Report of Project 1

B06902022陳翰霆

1.設計

這次的 project，我使用了兩個 cpu，一個用來跑 main process，另一個用來跑 main process 的 child，目的是兩種 process 跑的時候就不會互相干擾，比較好實作。

我在 main 裡面，會根據 input，看什麼時候有 process ready，就把該 process 加到 pool 裡面，每當我們要跑 process 的時候，就從 pool 裡找一個合適的 process 出來執行，根據不同的 policy 就做出相應的動作。

若我們需要中斷某個 process，因為所有的 child process 都在同一個 CPU 上，所以可以直接用 sched_setscheduler，將他 set 成 SCHED_IDLE；如果要讓他恢復執行，就 set 成 SCHED_OTHER。

我使用了 clock_gettime 加上 CLOCK_REALTIME 參數，來取得現在的時間。每當有程式執行完畢，就會使用我自己寫的 system call，把資訊寫進 dmesg。此時 main process 就會重新從 pool 裡找使否有 child process 要執行，並找一個出來恢復執行。

2. 核心版本

使用 Linux 4.14.25，與助教投影片介紹的版本相同。

3. 比較實際結果與理論結果，並解釋造成差異的原因

比較

順序都是正確的，不過跟理論值比較起來，在 main process 會多需要一些時間，大概每一千個 unit time 的理論值，main process 會多用十幾到幾十個 unit time，尤其是 PSJF。

差異原因

因為我們的 main process 還需要判斷很多條件，在 FIFO 和 SJF 比較簡單，可以不用牽涉到 context switch，但在 RR 和 PSJF 的排程時，需要有比較多的 overhead 在選擇一個 process 來跑、和升降它們的 priority 的時間，所以跟 child process 比較起來會花很多額外的時間。無法做到跟 child 一樣，完全只跑 unit time。

當我們使用 sched_setscheduler 的時候，是請 kernel 幫我們把 child process 的 priority 調整為 idle，中間會有一點時間差，並且 idle 是以極低的 priority 來跑，可能有時候還是被 CPU 排程到一下，有偷跑的可能。

而且在 main process裡，fork 出 child process 和升降child process 的 priority 都是 system call，本身就會花比較多的時間，所以有這樣的差異，我覺得應該是合理的。