HW4_109403021

編譯與執行:

```
resource(s) deleted
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ gcc -o p1 p1.c sem.c
p1.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
```

```
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ gcc -o p2 p2.c sem.c
p2.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
```

```
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ gcc -o p3 p3.c sem.c
p3.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
```

```
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ ./p1 &
[1] 19182
P1111111111 is here
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ ./p3 &
[2] 19215
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ ./p2 &
[3] 19236
P222222222 is here
P3333333 is here
P3333333 is here
P1111111111 is here
P222222222 is here
P3333333 is here
P3333333 is here
P1111111111 is here
P222222222 is here
P3333333 is here
P3333333 is here
P1111111111 is here
P222222222 is here
P3333333 is here
P3333333 is here
P1111111111 is here
P222222222 is here
P3333333 is here
P3333333 is here
```

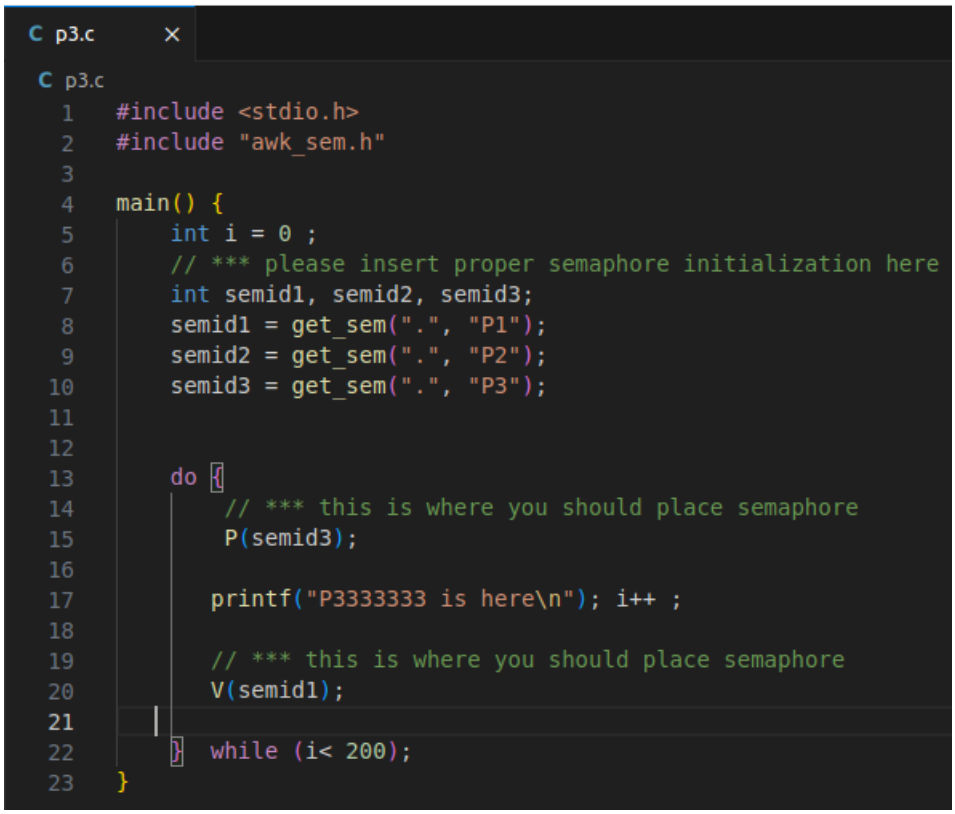
```
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ ./p1 &
[1] 19611
P1111111111 is here
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ ./p2 &
[2] 19632
P222222222 is here
harrison@harrison-VirtualBox:~/Desktop/linux semaphores/proj3$ ./p3 &
[3] 19682
P3333333 is here
P3333333 is here
P1111111111 is here
P222222222 is here
P3333333 is here
P3333333 is here
P1111111111 is here
P222222222 is here
P3333333 is here
P3333333 is here
```

解釋：

已假設 p1 永遠第一個執行，故直接在 p1 建立 semaphore，並把「.P1」初始值設為 1，這樣等等第一次遇到 P(semid1)不會被 block，只會扣 1。其餘則把初始值設為 0

```c
#include <stdio.h>
#include "awk_sem.h"

main() {
    int i = 0 ;
    // *** Please insert proper semaphore initialization here
    int semid1, semid2, semid3;
    semid1 = create_sem(".", "P1", 1);
    semid2 = create_sem(".", "P2", 0);
    semid3 = create_sem(".", "P3", 0);

    do {
        // *** this is where you should place semaphore
        P(semid1);

        printf("P1111111111 is here\n"); i++;

        // *** this is where you should place semaphore
        V(semid2);
        P(semid1);

    } while (i < 100) ;
}
```

```c
#include <stdio.h>
#include "awk_sem.h"

main() {
    int i = 0 ;
    // *** please insert proper semaphore initialization here
    int semid1, semid2, semid3;
    semid1 = get_sem(".", "P1");
    semid2 = get_sem(".", "P2");
    semid3 = get_sem(".", "P3");

    do {
        // *** this is where you should place semaphore
        P(semid2);

        printf("P222222222 is here\n"); i++ ;

        // *** this is where you should place semaphore
        V(semid3);
        V(semid3);

    } while (i < 100);
}
```



```c
#include <stdio.h>
#include "awk_sem.h"

main() {
    int i = 0 ;
    // *** please insert proper semaphore initialization here
    int semid1, semid2, semid3;
    semid1 = get_sem(".", "P1");
    semid2 = get_sem(".", "P2");
    semid3 = get_sem(".", "P3");


    do {
        // *** this is where you should place semaphore
        P(semid3);

       printf("P3333333 is here\n"); i++ ;

       // *** this is where you should place semaphore
       V(semid1);

    } while (i< 200);
}
```

由於 p3 一回合要跑兩圈其他只跑一圈，所以在 p2 處多叫一次 V(semid3)，這樣會讓其 semaphore 值再更多 1 使 p3 遇到兩次 P(semid3)才會被 block，但這樣同時也會造成 V(semid1)被執行兩次，所以在 p1 處多放一個 P(semid1)，這樣兩次叫醒對應到兩個 block 才可以使 p1 不會一回合跑兩次。

如此就可以達成:

1111

222222

3333

33333

11111

2222

33333

3333

…

…