

1. 設計

My_code:回傳 child 開始與結束時間

MY_code-2:把 pid 和兩時間 print 在 knernel

Scheduler(b07902071.c):把 CPU 設置在第 0 顆，用 sched_FIFO 把 priority 設成 50，然後讀 input 將其依找 ready time 排序，再以 ready_num 表示進到 ready queue 的 task 數量。

(A) FIFO

若 ready queue 沒有 process 在等，scheduler 跑距離下一個 process 開始之時間長度，然後更新時間再 fork 新 process

若 ready queue 有 process 再等，依照 FIFO 原則，先進先出，除非出現新 process 需要被 create，不然跑到結束。

所有 process 跑完跳出迴圈

(B) RR

若 ready queue 沒有 process 等待，同上 FIFO 作法

若有 process 在等待，讓每個 process 跑 500 clock time 之後就換下一隻 process

(C) SJF

若 ready queue 沒有 process 等待，同上 FIFO 作法

若有 process 在等待，挑選 exec time 最小者執行，結束後再挑下一個最短的

(D) PSJF

若 ready queue 沒有 process 等待，同上 FIFO 作法

若有 process 在等待，與 SJF 差別只有當新 process 進入 ready queue 之後就要重新挑 exectime 最小的執行

2. 核心版本

Linux 4.12.25

3. 比較實際與理論結果

順序與理論值一樣

執行時間則會受到 scheduler 造成的 overhead 影響