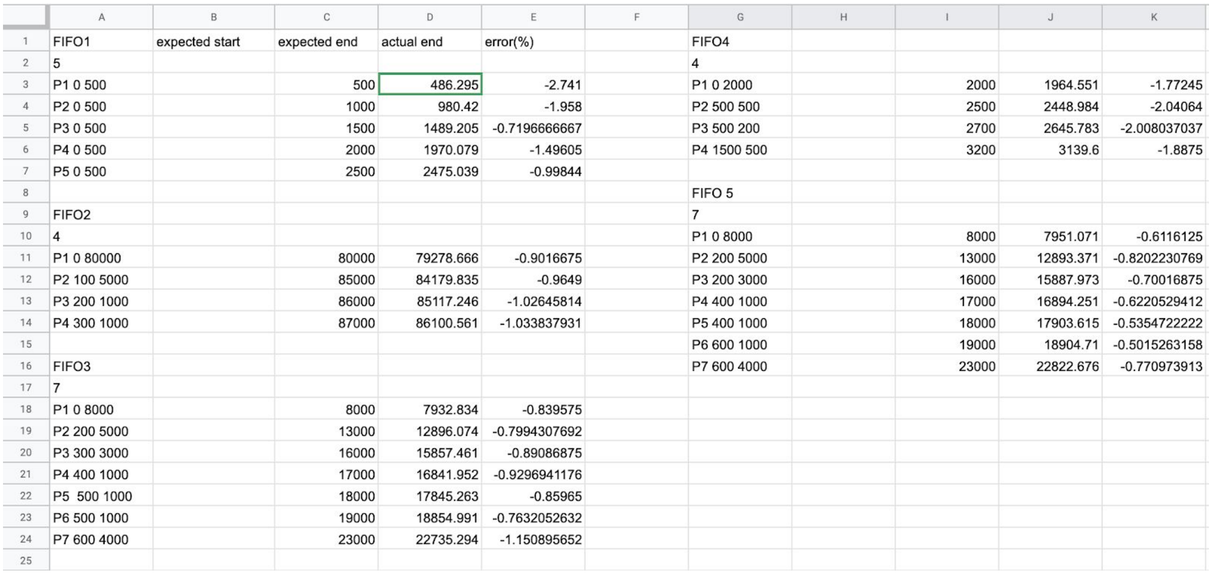
OS Project1 Report

B05202010 章瑋麟

1. 設計
   讀進input之後，程式會紀錄各個process的資料並按照schduling policy呼叫FIFO, RR, SJF或PSJF scheduler進行排程。
   函數proc_create會fork出需要的child process來排程，fork之後child除了使用和parent不同的cpu來跑減少對scheduling performance的衝擊之外，還會先被parent assign到SCHED_IDLE降低優先權。
   而小孩每做一次TIME_UNIT()之前會呼叫read system call block住自己，等到parent使用write給他一個訊息才能繼續，所以說parent要叫一個小孩跑需要：
    1. 提升priority group至SCHED_OTHER
    2. 執行write來unblock child

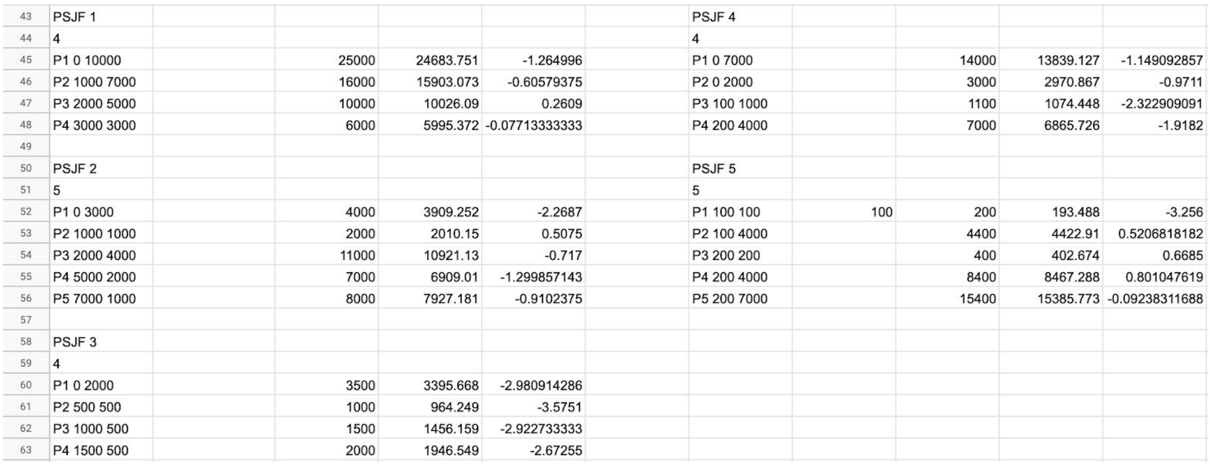
2. 版本：
   Linux 4.14.25 on virtualbox on LENOVO ideapad 710S

