a2d <+158>:   mov    DWORD PTR [rip+0x2015fd],0x0        # 0x
202034 <flag0>
```

gdb ./peterson_trival -O3   disassemble p0



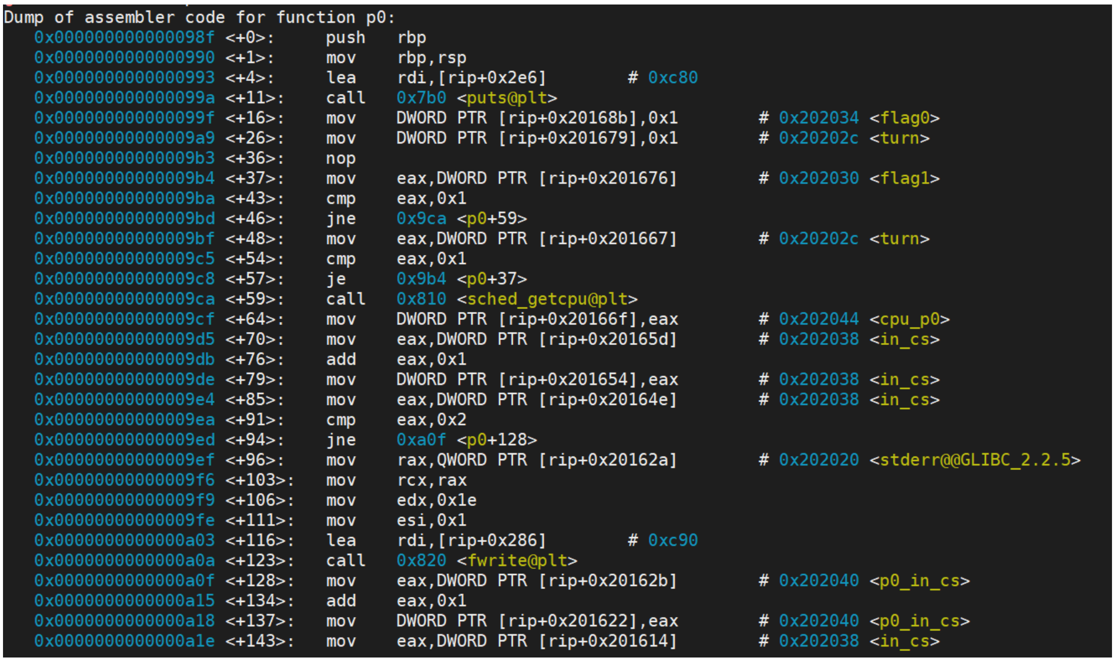
```
Dump of assembler code for function p0:
   0x0000000000000a50 <+0>:     push   rbx
   0x0000000000000a51 <+1>:     lea    rdi,[rip+0x2b8]        # 0xd10
   0x0000000000000a58 <+8>:     lea    rbx,[rip+0x291]        # 0xcf0
   0x0000000000000a5f <+15>:    call   0x7d0 <puts@plt>
   0x0000000000000a64 <+20>:    cmp    DWORD PTR [rip+0x2015d1],0x1        # 0x20203c <flag1>
   0x0000000000000a6b <+27>:    mov    DWORD PTR [rip+0x2015c3],0x1        # 0x202038 <flag0>
   0x0000000000000a75 <+37>:    mov    DWORD PTR [rip+0x2015c1],0x1        # 0x202040 <turn>
   0x0000000000000a7f <+47>:    jne    0xad4 <p0+132>
   0x0000000000000a81 <+49>:    nop    DWORD PTR [rax+0x0]
   0x0000000000000a88 <+56>:    jmp    0xa88 <p0+56>
   0x0000000000000a8a <+58>:    nop    WORD PTR [rax+rax*1+0x0]
   0x0000000000000a90 <+64>:    mov    rcx,QWORD PTR [rip+0x201589]        # 0x202020 <stderr@@GLIBC_2.2.5>
   0x0000000000000a97 <+71>:    mov    edx,0x1e
   0x0000000000000a9c <+76>:    mov    esi,0x1
   0x0000000000000aa1 <+81>:    mov    rdi,rbx
   0x0000000000000aa4 <+84>:    call   0x840 <fwrite@plt>
   0x0000000000000aa9 <+89>:    add    DWORD PTR [rip+0x20157c],0x1        # 0x20202c <p0_in_cs>
   0x0000000000000ab0 <+96>:    sub    DWORD PTR [rip+0x20157d],0x1        # 0x202034 <in_cs>
   0x0000000000000ab7 <+103>:   cmp    DWORD PTR [rip+0x20157e],0x1        # 0x20203c <flag1>
   0x0000000000000abe <+110>:   mov    DWORD PTR [rip+0x201570],0x1        # 0x202038 <flag0>
   0x0000000000000ac8 <+120>:   mov    DWORD PTR [rip+0x20156e],0x1        # 0x202040 <turn>
   0x0000000000000ad2 <+130>:   je     0xa88 <p0+56>
   0x0000000000000ad4 <+132>:   call   0x830 <sched_getcpu@plt>
   0x0000000000000ad9 <+137>:   mov    edx,DWORD PTR [rip+0x201555]        # 0x202034 <in_cs>
   0x0000000000000adf <+143>:   mov    DWORD PTR [rip+0x20155f],eax        # 0x202044 <cpu_p0>
