

## homework 4-15 multi-thread pid &&6-30 with-mutex

### 程式說明

```
void allocate_thread()
{
    int i;
    pthread_pool = (pthread_t*)malloc(sizeof(pthread_t)*THREAD_AMOUNT);
    for(i = 0; i < THREAD_AMOUNT; i++)
        pthread_create(&pthread_pool[i], NULL, randomAllocate_pid, NULL);
}

void* randomAllocate_pid()
{
    int r;
    while(1)
    {
        r = (rand() % 2500) + 501;

        pthread_mutex_lock( &mutex1 );
        if(!pid_map[r])
        {
            printf("%d. Allocate PID = %d\n", count++, r);
            pid_map[r] = 1;
            release_pid(r);
            pthread_mutex_unlock( &mutex1 );
            break;
        }
    }
    pthread_exit(NULL);
}
```

`allocate_thread`是初始化 那些thread, 讓他們能夠呼叫`randomAllocate_pid`

`randomAllocate_pid`是自動配置pid並且用`mutex_lock`確保不會同時存取的問題, 然後配置完後會馬上呼叫`release_pid`, 並且最後在`mutex_unlock`。  
總共30個。

## 結果展示

```
user@instant-contiki:~/Desktop/os/hw2$ ./pid.out  
Start process PID allocate
```

```
0. Allocate PID = 685  
PID 685 released!  
1. Allocate PID = 2628  
PID 2628 released!  
2. Allocate PID = 2886  
PID 2886 released!  
3. Allocate PID = 1891  
PID 1891 released!  
4. Allocate PID = 576  
PID 576 released!  
5. Allocate PID = 1241  
PID 1241 released!  
6. Allocate PID = 976  
PID 976 released!  
7. Allocate PID = 1617  
PID 1617 released!  
8. Allocate PID = 2950  
PID 2950 released!  
9. Allocate PID = 1345  
PID 1345 released!  
10. Allocate PID = 1334  
PID 1334 released!  
11. Allocate PID = 2779  
PID 2779 released!  
12. Allocate PID = 2371  
PID 2371 released!  
13. Allocate PID = 738  
PID 738 released!  
14. Allocate PID = 1476  
PID 1476 released!  
15. Allocate PID = 2990  
PID 2990 released!
```

```
16. Allocate PID = 1469  
PID 1469 released!  
17. Allocate PID = 2610  
PID 2610 released!  
18. Allocate PID = 1958  
PID 1958 released!  
19. Allocate PID = 925  
PID 925 released!  
20. Allocate PID = 949  
PID 949 released!  
21. Allocate PID = 2736  
PID 2736 released!  
22. Allocate PID = 1990  
PID 1990 released!  
23. Allocate PID = 2998  
PID 2998 released!  
24. Allocate PID = 885  
PID 885 released!  
25. Allocate PID = 976  
PID 976 released!  
26. Allocate PID = 2855  
PID 2855 released!  
27. Allocate PID = 2770  
PID 2770 released!  
28. Allocate PID = 2453  
PID 2453 released!  
29. Allocate PID = 2741  
PID 2741 released!
```

```
End Process PID allocate
```

## homework 4-19 prime number with multi-thread

### 程式說明

```
void* prime(void* data)
{
    printf("Start Threading Prime\n");

    int *n = (int*) data;
    int num = (*n), prime;
    int i, j;

    printf("following numbers are prime numbers\n");
    num > 2 ? printf("2\n") : pthread_exit(NULL);
    for(i = 3; i < num; i+=2)
    {
        prime = 1;
        for(j = 2; j*j<=i; j++)
        {
            if(i % j == 0)
            {
                prime = 0;
                break;
            }
        }
        if(prime) printf("%d\n", i);
    }
    printf("End Threading Prime\n");
    pthread_exit(NULL);
}
```

計算 n 以下的質數有哪些(不包含n)。

```
user@instant-contiki:~/Desktop/os/hw2$ ./prime.out
Start Process
input number: 89
Start Threading Prime
following numbers are prime numbers
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
End Threading Prime
End Process
```