3. 比較
   FIFO



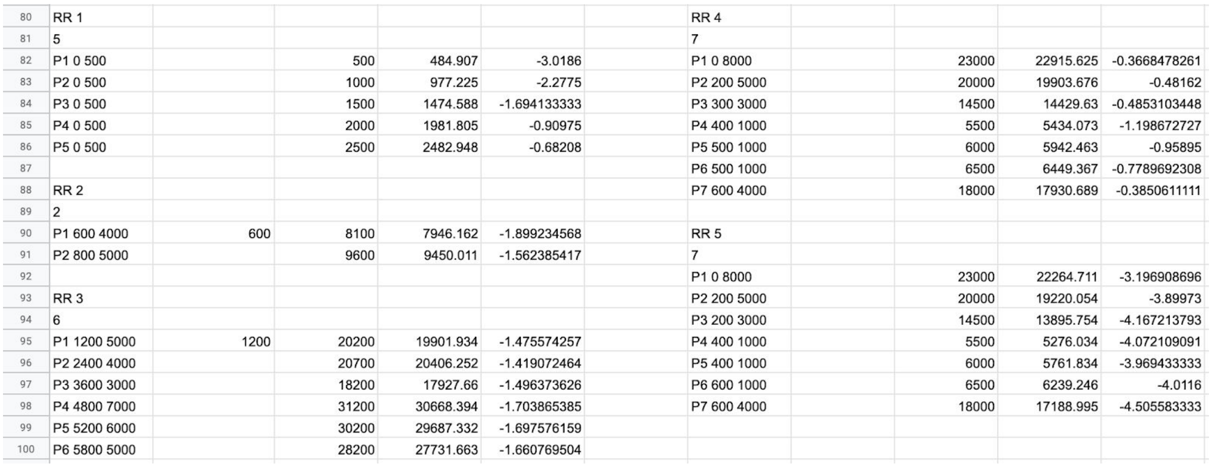
	A	B	C	D	E	F	G	H	I	J	K
1	FIFO1	expected start	expected end	actual end	error(%)		FIFO4				
2	5						4				
3	P1 0 500		500	486.295	-2.741		P1 0 2000		2000	1964.551	-1.77245
4	P2 0 500		1000	980.42	-1.958		P2 500 500		2500	2448.984	-2.04064
5	P3 0 500		1500	1489.205	-0.7196666667		P3 500 200		2700	2645.783	-2.008037037
6	P4 0 500		2000	1970.079	-1.49605		P4 1500 500		3200	3139.6	-1.8875
7	P5 0 500		2500	2475.039	-0.99844						
8							FIFO 5				
9	FIFO2						7				
10	4						P1 0 8000		8000	7951.071	-0.6116125
11	P1 0 80000		80000	79278.666	-0.9016675		P2 200 5000		13000	12893.371	-0.8202230769
12	P2 100 5000		85000	84179.835	-0.9649		P3 200 3000		16000	15887.973	-0.70016875
13	P3 200 1000		86000	85117.246	-1.02645814		P4 400 1000		17000	16894.251	-0.6220529412
14	P4 300 1000		87000	86100.561	-1.033837931		P5 400 1000		18000	17903.615	-0.5354722222
15							P6 600 1000		19000	18904.71	-0.5015263158
16	FIFO3						P7 600 4000		23000	22822.676	-0.770973913
17	7										
18	P1 0 8000		8000	7932.834	-0.839575						
19	P2 200 5000		13000	12896.074	-0.7994307692						
20	P3 300 3000		16000	15857.461	-0.89086875						
21	P4 400 1000		17000	16841.952	-0.9296941176						
22	P5 500 1000		18000	17845.263	-0.85965						
23	P6 500 1000		19000	18854.991	-0.7632052632						
24	P7 600 4000		23000	22735.294	-1.150895652						
25											

