作業系統

HW2

實作CPU Scheduling

班級：資訊三甲

學號：10627116

姓名：許逸翔

老師：鍾斌賢

1. 開發平台
   1. 作業系統：Windows10教育版 64bit
   2. CPU：i5-7200U 2.5GHz 四核心
   3. RAM：8GB DDR4-2134 雙通道
2. 使用開發環境
   1. IDE：Visual Studio
   2. Language：C++
   3. GDB：windows SDK 10.0.18362.0
3. 完成的功能
   1. FCFS、RR、PSJF、NSJF、Priority等CPU Scheduling
   2. FCFS、NSJF單純照順序排，Process完成時才輪替
   3. RR：Process完成或time out才輪替
      1. RR需要更新arrival time
      2. RR需要Process完成時和CPU空閒時歸零time slice
      3. RR要紀錄是否被使用過
   4. PSJF：每秒都要判斷搶奪
      1. PSJF需要更新CPU Burst
      2. PSJF搶奪時只比較CPU Burst，排隊才需要考慮其他因素
      3. PSJF要紀錄是否被使用過
   5. Priority：每秒都要判斷搶奪
      1. Priority搶奪時只比較Priority，排隊才需要考慮其他因素
      2. Priority要紀錄是否被使用過
4. 使用到的函式庫
   1. 使用<iostream>來完成螢幕I/O
   2. 使用<fstream>來完成資料I/O
   3. 使用<iomanip>來控制輸出文件格式
   4. 使用<string>來處理字串
   5. 使用<deque>存資料，因為需要pop\_front
5. 資料結構
   1. Dataset
      1. INPUT (1 mode)：deque < Process Control Block >
      2. OUTPUT (1 mode)：deque < Process Control Block >
      3. OUTPUT (5 mode)：deque < deque < Process Control Block > >
   2. Process Control Block
      1. ID
      2. CPU\_burst(會更新) / CPU\_burst\_o(不會更新)
      3. arrival(會更新) / start(不會更新)
      4. priority
      5. CPU\_actual(目前已用時間)
      6. CPU\_last(最後使用時間，判斷有沒有使用過)
   3. 運用方式
      1. 當被執行到時CPU\_burst--，CPU\_actual++，CPU\_last=當前時間
      2. 使用CPU\_burst\_o == CPU\_actual來判斷是否完成
      3. 當被搶奪或time out時arrival要更新
      4. Waiting time = CPU\_last(後) – start(前) + 1 – CPU\_burst\_o(使用時間)
      5. Turnaround time = CPU\_last(後) – start(前) + 1
6. 流程
   1. Main
      1. 使用者將給定的資料準備
      2. 程式依據文件敘述的選擇CPU Scheduling模式
      3. 讀取資料、進行模擬並保存結果
   2. Simulate
      1. 將資料依據「arrival time、(Priority)、ID」排序，完成進入順序
      2. 判斷當前時間有沒有程式要執行
      3. 更新arrival time和CPU Burst
      4. 判斷Process是否完成
      5. 不進行判斷/判斷是否搶奪/判斷是否timeout (mode:14/35/2)
      6. 重新依據mode排序，回到4.2.2