

## HW7 實現 Peterson's solution

系級：資工三 學號：409410114 姓名：周述君

甲、說明你的執行環境（請自己想一下，哪些會和結果的正確語法有關，列出有關的執行環境就好）。

- gcc version 7.5.0
- Intel(R) Core(TM) i5-1035G4 CPU @ 1.10GHz 1.50 GHz

```
sujean@vm:~/osdi/hw7$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured With: ../src/configure -v --with-pkgversion='Ubuntu 7.5.0-6ubuntu2' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable
-languagesc,ada,c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --program-suffix=-7 --program-prefix=x86_64-l
inux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib
--enable-nls --enable-bootstrap --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --
enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --enable-plugin --enable-default-pie --with-system-zlib --with-target-system
-zlib --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch=32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable
-multilib --with-tune=generic --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --
host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 7.5.0 (Ubuntu 7.5.0-6ubuntu2)
sujean@vm:~/osdi/hw7$ cat /proc/cpuinfo | grep 'model name' | uniq
model name      : Intel(R) Core(TM) i5-1035G4 CPU @ 1.10GHz
```

乙、請將題目中的四個小題的答案寫在文件中，繳交的形式必須為 pdf 檔案。

1. 執行 make，之後會產生四個執行檔案。請問你的執行結果為何？請附上畫面截圖

- make 後產生的四個執行檔

```
sujean@vm:~/osdi/hw7$ make
gcc -pthread -O3 peterson_correct.c -o peterson_correct-O3
gcc -pthread -g peterson_correct.c -o peterson_correct-g
gcc -pthread -O3 peterson_trival.c -o peterson_trival-O3
gcc -pthread -g peterson_trival.c -o peterson_trival-g
```

- ./peterson\_correct-O3

```
sujean@vm:~/osdi/hw7$ ./peterson_correct-O3
start p0
進入次數 (每秒) p0: 6820122, p1: 6813024, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6768861, p1: 6917053, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6727223, p1: 6716803, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7197764, p1: 7191406, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7432089, p1: 7297481, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7272017, p1: 7231493, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7741092, p1: 7742665, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7710297, p1: 7670643, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7221981, p1: 7214569, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 8268378, p1: 8234594, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6374060, p1: 6372017, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7732858, p1: 7741903, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7882269, p1: 8045099, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6925026, p1: 6927038, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7662986, p1: 7662790, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6421350, p1: 6406552, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 8321704, p1: 8340736, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6821413, p1: 6854858, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 7101475, p1: 7120170, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4488357, p1: 4509474, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4727734, p1: 4543028, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4298975, p1: 4180702, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 3668081, p1: 3673793, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 3731629, p1: 3715217, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 3618151, p1: 3626140, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 2773751, p1: 2718855, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 2451325, p1: 2481210, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 2695972, p1: 2672771, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 3378620, p1: 3366235, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5905093, p1: 6012810, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4278797, p1: 4258875, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4784755, p1: 4869216, 分別執行於 core#0 及 core#1
```

- `./peterson_correct-g`

```

[~] Terminal v... Nov 4 15:44
shihao@vm:~/osdi/hw5$ ./peterson_correct-g
start p0
start p1
進入次數 (每秒) p0: 5248504, p1: 5237845, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6429884, p1: 6453227, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5628017, p1: 5621980, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6463525, p1: 6454405, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6309336, p1: 6309064, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5878748, p1: 5912949, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5954548, p1: 5929788, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6049999, p1: 6038284, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6516115, p1: 6478102, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5660484, p1: 5644711, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6423581, p1: 6531267, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5677707, p1: 5683289, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6260371, p1: 6261632, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6226086, p1: 6303161, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6156066, p1: 6157690, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5590948, p1: 5575937, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6312079, p1: 6336975, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5790908, p1: 5711460, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4682639, p1: 4738065, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5785782, p1: 5813220, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5962422, p1: 6103409, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5793328, p1: 5804703, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6045268, p1: 6035246, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5545632, p1: 5631062, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6184046, p1: 6250802, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6063865, p1: 6068055, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6329915, p1: 6319491, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5589934, p1: 5588865, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5754300, p1: 5773641, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5676637, p1: 5707420, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5684889, p1: 5706540, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5037927, p1: 5072111, 分別執行於 core#0 及 core#1

```

- ./peterson\_trival-03

[illegible]