   0x0000000000000ae5 <+149>:   lea    eax,[rdx+0x1]
   0x0000000000000ae8 <+152>:   mov    DWORD PTR [rip+0x201546],eax        # 0x202034 <in_cs>
   0x0000000000000aee <+158>:   cmp    eax,0x2
   0x0000000000000af1 <+161>:   je     0xa90 <p0+64>
   0x0000000000000af3 <+163>:   add    DWORD PTR [rip+0x201532],0x1        # 0x20202c <p0_in_cs>
   0x0000000000000afa <+170>:   cmp    DWORD PTR [rip+0x20153b],0x1        # 0x20203c <flag1>
   0x0000000000000b01 <+177>:   mov    DWORD PTR [rip+0x20152d],edx        # 0x202034 <in_cs>
   0x0000000000000b07 <+183>:   mov    DWORD PTR [rip+0x201527],0x1        # 0x202038 <flag0>
   0x0000000000000b11 <+193>:   mov    DWORD PTR [rip+0x201525],0x1        # 0x202040 <turn>
   0x0000000000000b1b <+203>:   jne    0xad4 <p0+132>
   0x0000000000000b1d <+205>:   jmp    0xa88 <p0+56>
End of assembler dump.
```

在反組譯-O3 優化時沒有在進入 while 時判斷 flag 0，導致編譯器認為 flag 0，所以導致執行結果錯誤。

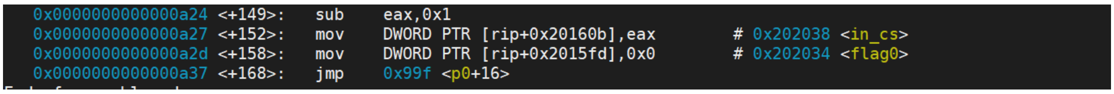
3&4.

peterson_trival -g 比 peterson_correct -O3 快

peterson_trival -g 反組譯：

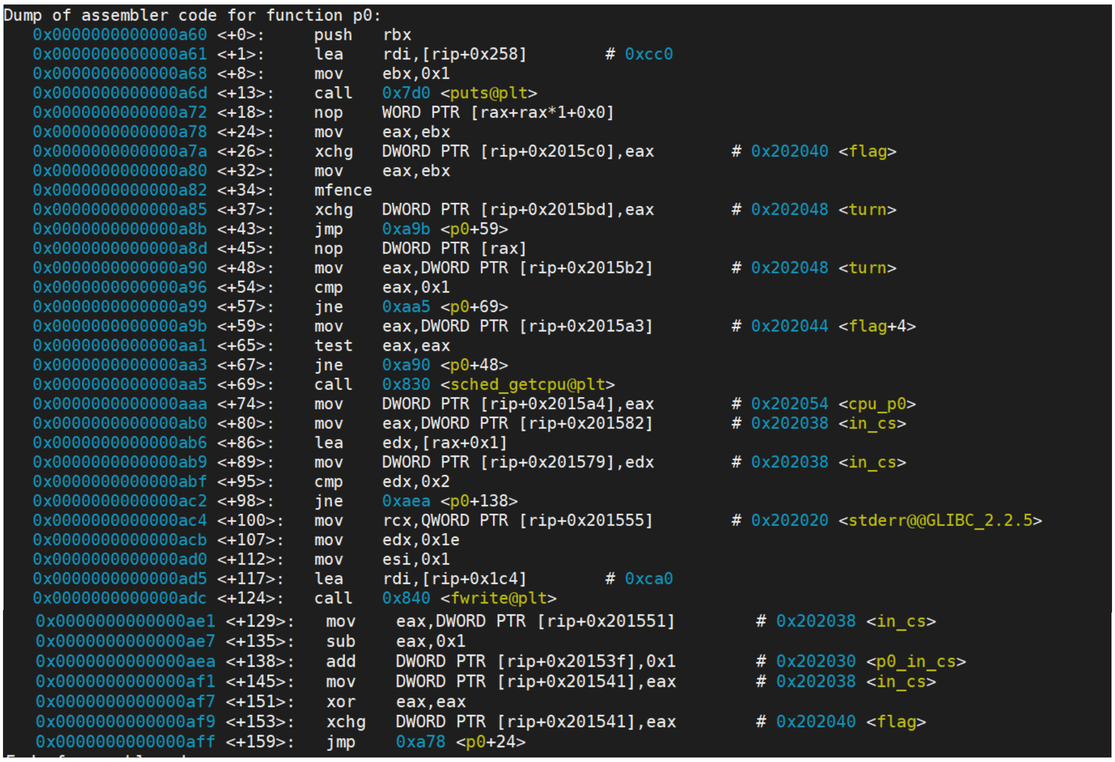


```
Dump of assembler code for function p0:
   0x000000000000098f <+0>:     push   rbp
   0x0000000000000990 <+1>:     mov    rbp,rsp
   0x0000000000000993 <+4>:     lea    rdi,[rip+0x2e6]        # 0xc80
   0x000000000000099a <+11>:    call   0x7b0 <puts@plt>
   0x000000000000099f <+16>:    mov    DWORD PTR [rip+0x20168b],0x1        # 0x202034 <flag0>
   0x00000000000009a9 <+26>:    mov    DWORD PTR [rip+0x201679],0x1        # 0x20202c <turn>
   0x00000000000009b3 <+36>:    nop
   0x00000000000009b4 <+37>:    mov    eax,DWORD PTR [rip+0x201676]        # 0x202030 <flag1>
   0x00000000000009ba <+43>:    cmp    eax,0x1
   0x00000000000009bd <+46>:    jne    0x9ca <p0+59>
   0x00000000000009bf <+48>:    mov    eax,DWORD PTR [rip+0x201667]        # 0x20202c <turn>
   0x00000000000009c5 <+54>:    cmp    eax,0x1
   0x00000000000009c8 <+57>:    je     0x9b4 <p0+37>
   0x00000000000009ca <+59>:    call   0x810 <sched_getcpu@plt>
   0x00000000000009cf <+64>:    mov    DWORD PTR [rip+0x20166f],eax        # 0x202044 <cpu_p0>
