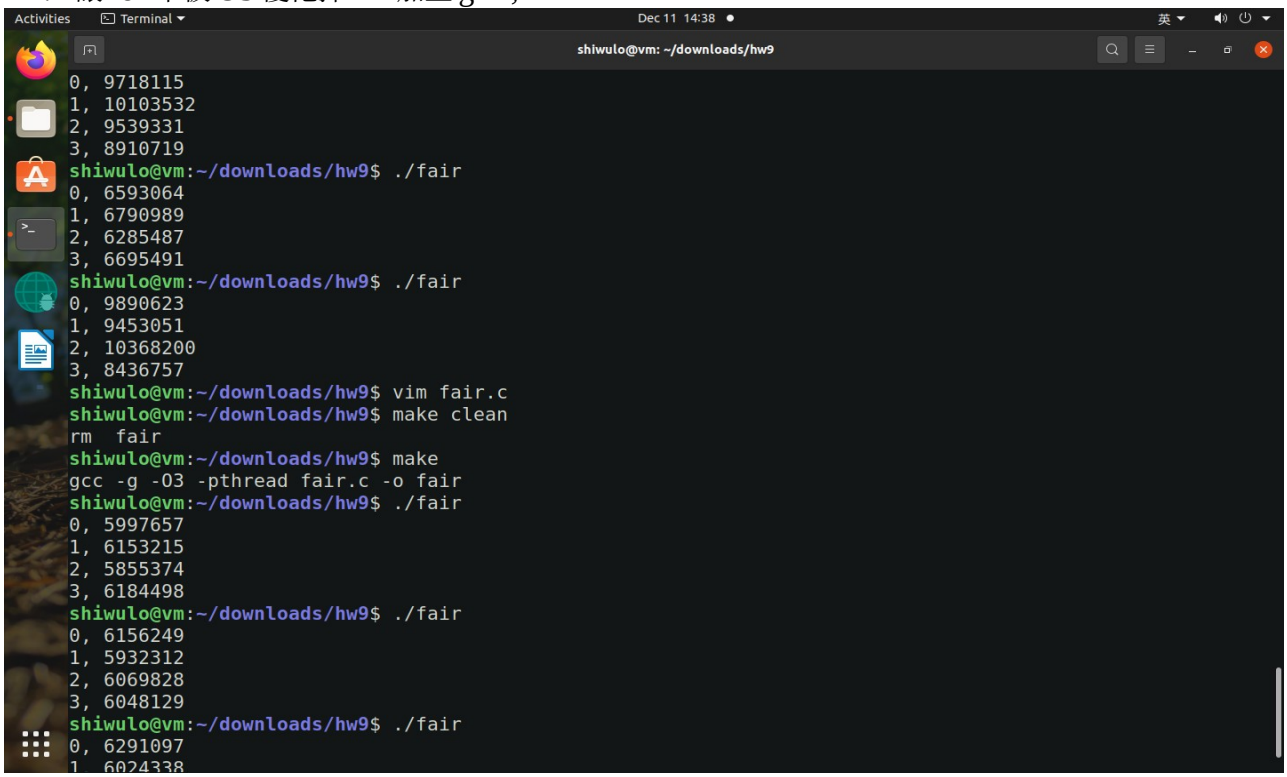
408410007 鄭 O 云

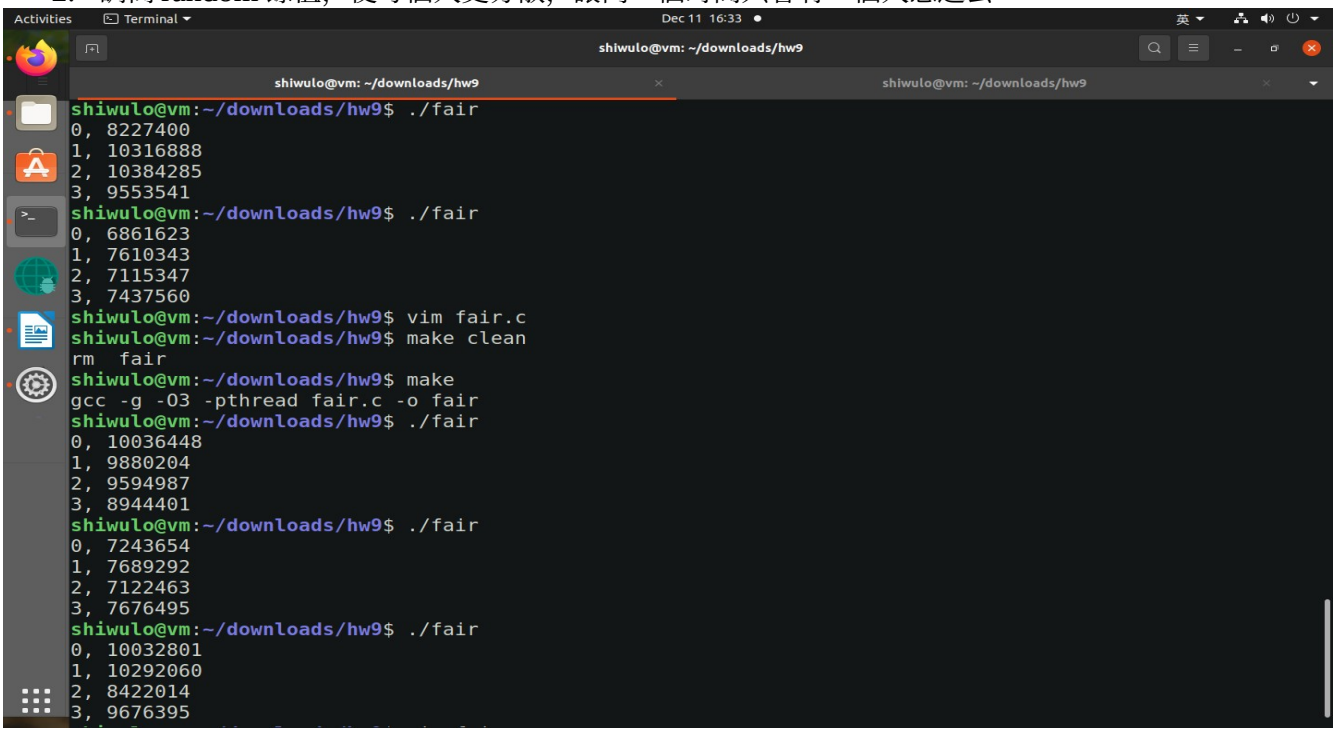
老師總共試了 3 種方法，我也試試看，以下是我的結果：