- `./peterson_trival-g`

[illegible]

- ```

shh@kali:~/p0j4/mut$ (gdb) disassemble p0
Dump of assembler code for function p0:
0x0000000000001200 <+0>: push    %rbx
0x0000000000001201 <+1>: lea    0xd97(%rip),%rdi    # 0x207f
0x0000000000001208 <+8>: lea    0xd71(%rip),%rbx    # 0x2060
0x00000000000012ef <+15>: callq  0x1040 <puts@plt>
0x00000000000012f4 <+20>: nopl    0x0(%rax)
0x00000000000012f8 <+24>: cmpl   $0x1,0x2d3d(%rip)    # 0x403c <flag1>
0x00000000000012ff <+31>: movl   $0x1,0x2d2f(%rip)    # 0x4038 <flag0>
0x0000000000001309 <+41>: movl   $0x1,0x2d2d(%rip)    # 0x4040 <turn>
0x0000000000001313 <+51>: jne     0x1320 <p0+64>
0x0000000000001315 <+53>: jmp     0x1315 <p0+53>
0x0000000000001317 <+55>: nopw    0x0(%rax,%rax,1)
0x0000000000001320 <+64>: callq  0x10a0 <scanf_getcpu@plt>
0x0000000000001325 <+69>: mov     %eax,0x2d19(%rip)    # 0x4044 <cpu_p0>
0x0000000000001326 <+75>: mov     0x2d03(%rip),%eax    # 0x4034 <in_cs>
0x0000000000001331 <+81>: add     $0x1,%eax
0x0000000000001334 <+84>: cmp     $0x2,%eax
0x0000000000001337 <+87>: mov     %eax,0x2cf7(%rip)    # 0x4034 <in_cs>
0x000000000000133d <+93>: je       0x1358 <p0+120>
0x000000000000133f <+95>: sub     $0x1,%eax
0x0000000000001342 <+98>: addl    $0x1,0x2ce3(%rip)    # 0x402c <p0_in_cs>
0x0000000000001349 <+105>: mov     %eax,0x2ce5(%rip)    # 0x4034 <in_cs>
0x000000000000134f <+111>: jmp     0x12f8 <p0+24>
0x0000000000001351 <+113>: nopl    0x0(%rax)
0x0000000000001358 <+120>: mov     0x2cc1(%rip),%rcx    # 0x4020 <stderr@GLIBC_2.2.5>
0x000000000000135f <+127>: mov     $0x1,%eax
0x0000000000001364 <+132>: mov     $0x1,%esi
0x0000000000001369 <+137>: mov     %rbx,%rdi
0x000000000000136c <+140>: callq  0x1080 <fwrite@plt>
0x0000000000001371 <+145>: mov     0x2cbd(%rip),%eax    # 0x4034 <in_cs>
0x0000000000001377 <+151>: jmp     0x133f <p0+95>

End of assembler dump.
(gdb)

```

- peterson\_trival-g

[illegible]

- peterson\_correct-03

```

# sujean@vm:~/ssdi/hw7$ ./peterson_correct-03
start p1
進入次數 (每秒) p0: 5847792, p1: 5903751, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5014557, p1: 5017136, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4765313, p1: 4722645, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4693366, p1: 4673095, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4751919, p1: 4834748, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5255527, p1: 5208292, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5519721, p1: 5458671, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6479741, p1: 6454526, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5740163, p1: 5750763, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6397969, p1: 6397939, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5459196, p1: 5597882, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5493976, p1: 5956940, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 4875085, p1: 4673095, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5798879, p1: 6815228, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6320800, p1: 6322089, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5333939, p1: 5328719, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 5627874, p1: 5612765, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6817318, p1: 5713181, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6887825, p1: 6860107, 分別執行於 core#0 及 core#1
進入次數 (每秒) p0: 6281095, p1: 6285721, 分別執行於 core#0 及 core#1

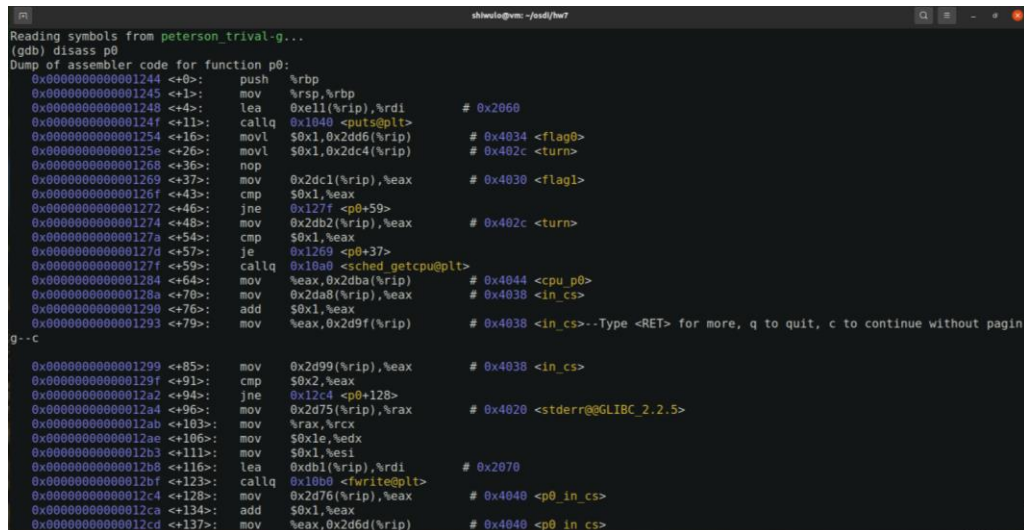
```