   0x00000000000009d5 <+70>:    mov    eax,DWORD PTR [rip+0x20165d]        # 0x202038 <in_cs>
   0x00000000000009db <+76>:    add    eax,0x1
   0x00000000000009de <+79>:    mov    DWORD PTR [rip+0x201654],eax        # 0x202038 <in_cs>
   0x00000000000009e4 <+85>:    mov    eax,DWORD PTR [rip+0x20164e]        # 0x202038 <in_cs>
   0x00000000000009ea <+91>:    cmp    eax,0x2
   0x00000000000009ed <+94>:    jne    0xa0f <p0+128>
   0x00000000000009ef <+96>:    mov    rax,QWORD PTR [rip+0x20162a]        # 0x202020 <stderr@@GLIBC_2.2.5>
   0x00000000000009f6 <+103>:   mov    rcx,rax
   0x00000000000009f9 <+106>:   mov    edx,0x1e
   0x00000000000009fe <+111>:   mov    esi,0x1
   0x0000000000000a03 <+116>:   lea    rdi,[rip+0x286]        # 0xc90
   0x0000000000000a0a <+123>:   call   0x820 <fwrite@plt>
   0x0000000000000a0f <+128>:   mov    eax,DWORD PTR [rip+0x20162b]        # 0x202040 <p0_in_cs>
   0x0000000000000a15 <+134>:   add    eax,0x1
   0x0000000000000a18 <+137>:   mov    DWORD PTR [rip+0x201622],eax        # 0x202040 <p0_in_cs>
   0x0000000000000a1e <+143>:   mov    eax,DWORD PTR [rip+0x201614]        # 0x202038 <in_cs>
```



```
   0x0000000000000a24 <+149>:   sub    eax,0x1
   0x0000000000000a27 <+152>:   mov    DWORD PTR [rip+0x20160b],eax        # 0x202038 <in_cs>
   0x0000000000000a2d <+158>:   mov    DWORD PTR [rip+0x2015fd],0x0        # 0x202034 <flag0>
   0x0000000000000a37 <+168>:   jmp    0x99f <p0+16>
```

peterson_correct -O3 反組譯：



```
Dump of assembler code for function p0:
   0x0000000000000a60 <+0>:     push   rbx
   0x0000000000000a61 <+1>:     lea    rdi,[rip+0x258]        # 0xcc0
   0x0000000000000a68 <+8>:     mov    ebx,0x1
   0x0000000000000a6d <+13>:    call   0x7d0 <puts@plt>
   0x0000000000000a72 <+18>:    nop    WORD PTR [rax+rax*1+0x0]
   0x0000000000000a78 <+24>:    mov    eax,ebx
   0x0000000000000a7a <+26>:    xchg   DWORD PTR [rip+0x2015c0],eax        # 0x202040 <flag>
   0x0000000000000a80 <+32>:    mov    eax,ebx
   0x0000000000000a82 <+34>:    mfence
