2020 OS project1 report

袁才育 b06902115 資工三

設計

1. FIFO
    - 原理
        - 先到的process就先執行。
    - 執行方式
        - 將所有process按照ready time由小排序到大。
        - 當排在第一個process的ready time抵達時，則進行fork。
        - 當剩下的process的ready time到時，若此刻無process在執行，則進行fork，直到該process執行結束。
2. RR
    - 原理
        - 利用計時器固定週期，讓正在執行的process在執行一個週期後進入queue中，並從queue中選定下一個執行的process。
    - 執行方式
        -
3. SJF
    - 原理
        - 選擇執行時間最短的process來執行。
    - 執行方式
        - 將所有process照ready time由小到大排序，若ready time相同，再按照execution time由小到大排序。
        - 當排在第一個process的ready time抵達時，則進行fork。
        - 當剩下的process的ready time到時，對剩餘的process照ready time由小到大排序，若ready time相同，再按照execution time由小到大排序。若此刻無process在執行，則進行fork，直到該process執行結束。
4. PSJF
    - 原理
        - 每次選擇剩餘執行時間最短的 process 來執行，當新產生的 process 執行時間更短時，就可以插隊。
    - 執行方式
        -

核心版本

linux_4.14.25

比較

經過分析與討論，我認為理論與實際的差異主要來自於以下三點：

1. scheduler 除了排程外，還有其他執行所花費的時間，例如動態調整各process的priority、執行新的process與資料結構的操作等。
2. 同樣執行 1000000 次迴圈，但 CPU 所花費的時間不完全一樣，除了各process執行迴圈的單位時間不相同外，scheduler的計時器也不會完全和process的同步，進而造成排程上的誤差。
3. 因為電腦上不只有我跑的project1的程序，所以有可能我的程序因為context switch而被中斷，然而計時不會跟著扣掉，因此造成程序執行時間增加。