OS_Project1 report

B06902113 柯柏丞 資工三

1. 設計

基本函式（basic function）：

- **SET_CPU**：將指定的 process 設置到指定的 CPU 上。

  CPU_0 負責跑主程式（parent process），
  CPU_1 負責跑子程式（child process）。

- **SET_PRIORITY**：設置指定 process 的 priority。

  PRIORITY_1 為最高權重，process 處於 run 時擁有；
  PRIORITY_2 為次高權重，process 處於 ready 且為即將 run 時擁有；
  PRIORITY_3 為最低權重，process 處於 ready 時擁有。

- **unit_of_time**：跑 UNIT_TIME 的迴圈。

  UNIT_TIME 本次設置為 1000000UL。

- **create_process**：fork 一個 child process。
- **run_process**：執行 child process 並且呼叫 system call 紀錄資訊。

排程函式（scheduler function）：

- **FIFO**

  a. 主架構：

  將所有 process 依照 ready time 做排序（存到 process_index），
  接著按照 process_index 的順序逐一創建屬於他們的 child process，
  在每個單位時間和每次創建 process 時改變權重（見 b.），
  並且持續接收訊號（見 c.）去確認執行完的 process，
  一旦確認到所有 process 皆執行完畢就結束主程式。

  b. 改變權重：

  如果沒有 process 正在 run 而且還有 process 屬於 ready 狀態，
  就 run 被 ptr_run 指定到的 process（將其 priority 設為最高）。
  如果仍有 process 正在執行而且還有 process 屬於 ready 狀態，
  就提升 ptr_run 之後一個 process 的權重（將其 priority 設為次高），
  當正在 run 的 process 結束便會接著執行此 process。

  c. 接收訊號：

  一旦有 child process 結束就會接受到 SIGCHID 的訊號，
  增加 count 並且將 ptr_run 指到下一個 process。

- **RR**

  a. 主架構：

  將所有 process 先依照 ready time 做排序（存到 process_index），

接著按照 process_index 的順序逐一創建屬於他們的 child process，
在每個單位時間和每次創建 process 時改變權重（見 b.），
並且將其 process 放到 queue_index（先進先出）中，
每次 timer 計數到達 TIME_QUANTUM 且還有 ready process，
則打斷正在 run 的 process（hold = 1 時除外），
降低其 process 權重（將其 priority 設為最低），更新其 exc_time，
並且將其 process 放回 queue_index 等待接著執行。
接著持續接收訊號（見 c.）去確認執行完的 process，
一旦確認到所有 process 皆執行完畢就結束主程式。

b. 改變權重：

如果沒有 process 正在 run 而且還有 process 屬於 ready 狀態，
就 run 在 queue_index 中第一個 process（將其 priority 設為最高），
並且把 timer 歸零，且如果執行時間小於 TIME_QUANTUM，
使其 process 保證 run 完不被打斷（hold = 1）。
如果仍有 process 正在執行而且還有 process 屬於 ready 狀態，
就提升 queue_index 中第二個 process 的權重（將其 priority 設為次高），
當正在 run 的 process 結束便會接著執行此 process。

c. 接收訊號：

一旦有 child process 結束就會接受到 SIGCHID 的訊號，
增加 count 並且將 queue_index 中第一個 process 移除掉，
並且把 timer 歸零，接著解除掉 hold 的狀態（hold = 0）。

- **SJF**

a. 主架構：

將所有 process 先依照 exc time 做排序，
再將所有 process 依照 ready time 做排序（存到 process_index），
接著按照 process_index 的順序逐一創建屬於他們的 child process，
在每個單位時間和每次創建 process 時改變權重（見 b.），
創建新 process 前如果有 process 正在 run，
則降低下一個即將 run 的 process 之權重（將其 priority 設為最低）。
創建新 process 後將其放到 sjf_index（exc time 越短的越前面）中，
改變權重確保當前有最短 exc time 的 ready process 擁有次高權重，
並且持續接收訊號（見 c.）去確認執行完的 process，
一旦確認到所有 process 被執行完畢就結束主程式。

b. 改變權重：

如果沒有 process 正在 run 而且還有 process 屬於 ready 狀態，
就 run 在 sjf_index 中第一個 process（將其 priority 設為最高），
並且把其 process 從 sjf_index 中移除。