   0x0000000000000a85 <+37>:    xchg   DWORD PTR [rip+0x2015bd],eax        # 0x202048 <turn>
   0x0000000000000a8b <+43>:    jmp    0xa9b <p0+59>
   0x0000000000000a8d <+45>:    nop    DWORD PTR [rax]
   0x0000000000000a90 <+48>:    mov    eax,DWORD PTR [rip+0x2015b2]        # 0x202048 <turn>
   0x0000000000000a96 <+54>:    cmp    eax,0x1
   0x0000000000000a99 <+57>:    jne    0xaa5 <p0+69>
   0x0000000000000a9b <+59>:    mov    eax,DWORD PTR [rip+0x2015a3]        # 0x202044 <flag+4>
   0x0000000000000aa1 <+65>:    test   eax,eax
   0x0000000000000aa3 <+67>:    jne    0xa90 <p0+48>
   0x0000000000000aa5 <+69>:    call   0x830 <sched_getcpu@plt>
   0x0000000000000aaa <+74>:    mov    DWORD PTR [rip+0x2015a4],eax        # 0x202054 <cpu_p0>
   0x0000000000000ab0 <+80>:    mov    eax,DWORD PTR [rip+0x201582]        # 0x202038 <in_cs>
   0x0000000000000ab6 <+86>:    lea    edx,[rax+0x1]
   0x0000000000000ab9 <+89>:    mov    DWORD PTR [rip+0x201579],edx        # 0x202038 <in_cs>
   0x0000000000000abf <+95>:    cmp    edx,0x2
   0x0000000000000ac2 <+98>:    jne    0xaea <p0+138>
   0x0000000000000ac4 <+100>:   mov    rcx,QWORD PTR [rip+0x201555]        # 0x202020 <stderr@@GLIBC_2.2.5>
   0x0000000000000acb <+107>:   mov    edx,0x1e
   0x0000000000000ad0 <+112>:   mov    esi,0x1
   0x0000000000000ad5 <+117>:   lea    rdi,[rip+0x1c4]        # 0xca0
   0x0000000000000adc <+124>:   call   0x840 <fwrite@plt>
   0x0000000000000ae1 <+129>:   mov    eax,DWORD PTR [rip+0x201551]        # 0x202038 <in_cs>
   0x0000000000000ae7 <+135>:   sub    eax,0x1
   0x0000000000000aea <+138>:   add    DWORD PTR [rip+0x20153f],0x1        # 0x202030 <p0_in_cs>
   0x0000000000000af1 <+145>:   mov    DWORD PTR [rip+0x201541],eax        # 0x202038 <in_cs>
   0x0000000000000af7 <+151>:   xor    eax,eax
   0x0000000000000af9 <+153>:   xchg   DWORD PTR [rip+0x201541],eax        # 0x202040 <flag>
   0x0000000000000aff <+159>:   jmp    0xa78 <p0+24>
```

兩個最大的差別應該是差在 peterson_trival 沒有做 mfence，因為 mfence 可以保障資料的正確性，以及按照 CPU 和記憶體的順序執行，因此 mfence 應該是導致 peterson_correct -O3 效能比較差的原因。