

B06902127 資工三 鄭人愷

1. 設計

本次 project 使用到兩個Core，其一執行Scheduling Algorithm，而另一個執行fork()出來的child process。Scheduling Algorithm會在每次run_unit_time開始前挑選下個unit_time該執行的process，在挑選結束之後會用sched_setscheduler重新設定process的priority，在前一個unit_time執行的process的priority會被下修，而下個unit_time該執行的process的priority會被上調。

在FIFO Scheduling中，會依照最早可以開始執行的順序作為排列。在SJF Scheduling中，會依照最短所需執行時間作為排列。在PSJF Scheduling中，和SJF Scheduling相同，會依照最短所需執行時間作為排列，而和SJF Scheduling不同之處在於，SJF Scheduling開始執行一個process後不會preempt其他process，也就是process需完整執行完畢才會開始執行下個執行時間最短的process，但PSJF Scheduling可以在每個unit_time之間下修目前還在執行的process的priority，並上修其他執行時間最短的process的priority，令所需執行時間較長的process之後再執行。Round Robin Scheduling會在每個time quantum之後執行下一個在ready queue中的process。

Scheduling 的簡易內容如下

Loop :

- 1) 如有process arrive，則加入ready queue
- 2) 依據Scheduling Policy找出需要優先執行的process
- 3) set_scheduler修正priority
- 4) run unit_time
- 5) Update timer

2. 核心版本

Linux-4.15.0

3. 比較實際結果與理論結果

在理論上，所需執行時間相同的process在沒有被preempt的情況下執行時間（占用CPU的時間）應該也會相同，然而卻有幾微秒到幾百微秒不等的差距。

推測主要原因有二

- 1) CPU狀態

CPU允許在特定情況下進行加速，如intel的turbo boost就可能，導致運算效能上的差距，用gettimeofday之類的方式無法正確預測。也有可能CPU同時在處理許多real-time process時導致process被preempt。

2) Virtual Machine上執行

由於有些VM的resource是動態配置，可能在執行過程將resource動態配置導致運算效能的減少或增加，以致process執行速度的差異。