Peterson\_trival-g 速度比較快，根據上圖結果 peterson\_trival-g 執行一下 p0 和 p1 就進入 critical section 了，所以是錯的，而 peterson\_correct-O3 正常回報每秒 p0 和 p1 進入 critical section 的次數，結果正確。

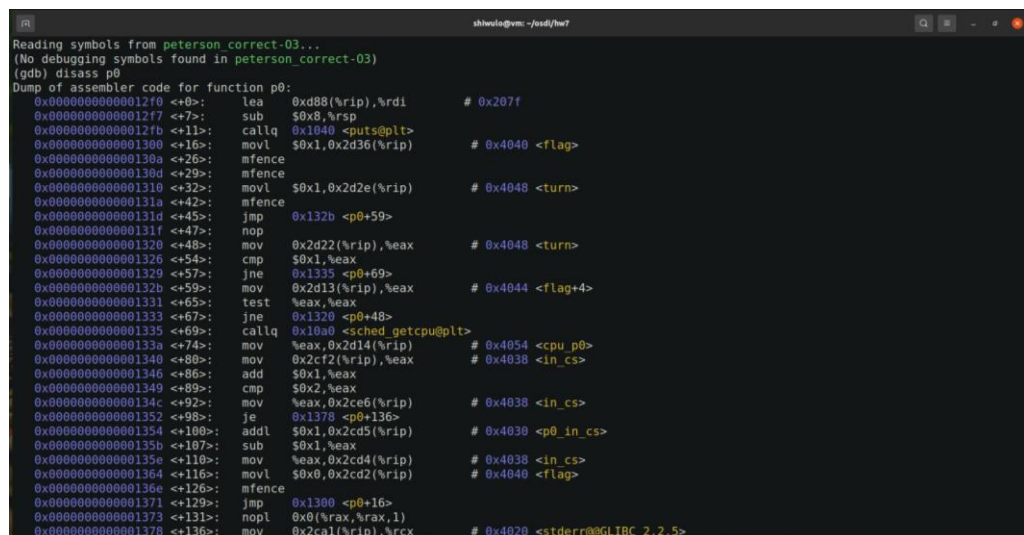
4. 請「確實的」解釋「題三」，某個程式比另一個程式快或者慢的理由。  
提示：『確實的』我建議使用組合語言和計算機結構來解釋