1. 讓 for 不被 O3 優化掉 => 加上 g++;



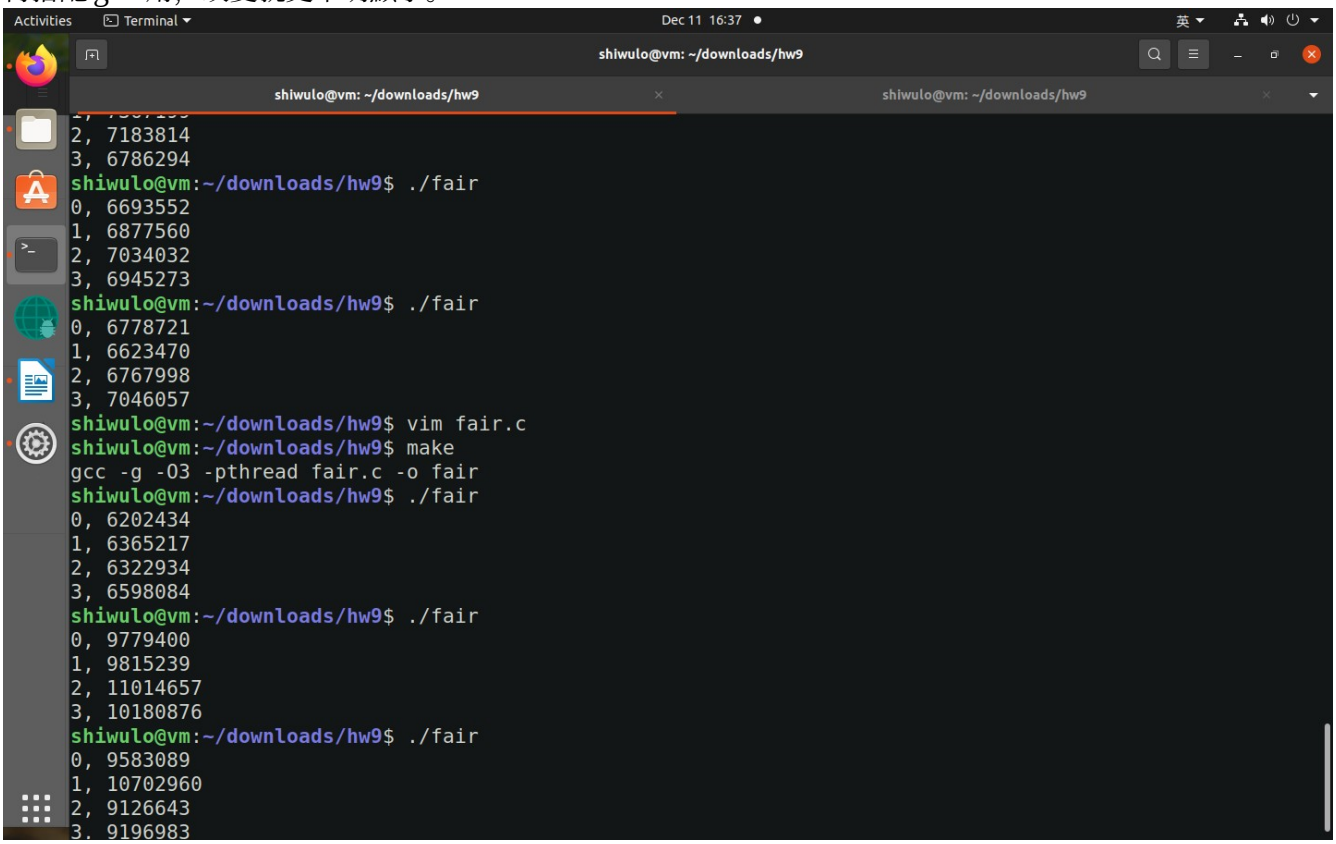
實做：上面是 for 會被優化掉的結果，有差到 16 的；下面修改過 fair.c，在 for 裡加入 g++之後的結果，就相差的比較小了。

