R08922119 蕭景忠

NTU OS 2020 Spring Project 1

CPU Scheduling

一、設計

1. System call

   新增systemcall set_time: 利用 SYSCALL_DEFINE3(set_time,int,isStart,int,pid,timespec*,timeStamp) 在每次fork()時呼叫syscall(350, 1, pid, timeStamp)記錄開始時間，並在process結束時呼叫syscall(350, 0, pid, timeStamp)記錄結束時間。

2. 優化

   若child process與parent process共用同一CPU，會互相搶奪資源而影響時間計算還有執行效率，所以利用sched_setaffinity()將他們分配給不同CPU，以精確計算時間。

3. 演算法

   a. FIFO

      process依ready time順序(由小到大，相同則按照讀入順序)逐一丟入Ready Queu。同時，只要Queue非空，則進行以下步驟：從Ready Queue中取出第一個process執行，執行結束後再從Ready Queue中取中下一個，重複上述步驟一直到Ready Queue空了為止。

   b. RR

      基本與FIFO相同，但是每個process執行最多500 time units，如果達到上限則再丟入Ready Queue中，並從Queue中取出下一個process來執行。

   c. SJF

      將process按照ready time丟入ready queue中，當ready queue被插入，依照queue中每個process的execution time將process排序(ready time決定誰先進queue，而queue內是依execution time 排序)。同時，只要Queue非空，則進行以下步驟：從Ready Queue中取出第一個process執行，執行結束後再從Ready Queue中取中下一個，重複上述步驟一直到Ready Queue空了為止。

   d. PSFJ

基本上與SJF相同，差別是當ready queue被插入時，會與執行中的process一同比較execution time，若執行中的process剩餘execution time較高，則被丟入ready queue，先執行被插入的process。

二、核心版本

作業系統：Ubuntu 16.04

kernel: linux kernel 4.14.25

三、比較

單位：time units

1. FIFO

	預估	實際
FIFO_1	2500	2555
FIFO_2	87000	86304
FIFO_3	23000	22395
FIFO_4	3200	3160
FIFO_5	23000	22293

2. RR

	預估	實際
RR_1	2500	2444
RR_2	9000	8716
RR_3	30000	29164
RR_4	23000	22325
RR_5	23000	22317

3. SJF

	預估	實際
SJF_1	14000	13731
SJF_2	15300	15019
SJF_3	32020	31603

SJF_4	11000	10810
SJF_5	3500	3450

4. PSJF

	預估	實際
PSJF_1	25000	24703
PSJF_2	11000	10849
PSJF_3	3500	3466
PSJF_4	14000	13781
PSJF_5	15300	15171

大部分的實際情況都比理論的預估值還小，可能是因為time measure的誤差值較大，十次測量結果中最大與最小差了9%，最後取十次的平均作為time unit的標準。