如果仍有 process 正在執行而且還有 process 屬於 ready 狀態，
就提升 sjf_index 中第一個 process 權重（將其 priority 設為次高），
當正在 run 的 process 結束便會接著執行此 process。

c. 接收訊號：

一旦有 child process 結束就會接受到 SIGCHID 的訊號，增加 count。

- **PSJF**

a. 主架構：

將所有 process 先依照 exc time 做排序，
再將所有 process 依照 ready time 做排序（存到 process_index），
接著按照 process_index 的順序逐一創建屬於他們的 child process，
在每個單位時間和每次創建 process 時改變權重（見 b.），
創建新 process 前如果有 process 正在 run，
則降低下一個即將 run 的 process 之權重（將其 priority 設為最低），
創建新 process 後將其放到 sjf_index（exc time 越短的越前面）中，
改變權重確保當前有最短 exc time 的 process 擁有次高權重，
一旦原本正在 run 的 process 不再是擁有最短 exc time 的 process，
便打斷其 process（將其 priority 設為最低）。
並且持續接收訊號（見 c.）去確認執行完的 process，
一旦確認到所有 process 被執行完畢就結束主程式。

b. 改變權重：

如果沒有 process 正在 run 而且還有 process 屬於 ready 狀態，
就 run 在 sjf_index 中第一個 process（將其 priority 設為最高），
如果仍有 process 正在執行而且還有 process 屬於 ready 狀態，
就提升 sjf_index 中第二個 process 權重（將其 priority 設為次高），
當正在 run 的 process 結束便會接著執行此 process。

c. 接收訊號：

一旦有 child process 結束就會接受到 SIGCHID 的訊號，增加 count，
並且把其 process 從 sjf_index 中移除。

2. 核心版本

Linux-4.14.25 (https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.14.25.tar.xz)
並參考 https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.14.25.tar.xz 設計。

3. 比較實際結果與理論結果，並解釋造成差異的原因

以下皆使用 TIME_MEASUREMENT 算出來的單位時間做換算表示，
也就是將執行時間除以每一個 unit time 所需花費的時間，

並且計算其平均錯誤值平均誤差 $= \frac{\sum_1^n |(實際執行時間)-(理論執行時間)|}{n(PROCESS數量)}$ 。

TIME_MEASUREMENT

PROCESS	實際結果		理論結果	
P0	0.0	493.69	0.0	500.0
P1	980.55	980.55	1000.0	1000.0
P2	1974.2	2476.11	2000.0	2500.0
P3	2958.7	3436.38	3000.0	3500.0
P4	3935.27	4446.54	4000.0	4500.0
P5	4911.64	5403.81	5000.0	5500.0
P6	5873.79	6367.72	6000.0	6500.0
P7	6841.07	7348.08	7000.0	7500.0
P8	7812.22	8354.41	8000.0	8500.0
P9	8812.97	9298.84	9000.0	9500.0
平均誤差 **= 12.48 (UNIT TIME)**				

FIFO_1

PROCESS	實際結果		理論結果	
P1	0.0	546.92	0.0	500.0
P2	547.02	1111.63	500.0	1000.0
P3	1111.75	1608.4	1000.0	1500.0
P4	1608.49	2151.19	1500.0	2000.0
P5	2151.3	2647.06	2000.0	2500.0
平均誤差 **= 32.36 (UNIT TIME)**				

FIFO_2

PROCESS	實際結果		理論結果	
P1	0.0	85085.78	0.0	80000.0
P2	85085.89	90413.18	80000.0	85000.0
P3	90413.27	91470.29	85000.0	86000.0
P4	91470.43	92539.74	86000.0	87000.0
平均誤差 **= 1384.85 (UNIT TIME)**				

