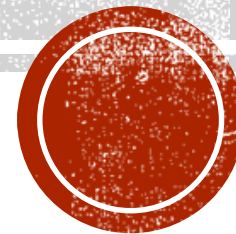


PROJECT_2

各 Scheduler 實作
&
效能比較
&
多處理器排班



(%)PART A — SCHEDULER 實作

- 目標：了解各種排班方式(單處理器)的觀念與實作。
- 作業要求：
 - ✓ 實作以下排班機制：
 1. First-Come, First-Served (FCFS) Scheduling
 2. Shortest-Job-Next (SJN) Scheduling
 3. Shortest-Remaining-Time-First Scheduling
 4. Priority Scheduling
 5. Round-Robin Scheduling
 - ✓ 在報告中介紹自己的程式碼(含註解)，請勿只貼程式碼+註解，最好有 work flow 圖。
 - ✓ 可紀錄的資訊(供 part B 使用，自己決定要記錄的資訊)：
 1. 總置換次數(context switch 發生的次數)
 2. 最長的等待時間、平均等待時間，並判斷是否有 starvation、infinite blocking 的現象。
 3. 固定時間內完成的任務數量(或比例)。
 - ✓ 限用 C/C++。



(40%)PART A — SCHEDULER 實作 CONT.

- 輸入資料如右圖，資料包含：

[Process id, 優先權, 執行時間, 到達時間]

- 優先權以數字小的優先
- 若排班機制不需要優先權則不用考慮(如：FCFS、SJN等)，
SJN 以服務時間的倒數作為優先權。
- ~~Round Robin Scheduling~~ 假設所有程序在同時間到達，都已經可以準備排班。
[更正為：按照到達時間處理]
- 輸入的資料分三種：data_1(10000筆)、data_2(1000筆) 皆符合 Poisson 分布，data_1 產生的程序較頻繁執行時間短；data_2 則是執行時間長，但數量少；data_3 每一個單位時間產生一個程序。
- 到達時間相同是可能發生的，依據優先權或 process_ID 選擇順序。

P43	2	7	437
P44	6	10	448
P45	10	14	459
P46	7	16	467
P47	3	14	477
P48	10	12	486
P49	10	16	496
P50	7	8	504
P51	8	8	508
P52	7	8	521
P53	8	13	531
P54	5	7	539
P55	4	13	550
P56	5	7	559
P57	5	6	565
P58	2	13	572
P59	1	11	582
P60	1	10	594
P61	8	14	606



(40%)PART A — SCHEDULER 實作 CONT.

- 建議先以 data_3 來做 Round-Robin 排班。
- Round-Robin 簡介：
 - ✓ 更正講解時說將所有行程視為同時到達的情況。
 - ✓ Data_3 每一個程序間都間格一個單位時間
 - ✓ 到達的程序就加入等候佇列內，到達切換的時間時，若目前的程序還未做完則放到等候佇列的最後方。
 - ✓ 若進行切換的同時有新的程序出現，新的程序先進放等候佇列內，再把當前程序放到最後一個。
- 排班時都先假設：置換時間為零、佇列的插入、提出時間也為零。



(30%)PART B – 效能分析

- 目標：討論各排班程式效能與適用情境。
- 作業要求：
 - ✓ 將 **part A** 各排班的程式進行效能比較，分析(繪圖、做成表格佳)：
 1. 等待時間、平均等待時間、是否發生飢餓、無限期阻塞等，以
 2. 回復時間(Turnaround time)。
 3. 產量(throughput)，單位時間完成的工作數。
 - ✓ 討論各排班適用的情境。
- 比較情境自訂：
 - ✓ 自行假設置換需要的時間進行比較(置換速度可快可慢)。
 - ✓ 自行設定總執行時間(分析單位時間內完成的程序比率)。
 - ✓ Round-Robin Scheduling 的切換周期自訂。
 - ✓ 各種情境、參數都自訂，需要在報告中明確說明。



(30%)PART B — 效能分析

- 自訂比較的內容(如：吞吐量、等待時間等)。
- 實際上置換是會有時間耗損的，因此分析時請務必設定置換的時間。
- 請大家發揮創意，任何關於效能的探討都是允許的。



(30%)PART C — 多處理器排班

- 目標：認識多處理器的排班概念。
- 多處理器的排班較複雜，可分為 **SMP** 架構與 **ASMP** 架構，這部分請蒐集資料完成報告。
- 作業要求：
 - ✓ 找一篇論文或技術文件或現行的排班方法，介紹多處理器的排班是如何運行的。
 - ✓ 報告中附上論文、文件名稱。
 - ✓ 善用 **google scholar**，搜尋“**multiprocessor scheduling**”等關鍵字。
 - ✓ 介紹：概念、流程、時間複雜度等計算，愈詳細分數愈高。



繳交方式

- 本次為團體作業，「每組」繳交一份報告(.pdf)，頁數不限，誠意至上。
- 繳交之檔案分別命名為 **report.pdf** 與 各排班程式碼 (加上註解)，放置於"OS_PJ2_組別"的資料夾進行壓縮，以zip格式寄至助教信箱，信件主旨為"OS_PJ2_組別"。
- 繳交期限：2019/5/17(五) 23:59，**不接受補交**。
- **嚴禁抄襲，抄襲者整份 project 0分計算，以下情況皆視為抄襲：**
 - ◆ 參考但未附上參考資料
 - ◆ 複製貼上別人的文章、維基百科、stackoverflow 等
 - ◆ "改寫"參考資料，請理解後以自己的方式說明
- 未依循上述規則斟酌扣分。
- Email： course.itlab@gmail.com