- peterson\_trival-g



```
Reading symbols from peterson_trival-g...
(gdb) disass p0
Dump of assembler code for function p0:
0x000000000001244 <+0>: push %rbp
0x000000000001245 <+1>: mov %rsp,%rbp
0x000000000001248 <+4>: lea 0x1(%rip),%rdi # 0x2060
0x00000000000124f <+11>: callq 0x1040 <puts@plt>
0x000000000001254 <+16>: movl $0x1,0x2dd6(%rip) # 0x4034 <flag0>
0x00000000000125e <+26>: movl $0x1,0x2dc4(%rip) # 0x402c <turn>
0x000000000001268 <+36>: nop
0x000000000001269 <+37>: mov 0x2dc1(%rip),%eax # 0x4030 <flag1>
0x00000000000126a <+38>: cmp $0x1,%eax
0x00000000000126b <+39>: jne 0x127f <+0+59>
0x000000000001274 <+48>: mov 0x2db2(%rip),%eax # 0x402c <turn>
0x00000000000127a <+54>: cmp $0x1,%eax
0x00000000000127d <+57>: je 0x1269 <+0+37>
0x00000000000127f <+59>: callq 0x10a0 <sched_getcpu@plt>
0x000000000001284 <+64>: mov %eax,0x2dba(%rip) # 0x4044 <cpu p0>
0x00000000000128a <+70>: mov 0x2da8(%rip),%eax # 0x4038 <in_cs>
0x000000000001290 <+76>: add $0x1,%eax
0x000000000001293 <+79>: mov %eax,0x2d9f(%rip) # 0x4038 <in_cs>--Type <RET> for more, q to quit, c to continue without pagin
q--c
0x000000000001299 <+85>: mov 0x2d99(%rip),%eax # 0x4038 <in_cs>
0x00000000000129f <+91>: cmp $0x2,%eax
0x0000000000012a2 <+94>: jne 0x12c4 <+0+128>
0x0000000000012a4 <+96>: mov 0x2d75(%rip),%rax # 0x4020 <stderr@@GLIBC 2.2.5>
0x0000000000012ab <+103>: mov %rax,%rcx
0x0000000000012ae <+106>: mov $0x1,%edx
0x0000000000012b3 <+111>: mov $0x1,%esi
0x0000000000012b4 <+116>: lea 0x0(%rip),%rdi # 0x2070
0x0000000000012b7 <+123>: callq 0x10a0 <fwrite@plt>
0x0000000000012c4 <+128>: mov 0x2d75(%rip),%eax # 0x4040 <p0_in_cs>
0x0000000000012ca <+134>: add $0x1,%eax
0x0000000000012cd <+137>: mov %eax,0x2d6d(%rip) # 0x4040 <p0_in_cs>
```

- peterson\_correct-O3



```
Reading symbols from peterson_correct-O3...
(gdb) disass p0
Dump of assembler code for function p0:
0x0000000000012f0 <+0>: lea 0xd88(%rip),%rdi # 0x207f
0x0000000000012f7 <+7>: sub $0x8,%rsp
0x0000000000012fb <+11>: callq 0x1040 <puts@plt>
0x000000000001300 <+16>: movl $0x1,0x2d36(%rip) # 0x4040 <flag>
0x00000000000130a <+26>: mfence
0x00000000000130d <+29>: mfence
0x000000000001310 <+32>: movl $0x1,0x2d2e(%rip) # 0x4048 <turn>
0x00000000000131a <+42>: mfence
0x00000000000131d <+45>: jmp 0x132b <+0+59>
0x00000000000131f <+47>: nop
0x000000000001320 <+48>: mov 0x2d22(%rip),%eax # 0x4048 <turn>
0x000000000001326 <+54>: cmp $0x1,%eax
0x000000000001329 <+57>: jne 0x1335 <+0+69>
0x00000000000132b <+59>: mov 0x2d13(%rip),%eax # 0x4044 <flag+4>
0x000000000001331 <+65>: test %eax,%eax
0x000000000001333 <+67>: jne 0x1320 <+0+48>
0x000000000001335 <+69>: callq 0x10a0 <sched_getcpu@plt>
0x00000000000133a <+74>: mov %eax,0x2d14(%rip) # 0x4054 <cpu p0>
0x000000000001340 <+80>: mov 0x2cf2(%rip),%eax # 0x4038 <in_cs>
0x000000000001346 <+86>: add $0x1,%eax
0x000000000001349 <+89>: cmp $0x2,%eax
0x00000000000134c <+92>: mov %eax,0x2ce6(%rip) # 0x4038 <in_cs>
0x000000000001352 <+98>: je 0x1378 <+0+136>
0x000000000001354 <+100>: addl $0x1,0x2cd5(%rip) # 0x4030 <p0_in_cs>
0x00000000000135b <+107>: sub $0x1,%eax
0x00000000000135e <+110>: mov %eax,0x2cd4(%rip) # 0x4038 <in_cs>
0x000000000001364 <+116>: movl $0x0,0x2cd2(%rip) # 0x4040 <flag>
0x00000000000136a <+126>: mfence
0x000000000001371 <+129>: jmp 0x1300 <+0+16>
0x000000000001373 <+131>: nopl 0x0(%rax,%rax,1)
0x00000000000137b <+136>: mov 0x2ca1(%rip),%rcx # 0x4020 <stderr@@GLIBC 2.2.5>
```

反組譯兩程式 p0 後可以看出 peterson\_correct-O3 會使用 mfence 確保程式執行的順序和我們所希望的一樣，比免編譯器打亂執行順序。peterson\_correct-O3 是以 atomic operation 來操作，確保變數在操作時不會被其他 thread 同時存取; 但 peterson\_trival-g 因為沒有以 atomic operation 操作，導致讀取 turn 時 p0 和 p1 同時出現在 critical section。因此 peterson\_correct-O3 的執行速度較 peterson\_trival-g 慢。