FIFO_3

PROCESS	實際結果		理論結果	
P1	0.0	8709.09	0.0	8000.0
P2	8607.19	13958.21	8000.0	13000.0
P3	13958.31	17321.0	13000.0	17000.0

P4	17321.1	18453.32	16000.0	17000.0
P5	18453.43	19560.67	17000.0	18000.0
P6	19560.77	20766.99	18000.0	19000.0
P7	20767.2	25283.84	19000.0	23000.0
平均誤差 **= 326.16 (UNIT TIME)**				

FIFO_4

PROCESS	實際結果		理論結果	
P1	0.0	2131.73	0.0	2000.0
P2	2131.84	2632.66	2000.0	2500.0
P3	2632.77	2829.04	2500.0	2700.0
P4	2829.14	3402.96	2700.0	3200.0
平均誤差 **= 52.52 (UNIT TIME)**				

FIFO_5

PROCESS	實際結果		理論結果	
P1	0.0	8532.44	0.0	8000.0
P2	8532.54	13951.21	8000.0	13000.0
P3	13951.31	17158.51	13000.0	16000.0
P4	17158.61	18220.09	16000.0	17000.0
P5	18220.19	19273.45	17000.0	18000.0
P6	19273.57	20377.52	18000.0	19000.0
P7	20377.65	24515.89	19000.0	23000.0
平均誤差 **= 216.46 (UNIT TIME)**				

RR_1

PROCESS	實際結果		理論結果	
P1	0.0	498.88	0.0	500.0
P2	498.99	1047.27	500.0	1000.0
P3	1047.4	1584.94	1000.0	1500.0
P4	1585.04	2117.34	1500.0	2000.0
P5	2117.45	2665.47	2000.0	2500.0
平均誤差 **= 33.45 (UNIT TIME)**				

RR_2

PROCESS	實際結果		理論結果	
P1	0.0	7786.78	0.0	7500.0
P2	203.17	8915.8	500.0	9000.0

平均誤差 **= 249.70 (UNIT TIME)**

RR_3

PROCESS	實際結果		理論結果	
P3	2333.95	18702.94	3000.0	17000.0
P1	0.0	20228.33	0.0	18500.0
P2	1141.6	20775.46	1500.0	19000.0
P6	4552.28	26705.47	7000.0	27000.0
P5	3944.23	28423.92	5500.0	29000.0
P4	3545.47	29575.93	5000.0	30000.0

平均誤差 **= 1732.42 (UNIT TIME)**

RR_4

PROCESS	實際結果		理論結果	
P4	1592.45	6430.81	1500.0	5500.0
P5	2874.83	7551.63	2000.0	6000.0
P6	3423.33	8142.57	2500.0	6500.0
P3	1021.45	16571.57	1000.0	14500.0.
P7	3932.01	21284.34	3500.0	18500.0
P2	485.25	22927.3	500.0	20000.0
P1	0.0	26605.47	0.0	23000.0

平均誤差 **= 1883.48 (UNIT TIME)**

RR_5

PROCESS	實際結果		理論結果	
P4	1497.14	5802.68	1500.0	5500.0
P5	2010.28	6462.85	2000.0	6000.0
P6	3136.18	7570.24	3000.0	7000.0
P3	985.5	15636.45	1000.0	14500.0
P7	3639.13	20075.06	3500.0	18500.0
P2	468.29	21927.58	500.0	20000.0
P1	0.0	25408.34	0.0	23000.0

平均誤差 **= 1163.81 (UNIT TIME)**

SJF_1

PROCESS	實際結果		理論結果	
P2	0.0	2148.49	0.0	2000.0
P3	2148.6	3229.52	2000.0	3000.0

P4	3229.62	7511.02	3000.0	7000.0
P1	7511.12	15039.51	7000.0	14000.0
平均誤差 **= 259.80 (UNIT TIME)**				

