

2020 OS Project 1: Process Scheduling

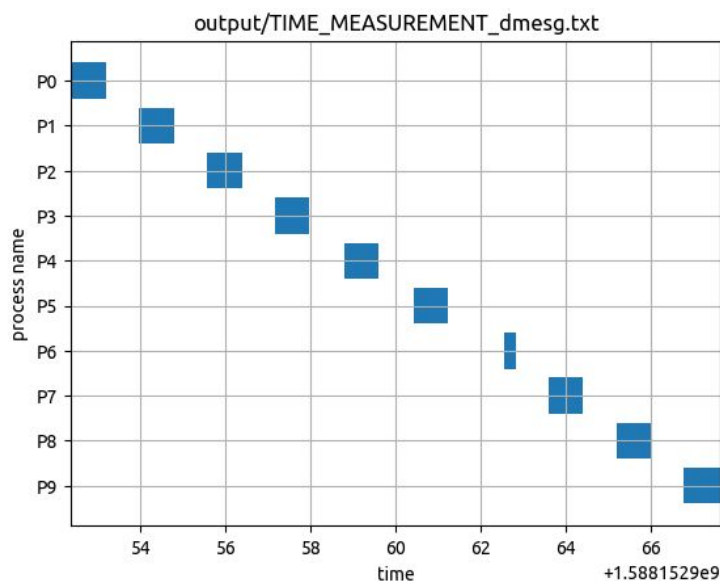
1. 設計

主程式是負責排程的，他會紀錄目前在第幾個時間的回合，一個時間回合的長度就是空迴圈跑一百萬次，同時也會紀錄子程序到達的時間，以及剩餘需要執行的時間。在主程式的部份可以分成三個步驟，第一步是等待應該要完成的子程式完成並紀錄，也就是剩餘時間剛好剩 1 的子程序，如果沒有的話就直接到下一步。第二步是檢查哪些程式到達了，到達的就會被 fork 起來並等待，在 round robin 的機制下，他們還會進到 queue 中排隊。第三步就是決定接下來這個回合適輪到誰執行，根據不同的 policy 會選擇到不同的結果。像是 round robin 就會確保要過 500 個單位才能換下一個程式（如果沒有提早結束的話）。這三步驟做完後就會等一個時間單位，並紀錄執行中的程式的剩餘時間，然後進下一回合。程式中的 running index 代表此回合在執行的程式，-1 代表目前沒有子程式執行中，所有程式都會用 array 存著資訊，coming index 就是指著已經到達的程式的尾端，有新到達的就會增加。time_cnt 紀錄從開始排程的時間回合。finished 紀錄已完成的數量，若全部都已完成就會結束。

2. 核心版本

linux kernel 使用 4.14.25

3. 比較實際結果與理論結果，並解釋造成差異的原因



這個是以其中一筆測試為例，輸入的資料所需的執行時間是相同的，可是實際紀錄到的卻有長有短，可能是因為有許多其他的程式也都在執行，如果原本其他在執行的程式空閒下來，就會導致這邊執行所需時間變短，或者是使用者突然開啟新的程式，佔用 CPU 資源，就會導致所需時間變長。