

2020_OS_Fall_HW1: Benchmark Your Computer Black Box

繳交期限：09/21-10/06 00:00

作業目標

- 請撰寫一支數值排序的程式，其功能是能夠將數值由小排序到大，且排序的資料量會遠大於電腦記憶體容量。
- 請觀察及分析程式執行期間，包括但不限於CPU、Memory、Disk I/O的使用情況，探討作業系統是如何服務我們的程式。

測試資料

請撰寫一支程式，自行產生出符合以下條件的測試資料，做為此次作業的輸入資料。

- 請在**單一檔案**產生N個亂數數值，並以換行字元 (LF) 分隔，做為待排序的資料。
 - 數值範圍: $-2^{31} \sim 2^{31}-1$ (4 bytes int)
 - 請注意，換行字元的前後不要包含空白字符。
- 測試資料的檔案大小，**至少要大於你電腦的實體記憶體大小**。
 - 這代表產生的資料量將會非常龐大，你所撰寫的排序程式必須要能夠妥善處理它。
- 請以 UTF-8 的字元編碼儲存，並將此測資檔案命名為「**input.txt**」。

你所產生的測試資料 (input.txt)，應該會長得像這樣：

1016805454
1475107312
-773922153
-1631946082
-2117758124
1441150487
-1097602902
790246949
-323167788
1476584887
-1453834325
-27133615
220352360
-1417575783
1375480777
-1264143939
-1709228479
1737296785
1750366440
-2075608694
-1480735945
1708474629
...(略)

如何開始

1. 請依照前項的說明，產生出撰寫此作業需要使用的輸入資料「input.txt」。
2. 請撰寫一支數值排序的程式，並將排序後的結果儲存為「output.txt」。
 - 請注意，你所撰寫的排序程式必須要能夠處理龐大的資料量。
3. 請使用任意工具或方法分析、觀察你所撰寫的程式，並優化你的程式（例如：降低執行時間）。
 - 試著用盡電腦的計算能力，讓它集中在處理你的程式上，以電腦閒置資源最小化為目標。
4. 將你所觀察到的現象，試著思考作業系統背後的行為，撰寫出一份完整的效能分析報告。
 - 此部分的內容將會是作業評分的重點，盡你所能說明的越詳盡越好。
 - 請嘗試往觀察CPU、Memory、Disk I/O的使用情形著手，對上述系統資源的觀察，並試著去結論作業系統應該要如何服務這些程式，使每隻程式能得到最好的服務。
 - 請嘗試執行自己撰寫的排序程式若干支，就是讓系統同時有多支相同的程式執行（假設你所撰寫的是Single-thread程式），這時作業系統會負責切換這些程式的執行，你可以測量這一批程式執行的時間，觀察及討論作業系統在背後做了什麼有利於這麼多支程式的"事"。
5. 將你撰寫的程式碼及說明文件，依照作業繳交的規定，於期限內上傳到Moodle平台。

說明文件

說明文件的格式不拘，你也可以使用Markdown來完成。
你在撰寫此份說明文件時，必須要包含下列基本內容。

學號：

姓名：

系級：

開發環境：

- OS: Ubuntu 20.04.1
- CPU: Intel® Core™ i7-10700 CPU @ 2.90GHz × 16
- Memory: 32GB
- Programming Language(version): Java 1.8.0_261

程式執行時間：

- 請在你的程式中加入量測執行時間的程式碼，以精準的獲取此數值。

程式開發與使用說明：

- 你是如何開發這支排序程式，以達成支援大資料排序的行為？
- 你的程式該如何使用，請詳細說明執行的步驟。

```
1  # 請確保助教能夠按照此步驟執行你的程式。
2  # 並讓程式有地方可以取得測試資料的路徑。
3  # Compile
4  $ javac ./YourSourceCode.java
5  # Run
6  $ java ./YourSourceCode [Data Path]
```

效能分析報告：

請注意，分析報告的內容必須包含下列兩個部分。

- 你開發的排序程式對硬體效能優化程度的說明與驗證。
- 在**同時運行**多支你所開發的排序程式下，你對系統效能的觀察，並結論OS的設計要提供哪些優化服務。
 - 你可以搭配圖片、圖表或外部資料來說明。
 - 這部分將會是此作業評分的重點，請同學盡量發揮。

作業繳交

- 繳交期限：09/21-10/06 00:00

- 逾期繳交將按下規則採連續扣分。
 - 逾期一日：得分扣10分。
 - 逾期二日：再扣20分。
 - 逾期三日：再扣30分。
 - 逾期四日以上：得分以0分計算。
- 請將你的「程式原始碼」、「說明文件」打包成ZIP壓縮檔（請以HW1_你的學號.zip命名）。
 - 說明文件 -> 請繳交Markdown（.md），或是PDF檔案。
 - 若你使用HackMD寫文件，可以接受以匯出的HTML檔案繳交。
 - 若你用其他形式撰寫文件，請轉換成PDF格式繳交。
 - 不需要繳交測試所使用的輸入檔（input.txt）及排序後的輸出檔（output.txt）。
- 請再次確認你的程式能夠正常執行，且說明文件中有包含指定的內容。
- 請將打包好的壓縮檔，上傳到Moodle的作業中，即可完成此次作業的繳交。

評分項目

作業會以下列原則評分，滿分為100分。

- 排序程式
 - 排序程式是否能夠處理龐大的資料量。
 - 助教會使用數種不同大小的測試資料（如：8G、16G、32G、64G、128G等）來驗證程式能否正確執行。
 - 排序程式的執行效率（速度）。
 - 對排序程式的優化程度。
- 說明文件：效能分析報告的內容是本項評分的重點。
 - 文件是否有包含指定的內容。
 - 報告內容的完整度及正確性。
 - 對系統資源觀察的程度以及呈現的說明內容。
 - 如何優化這支排序程式，驗證與說明的完整度。

注意事項

- 你可以使用任意程式語言撰寫作業，但助教只會用Ubuntu環境執行你的程式，因此建議使用Linux OS來撰寫此份作業。
- 你可以用任何OS(Windows、Windows Subsystem for Linux、Virtual Machine...)開發，但是就像上一點說的，助教只會在Ubuntu中執行你的程式。
- 你可以在 Virtual Machine 中開發，一樣是分析在 Virtual Machine 中的效能，並在開發環境的部份註記是使用 Virtual Machine。
- 請不要直接調用外部的函式庫來幫助你達成排序大檔案的行為。

- 嚴格禁止互相抄襲程式碼，助教會進程式碼比對，違者此次作業以零分計算。

參考資料

- Merge sort (https://en.wikipedia.org/wiki/Merge_sort).
- External sorting (https://en.wikipedia.org/wiki/External_sorting).
- Linux Perf (<http://wiki.csie.ncku.edu.tw/embedded/perf-tutorial>).