PSJF



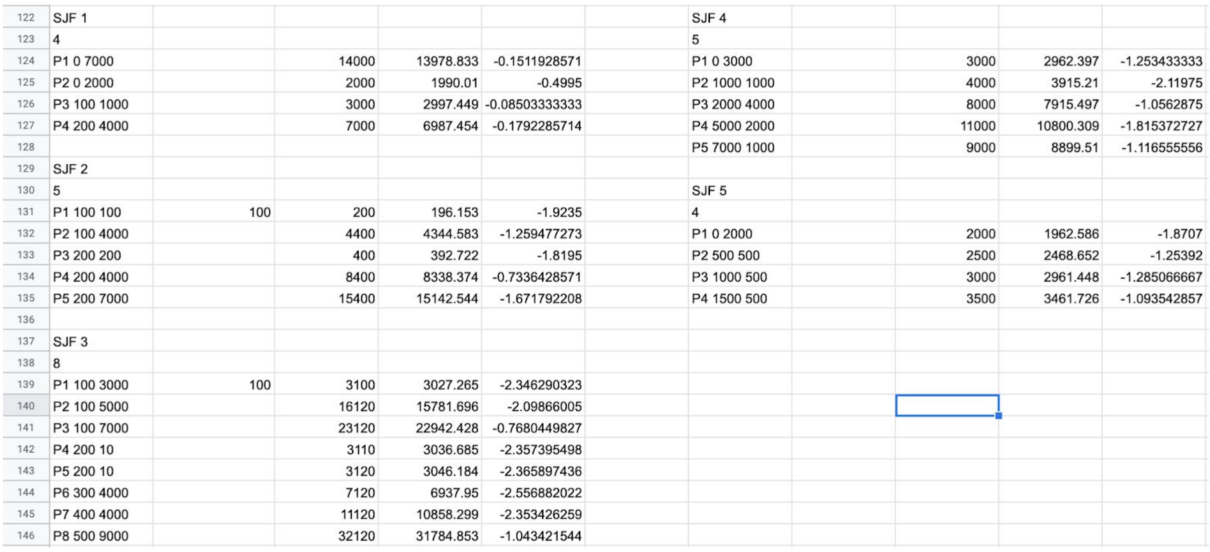
43	PSJF 1						PSJF 4				
44	4						4				
45	P1 0 10000		25000	24683.751	-1.264996		P1 0 7000		14000	13839.127	-1.149092857
46	P2 1000 7000		16000	15903.073	-0.60579375		P2 0 2000		3000	2970.867	-0.9711
47	P3 2000 5000		10000	10026.09	0.2609		P3 100 1000		1100	1074.448	-2.322909091
48	P4 3000 3000		6000	5995.372	-0.07713333333		P4 200 4000		7000	6865.726	-1.9182
49											
50	PSJF 2						PSJF 5				
51	5						5				
52	P1 0 3000		4000	3909.252	-2.2687		P1 100 100	100	200	193.488	-3.256
53	P2 1000 1000		2000	2010.15	0.5075		P2 100 4000		4400	4422.91	0.5206818182
54	P3 2000 4000		11000	10921.13	-0.717		P3 200 200		400	402.674	0.6685
55	P4 5000 2000		7000	6909.01	-1.299857143		P4 200 4000		8400	8467.288	0.801047619
56	P5 7000 1000		8000	7927.181	-0.9102375		P5 200 7000		15400	15385.773	-0.09238311688
57											
58	PSJF 3										
59	4										
60	P1 0 2000		3500	3395.668	-2.980914286						
61	P2 500 500		1000	964.249	-3.5751						
62	P3 1000 500		1500	1456.159	-2.922733333						
63	P4 1500 500		2000	1946.549	-2.67255						

