11027149 資訊三甲 游婕歆

1. 開發環境

C++

2. 實作流程和方法

FCFS :

排序 process：根據 process 的到達時間對所有 process 進行排序。

遍歷每個 process：對排序後的 process 列表進行遍歷，並對每個進程進行計算 waiting 時間和 turnaround 時間，因為是先到先做所以輪到他的時候就直接做完不用管其他事情。

RR :

使用 queue 來模擬等待執行的進程，並使用了 vector 來儲存已完成的 process。再根據 process 的到達時間對所有 process 進行排序。遍歷每個 process：對排序後的 process 列表進行遍歷，並對每個 process 進行以下操作：將進程加入 queue：如果當前時間大於或等於 process 的到達時間，且該進程還未被加入 queue，則將該進程加入隊列。處理 queue 中的進程：當 queue 不為空時，取出 queue 前面的 proces。如果該進程的剩餘時間大於 timrslice，則讓該進程執行一個 timeslice 的時間，並將其剩餘時間減去 timeslice。然後，將該 proces 放回 queue 的尾部。如果該 proces 的剩餘時間小於或等於 timeslice，則讓該 proces 執行其剩餘的時間，並將該 proces 從 queue 中移除，加入到已完成的 proces 列表中。

SJF :

遍歷每個 process：對排序後的 process 列表進行遍歷，並對每個 process 進行以下操作：將 process 加入等待列表：如果當前時間大於或等於 process 的到達時間，且該 process 還未完成，則將該 process 加入等待列表。處理等待列表中的進程：當等待列表不為空時，找出 CPUburst 時間最短的進程。如果該 process 的到達時間大於當前時間，則輸出 “-” 並增加當前時間，直到該 process 的到達時間。然後，將該 process 標記為已完成，並將其從等待列表中移除。最後，根據該進程的 CPUburst 時間來生成輸出結果。

SRTF :

將 process 加入等待列表：如果當前時間大於或等於 process 的到達時間，且該 process 還未完成，則將該 process 加入等待列表。處理等待列表中的 process：當等待列表不為空時，找出剩餘時間最短 process。如果該 process 的到達時間大於當前時間，則輸出 “-” 並增加當前時間，直到該 process 的到達時間。然後，將該 process 標記為已完成，並將其從等待列表中移除。最後，根據該 process 的 CPUburst 時間來生成輸出結果。

HRRN ：

將 process 加入等待列表：如果當前時間大於或等於 process 的到達時間，且該 process 還未完成，則將該 process 加入等待列表。處理等待列表中的 process：當等待列表不為空時，計算每個 process 的反應速率，並找出反應速率最高的 process。反應速率是由等待時間加上 CPUburst 時間，然後除以 CPU burst 時間得到的。然後，將該 process 標記為已完成，並將其從等待列表中移除。最後，根據該 process 的 CPU burst 時間來生成輸出結果。

PPRR ：

跟 rr 作法差不多只不過多了要比 priority。

3. 不同排程法的比較

以 input1 來說明

Waiting Time

ID	FCFS	RR	SJF	SRTF	HRRN	PPRR
0	19	18	5	0	19	0
1	13	8	5	0	5	0
2	22	19	2	2	16	14
3	18	25	6	6	14	0
4	13	19	7	0	13	11
5	20	27	19	19	23	21
6	0	15	5	6	0	11
7	15	2	2	0	3	55
8	21	14	0	0	11	9
9	5	13	0	1	6	0
10	8	37	49	49	18	45
13	18	3	4	0	4	0
20	13	17	5	0	13	40
27	16	28	9	19	9	10
29	14	31	15	19	20	4

FCFS 平均等待了 14.333 秒

RR 平均等待了 18.4 秒

SJF 平均等待了 8.8 秒

SRTF 平均等待了 7. 86 秒

HRRN 平均等待了 11.6 秒

PPRR 平均等待了 14.66 秒

Turnaround Time

ID	FCFS	RR	SJF	SRTF	HRRN	PPRR
0	23	22	9	4	23	4
1	15	10	7	2	7	2
2	25	22	5	5	19	17
3	22	29	10	10	18	4
4	16	22	10	3	16	14
5	26	33	25	25	29	27
6	5	20	10	11	5	16
7	16	3	3	1	4	56
8	23	16	2	2	13	11
9	9	17	4	5	10	4
10	16	45	57	57	26	53
13	19	4	5	1	5	1
20	16	20	8	3	16	43
27	22	34	15	25	15	16
29	20	37	21	25	26	10

以工作完成時間來看
FCFS 273 秒
RR 334 秒
SJF 191 秒
SRTF 179 秒
HRRN 232 秒
PPRR 278 秒

4. 結果與討論

1. FCFS（First Come First Served）：按照process到達的順序進行排程，先到先處理。
優點：實現簡單，無須對process進行優先級別排序，適用於短process。
缺點：無法優先處理重要的process，可能會發生長process佔用資源時間過長的問題。
2. RR（Round Robin）：時間片輪轉法，按照時間片輪流執行每個process。
優點：公平，適用於多個process佔用資源的情況，能夠確保每個process都有執行的機會。
缺點：可能會出現過多的上下文切換，增加了系統開銷。
3. SJF（Shortest Job First）：按照process所需的處理時間進行排序，先處理處理時間最短的process。
優點：能夠最大程度地減少平均等待時間和平均周轉時間。
缺點：可能會出現長作業等待時間過長的問題。
4. SRTF（Shortest Remaining Time First）：按照process還需處理的時間進行排序，先處理還需處理時間最短的process。
優點：能夠更加準確地估計每個process所需的時間，從而更好地優化系統性

能。
缺點：可能會出現長作業等待時間過長的問題。
5. HRRN（Highest Response Ratio Next）：按照反應時間進行排序，先處理反應時間最高的process。
優點：能夠優先處理等待時間較長的作業，避免長作業佔用資源時間過長。
缺點：計算複雜，需要考慮作業等待時間和作業執行時間的比例。
6. PPRR：適用於需要優先處理高優先級 process，高優先級 process 能夠優先執行，從而提高系統效率。

一開始寫 rr 的時候卡住了很久，所以先跳到 hrrn 寫完再寫 sjf 發現其實跟 hrrn 的寫法差不多只不過比較的東西不一樣而已，所以 sjf 不到 10 分鐘就寫完了，寫完 rr 的時候寫 pprr 本來想說可以直接用 rr 來改，但我發現我偷吃步時間不是一秒一秒加，所以會一直錯過時間，後來一秒一秒加才很快改完 pprr。