SJF_2

PROCESS	實際結果		理論結果	
P1	0.0	109.45	0.0	100.0
P3	109.55	311.92	100.0	300.0
P2	312.02	5012.7	300.0	4300.0
P4	5012.8	8969.21	4300.0	8300.0
P5	8969.32	16878.62	8300.0	15300.0
平均誤差 **= 333.08 (UNIT TIME)**				

SJF_3

PROCESS	實際結果		理論結果	
P1	0.0	3145.11	0.0	3000.0
P4	3145.21	3154.95	3000.0	3010.0
P5	3155.03	3186.93	3010.0	3020.0
P6	3187.14	7613.41	3020.0	7020.0
P7	7613.51	11849.25	7020.0	11020.0
P2	11849.36	17282.03	11020.0	16020.0
P3	17282.18	24912.44	16020.0	23020.0
P8	24912.54	34236.28	23020.0	32020.0
平均誤差 **= 276.99 (UNIT TIME)**				

SJF_4

PROCESS	實際結果		理論結果	
P1	0.0	3252.67	0.0	3000.0
P2	3252.78	4372.56	3000.0	4000.0
P3	4372.66	8666.11	4000.0	8000.0
P5	8666.22	9762.93	8000.0	9000.0
P4	9763.04	11945.65	9000.0	11000.0
平均誤差 **= 189.04 (UNIT TIME)**				

SJF_5

PROCESS	實際結果		理論結果	
P1	0.0	2189.13	0.0	2000.0
P2	2189.23	2743.11	2000.0	2500.0

P3	2743.21	3265.39	2500.0	3000.0
P4	3265.49	3798.93	3000.0	3500.0
平均誤差 **= 74.66 (UNIT TIME)**				

PSJF_1

PROCESS	實際結果		理論結果	
P4	3183.93	6312.25	3000.0	6000.0
P3	2126.33	10575.56	2000.0	10000.0
P2	1042.0	17094.94	1000.0	16000.0
P1	0.0	26279.51	0.0	25000.0
平均誤差 **= 727.50 (UNIT TIME)**				

PSJF_2

PROCESS	實際結果		理論結果	
P2	959.09	1985.64	1000.0	2000.0
P1	0.0	4139.71	0.0	4000.0
P4	5124.06	7260.11	5000.0	7000.0
P5	7260.21	8390.91	7000.0	8000.0
P3	4139.81	11591.69	4000.0	11000.0
平均誤差 **= 176.98 (UNIT TIME)**				

PSJF_3

PROCESS	實際結果		理論結果	
P2	464.09	966.03	500.0	1000.0
P3	966.13	1493.06	1000.0	1500.0
P4	1493.18	2017.03	1500.0	2000.0
P1	0.0	3564.73	0.0	3500.0
平均誤差 **= 29.36 (UNIT TIME)**				

PSJF_4

PROCESS	實際結果		理論結果	
P3	101.73	1182.04	100.0	1100.0
P2	0.0	3264.76	0.0	3000.0
P4	3264.86	7521.19	3000.0	7000.0
P1	7521.31	14959.38	7000.0	14000.0
平均誤差 **= 259.87 (UNIT TIME)**				

PSJF_5

PROCESS	實際結果		理論結果	
P1	0.0	93.81	0.0	100.0
P3	93.93	281.45	100.0	300.0
P2	281.55	4563.0	300.0	4300.0
P4	4563.1	8857.01	4300.0	8300.0
P5	8857.56	16464.63	8300.0	15300.0
平均誤差 **= 240.20 (UNIT TIME)**				

結論

1. 排程無法如理論結果完美，變更權重的過程、處理訊號的時間、資料結構的處理都會影響 process 間的速度，導致誤差的產生。
2. RR 相較其他三個排程策略有著較高的平均誤差，主要很可能是因為 context switch 過於頻繁導致較大的延遲，也就造就逐漸累積的誤差。
3. 執行時間較久的 process(例如 FIFO_2 的 P1)也會導致更大的誤差產生。
4. Unit time 也是估計的假設，無法完全反映現實的數字。