

System programming homework #8

王興彥 408410113

1. man capabilities 權限

(1) CAP_SETPCAP(Thread capability sets)

- A. Permitted: 這個集合定義了 process 所能夠擁有的特權的上限。如果某個 capability 不在 Permitted 集合中，那麼該 process 便不能進行這個 capability 所對應的特權操作。Permitted 集合是 Inheritable 和 Effective 集合的 superset。
- B. Inheritable: 執行 exec() 系運行其他命令時，能夠被新命令繼承的 capabilities，被包含在 Inheritable 集合中。
- C. Effective: 內核檢查該 process 是否可以進行特權操作時，檢查的對象便是 Effective 集合。Permitted 集合定義了上限。process 可以刪除 Effective 集合中的某 capability，隨後在需要時，再從 Permitted 集合中恢復該 capability，以此達到臨時禁用 capability 的功能。

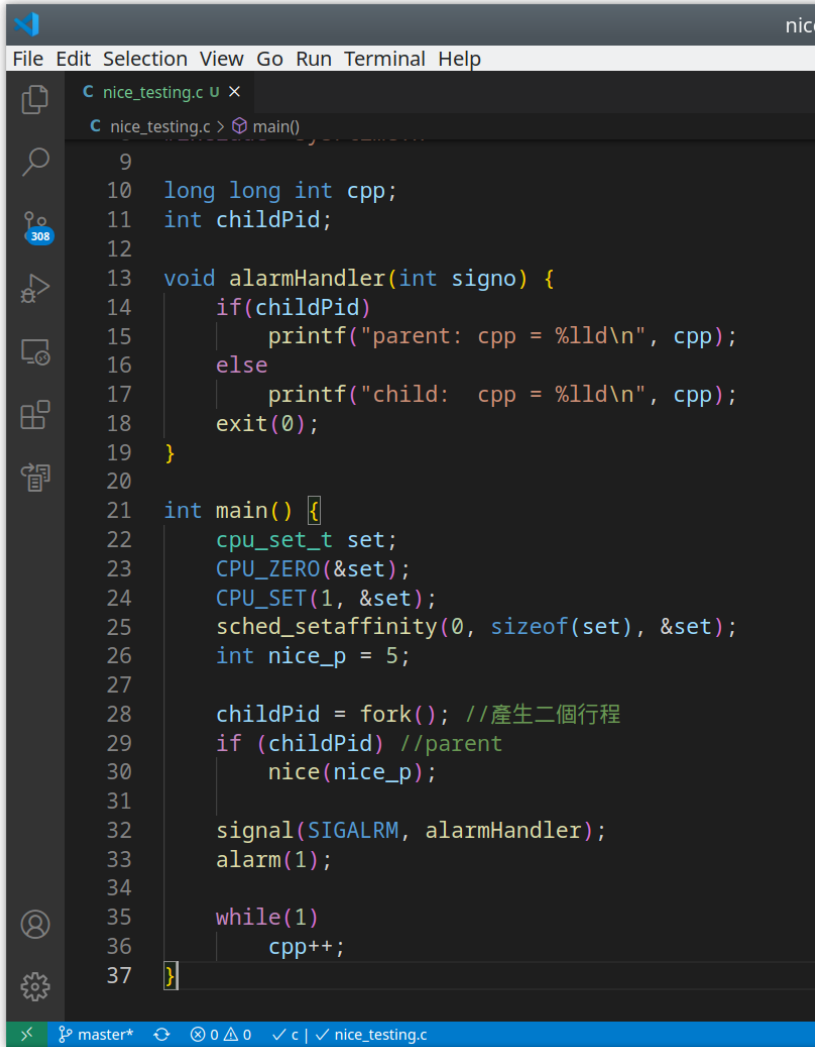
(2) CAP_SYS_NICE

允許提升優先等級及設置其他 process 的優先等級

(3) CAP_SYS_BOOT

允許重新啟動系統

2. 想辦法量測『優先權高一等級的 task 比正常優先權的 task 速度快多少』？



```
9
10 long long int cpp;
11 int childPid;
12
13 void alarmHandler(int signo) {
14     if(childPid)
15         printf("parent: cpp = %lld\n", cpp);
16     else
17         printf("child:  cpp = %lld\n", cpp);
18     exit(0);
19 }
20
21 int main() {
22     cpu_set_t set;
23     CPU_ZERO(&set);
24     CPU_SET(1, &set);
25     sched_setaffinity(0, sizeof(set), &set);
26     int nice_p = 5;
27
28     childPid = fork(); //產生二個行程
29     if (childPid) //parent
30         nice(nice_p);
31
32     signal(SIGALRM, alarmHandler);
33     alarm(1);
34
35     while(1)
36         cpp++;
37 }
```

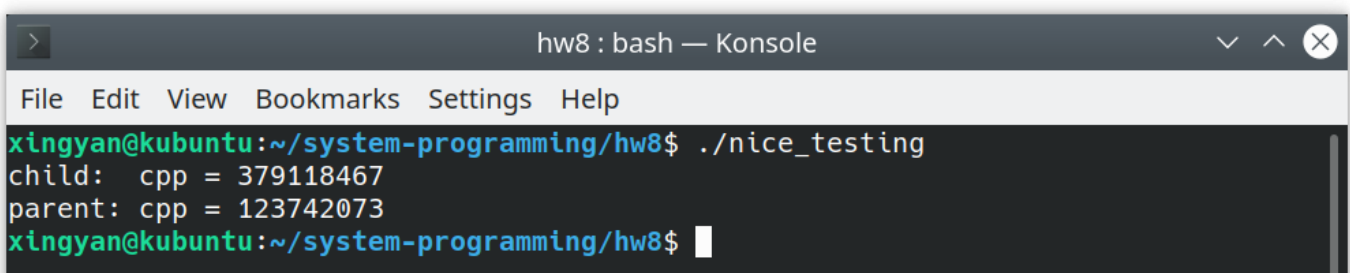
a. 先將程式設執行定到同一顆 CPU 執行，才能更明顯看出優先等級造成的差異。

b. 將 parent 行程的優先等級調低 5。

c. 設定 alarm 並讓 alarm 結束時因出 cpp 的值。

d. 期間不斷增加 cpp 的值。

e. $379118467 / 123742037 \approx 3.06 \approx 3.05 = (1.25)^5$ 符合調整的等級。



```
hw8 : bash — Konsole
File Edit View Bookmarks Settings Help
xingyan@kubuntu:~/system-programming/hw8$ ./nice_testing
child:  cpp = 379118467
parent: cpp = 123742073
xingyan@kubuntu:~/system-programming/hw8$
```