2. 調高 random 餘值，使每個人更分散，讓同一個時間只會有一個人想進去

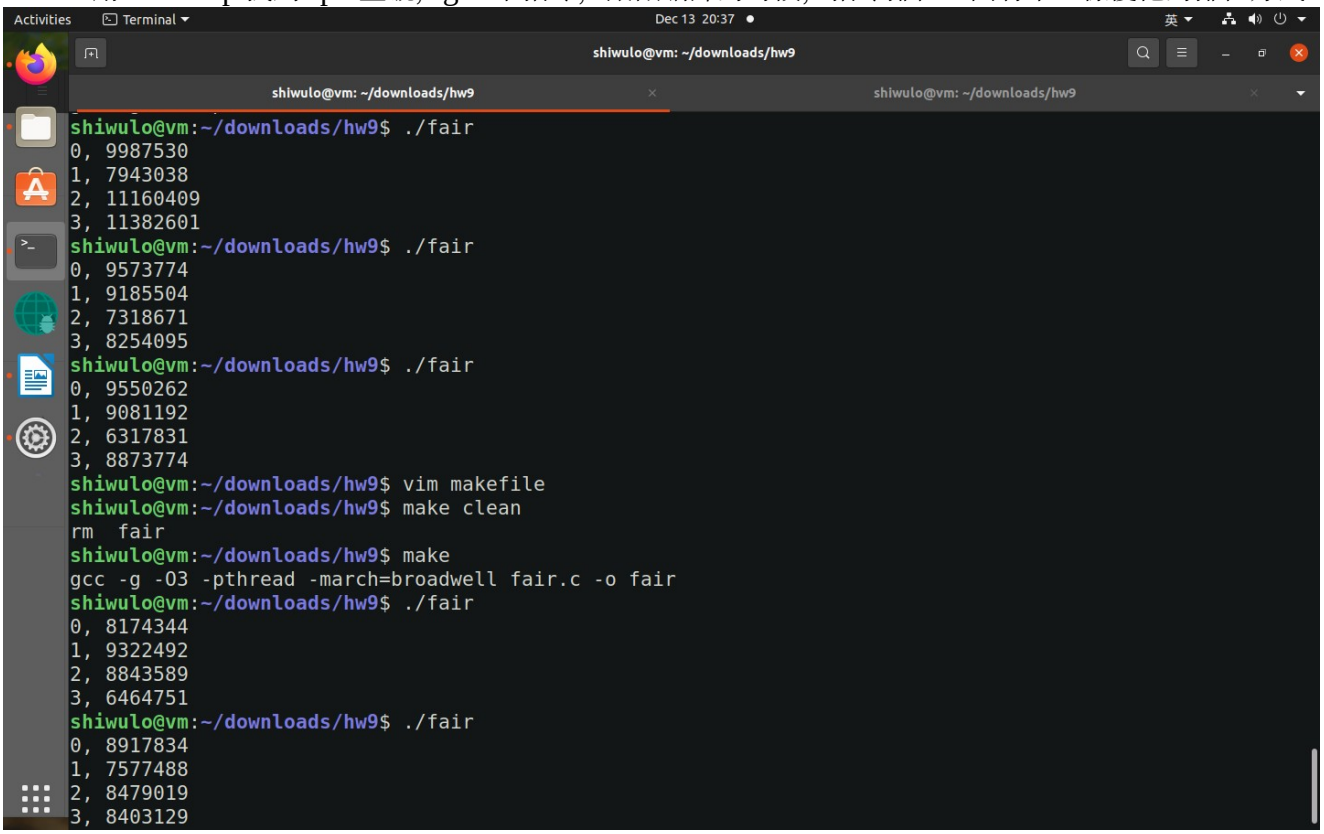


實做：其實我覺得只有些微的改變

再搭配 g++用，改變就更不明顯了。



3. 用 wikichip 找到 cpu 型號，gcc 下指令，讓做編譯的時候，指令排程上面有不一樣優化的排程方式



實做：我的 cpu 是 i5-5250U，在 wikichip 上找不到，在維基百科有找到這個是數於 Broadwell-U 微架構，以下是連結：https://zh.wikipedia.org/wiki/Intel_Core_i5%E8%99%95%E7%90%86%E5%99%A8%E5%88%97%E8%A1%A8

```
shiwulo@vm:~/downloads/hw9$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 61
model name      : Intel(R) Core(TM) i5-5250U CPU @ 1.60GHz
stepping        : 4
cpu MHz         : 1600.000
cache size      : 3072 KB
```

所以下在 makefile 下指令加上-march=broadwell，做編譯器優化支援，結果有比較公平。