RR



80	RR 1						RR 4				
81	5						7				
82	P1 0 500		500	484.907	-3.0186		P1 0 8000		23000	22915.625	-0.3668478261
83	P2 0 500		1000	977.225	-2.2775		P2 200 5000		20000	19903.676	-0.48162
84	P3 0 500		1500	1474.588	-1.694133333		P3 300 3000		14500	14429.63	-0.4853103448
85	P4 0 500		2000	1981.805	-0.90975		P4 400 1000		5500	5434.073	-1.198672727
86	P5 0 500		2500	2482.948	-0.68208		P5 500 1000		6000	5942.463	-0.95895
87							P6 500 1000		6500	6449.367	-0.7789692308
88	RR 2						P7 600 4000		18000	17930.689	-0.3850611111
89	2										
90	P1 600 4000	600	8100	7946.162	-1.899234568		RR 5				
91	P2 800 5000		9600	9450.011	-1.562385417		7				
92							P1 0 8000		23000	22264.711	-3.196908696
93	RR 3						P2 200 5000		20000	19220.054	-3.89973
94	6						P3 200 3000		14500	13895.754	-4.167213793
95	P1 1200 5000	1200	20200	19901.934	-1.475574257		P4 400 1000		5500	5276.034	-4.072109091
96	P2 2400 4000		20700	20406.252	-1.419072464		P5 400 1000		6000	5761.834	-3.969433333
97	P3 3600 3000		18200	17927.66	-1.496373626		P6 600 1000		6500	6239.246	-4.0116
98	P4 4800 7000		31200	30668.394	-1.703865385		P7 600 4000		18000	17188.995	-4.505583333
99	P5 5200 6000		30200	29687.332	-1.697576159						
100	P6 5800 5000		28200	27731.663	-1.660769504						

SJF



122	SJF 1						SJF 4				
123	4						5				
124	P1 0 7000		14000	13978.833	-0.1511928571		P1 0 3000		3000	2962.397	-1.253433333
125	P2 0 2000		2000	1990.01	-0.4995		P2 1000 1000		4000	3915.21	-2.11975
126	P3 100 1000		3000	2997.449	-0.08503333333		P3 2000 4000		8000	7915.497	-1.0562875
127	P4 200 4000		7000	6987.454	-0.1792285714		P4 5000 2000		11000	10800.309	-1.815372727
128							P5 7000 1000		9000	8899.51	-1.116555556
129	SJF 2										
130	5						SJF 5				
131	P1 100 100	100	200	196.153	-1.9235		4				
132	P2 100 4000		4400	4344.583	-1.259477273		P1 0 2000		2000	1962.586	-1.8707
133	P3 200 200		400	392.722	-1.8195		P2 500 500		2500	2468.652	-1.25392
134	P4 200 4000		8400	8338.374	-0.7336428571		P3 1000 500		3000	2961.448	-1.285066667
135	P5 200 7000		15400	15142.544	-1.671792208		P4 1500 500		3500	3461.726	-1.093542857
136											
137	SJF 3										
138	8										
139	P1 100 3000	100	3100	3027.265	-2.346290323						
140	P2 100 5000		16120	15781.696	-2.09866005						
141	P3 100 7000		23120	22942.428	-0.7680449827						
142	P4 200 10		3110	3036.685	-2.357395498						
143	P5 200 10		3120	3046.184	-2.365897436						
144	P6 300 4000		7120	6937.95	-2.556882022						
145	P7 400 4000		11120	10858.299	-2.353426259						
146	P8 500 9000		32120	31784.853	-1.043421544						

4. 差異的原因

   可以看到所有的誤差都在5%以內，而且幾乎全部都是負值，可能的原因有

   1. 在做TIME_MEASUREMENT的時候有不小心使用到測試中的電腦，造成不穩定的工作負載
   2. 實作的方式使用了pipe, read, write等，child在跑的時候使用了一些io來做IPC，造成誤差
   3. 測試時其他環境誤差
   4. virtualbox所產生的誤差