**OS Project2 document**

**資訊三甲**

**10627149**

**許峯僑**

**開發環境:** window10  
**開發軟體:** Dev C++

**設計流程:**

**FCFS =>** 我的想法上很直接，直接透過函數庫Algorithms的sort功能將vector依照抵達時間的先後順序安排，再根據CPU burst time印出相對應次數即可完成。

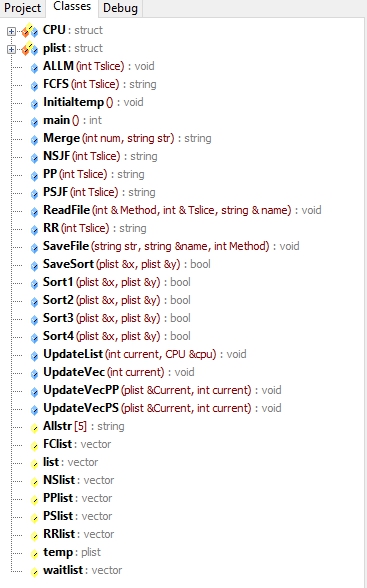
**NSJF =>** 透過函數庫Algorithms的sort功能並且自行設定sort中bool的規則，將CPU burst time最小的優先處理，若相同再來依照抵達時間最後透過id來做分辨。最後再根據CPU burst time印出相對應次數即可完成。

**PSJF =>** 透過函數庫Algorithms的sort功能並且自行設定sort中bool的規則，將CPU burst time最小的優先處理，其中還特別在vector當中多設定一個bool isExe = false，這個是用來判斷是否進入過cpu(執行過這個process)。透過這個變數再增加至sort當中的bool規則其餘的規則相同，依照抵達時間最後透過id來做分辨。最後再根據CPU burst time印出相對應次數即可完成。

**PP =>** 透過函數庫Algorithms的sort功能並且自行設定sort中bool的規則，將Priority number最小的優先處理，其中還特別在vector當中多設定一個bool isExe = false，這個是用來判斷是否進入過cpu(執行過這個process)。透過這個變數再增加至sort當中的bool規則其餘的規則相同，依照抵達時間最後透過id來做分辨。最後再根據CPU burst time印出相對應次數即可完成。

**RR =>** 我認為這個是所有處理方式當中最複雜的，其中還要考慮到很多其餘的條件，不單單只是透過sort 即可完成，其中還要特別考慮time slice是否大於cpu burst time，我是透過這個方式來分辨要做甚麼樣的處理，如果time slice小於cpu burst time我將該process的cpu burst time減少time slice再將他放入waitting list當中。若是time slice大於cpu burst time即可知道這個process已經完成作業，記錄他的turnaround time 和 waiting time 即可。

**程式function介紹:**



主要是透過vector的方式將各個list的結果和所有資料儲存起來。

在最剛開始讀檔時統一用list這個vector的方式去讀取，在執行各個function時會在為各自的function設計一個vector將結果儲存起來，方便在最後寫檔的時候可以輕鬆寫出所有的資訊。

**心得:**

　　其實這隻程式我從很早就開始一步一步慢慢寫了，並非最後一周才開始寫，所以其中我碰到了很多理解上不同的問題。在程式說明的部分跟我想法上設計出來的程式不同，也導致我一直碰壁沒辦法在第一時間直接將程式完成，透過問同學問老師才了解到程式的想法。也才了解到原來我的想法和程式設計的想法有些出入。也很感謝助教們後來在課程公告那邊有額外的作補充，補充內容十分詳細這才讓我寫程式十分順利，在寫程式後期就沒有碰到任何問題。雖然同學們仍然有時候會對助教的程式解答產生疑慮，但是我因為很早就開始寫所以大致上也了解程式說明文件那些說不清楚的盲點，也一一介紹給我身邊的朋友讓他們可以避免發生程式設計理念不同的問題。