

OS Project1 report

資工二 b07902033 陳郁鳳

一、設計

使每個 process 在到達 ready_time 時就先啟動，用 fork 獲得該 process 的 pid。由於我們使用的是單核心，cpu 在面臨 fork 時會先執行 parent process，也就是其他 priority 更高的 process，一直到最後 parent process 執行結束後，再用 waitpid 令 child process 結束，如此一來，child process 將不會影響 parent process 的進行。

將程式分成三個部分 main, run, process，main 負責基本輸入和一些前置作業（如將 process 依據到達時間排序），而後 call run 程式進行運算。

run 程式根據不同 policy 決定分配 process 的方法，為決定 cpu 該執行那個 process 的部分。

process 程式則進行較細微的 process 操作（如 fork 和實際設定 process 的 priority）。

輸出部分則直接用 sudo 寫進 dmesg 儲存位置，因此沒有使用到 syscall。

二、核心版本

Linux 4.15.0-96-generic #97~16.04.1-Ubuntu

三、比較實際結果與理論結果，並解釋造成差異的原因

雖然我們已經用 unit time 迴圈，使時間計算盡量貼近 cpu 實際運作的時間，但還是會和實際運算有所不同。理論上電腦跑任何一行都需要一定時間，而這是我們用 unit time 迴圈所不能實現的。因此在這樣時間誤差下，可能就會導致實際結果和理論結果不同。

除此之外，我們使用的 policy 一次只有一種，cpu 的實際作法很可能是混合多種 policy，並且因為我實作時，對於在所選 policy 下條件一樣的 process，多半使用 for 迴圈從 index 小的開始選擇 process，這點可能跟 cpu 實際操作也不一樣。