**一、開發環境 在主機上使用虛擬環境開發**

主機配備: Intel® Core™i7-9750H CPU @ 2.60GHz, 24G RAM

虛擬機: 2 核心處理器、8192MB 記憶體

作業系統: Linux 64bit (Ubuntu 18.04.5)

程式語言: C++

**二、實作方法與流程**

主程式流程:

1. 輸入檔名並保存資料

2. 輸入切割的份數、欲執行的方法

3. 計時開始

4. 依據輸入的方法，執行相應的排序

5. 計時結束

6. 寫檔

7. 回到步驟1.

氣泡排序:

1. 外迴圈為要執行排序的次數 。

2. 內迴圈選定開始排序的元素並比較，若當前的元素較大則交換，直到比較至特定元素。

合併:

1. 傳入大陣列(arr)、要排序的資料總比數、要排序的範圍(left、mid、right) 。

2. 依據要排序的範圍(left ~ mid ; mid ~ right)，複製陣列到兩個子陣列內。

3. 比較兩個子陣列的元素，較小者放回大陣列。

4. 重複步驟 3，直到某一子陣列比較完。

5. 檢查兩子陣列是否都完成比較，若沒有，將剩餘元素放回大陣列。

方法一(直接氣泡排序):

1. 呼叫氣泡排序函式做排序

方法二(切割成 K 份氣泡排序後合併):

1. 將 N 份資料切割成 K 份，算出其每筆資料的範圍，並傳入氣泡排序函式做排序 。

2. 以迴圈方式呼叫合併函式，以K 值為合併後份數，檢查 K是否大於1，若大於1，將 K 份

資料，兩兩傳入合併函式做合併。

3. 更新當前份數(k)，並回到步驟2。

方法三(多處理元對 K 份資料氣泡排序後合併):

1. 建立一份儲存 N 筆資料的共享記憶體。

2. 將 N 份資料切割成 K 份，算出其每筆資料的範圍。

3. 以迴圈方式建立 K 個處理元，每個處理元傳入一份資料，進行氣泡排序。

4. 使用 waitpid()來等待每個 Fork 出去的 Child Process 返回。

5. 以迴圈方式呼叫合併函式，以 K 值為合併後份數，檢查 K是否大於1若大於1，建立 K-1

個處理元，每個處理元傳入兩份資料，進行合併。

6. 使用 waitpid()來等待每個 Fork 出去的 Child Process 返回。

7. 更新當前份數(k)，並回到步驟5 。

方法四(多執行緒對 K 份資料氣泡排序後合併):

1. 將 N 份資料切割成 K 份，算出其每筆資料的範圍。

2. 以迴圈方式建立 K 個執行緒，每個執行緒傳入一份資料、氣泡排序函式做排序。

3. 使用 pthread\_join()函式來等待每個建立出的執行緒返回。

4. 以迴圈方式以 K 值為合併後份數，檢查 K是否大於1若大於1，建立 K-1 個執行緒，每個

執行緒傳入兩份資料、合併函式做合併。

5. 使用 pthread\_join()函式來等待每個建立出的執行緒返回。

6. 更新當前份數(k)，回到步驟4。

**三、探討結果和原因:**

實驗一. 相同K值不同筆資料: 單位s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K = { 20 } | 1w | 10w | 50w | 100w |
| 方法一 | 0.267 | 28.699 | 500.905 | 1813.263 |
| 方法二 | 0.024 | 1.182 | 36.791 | 134.944 |
| 方法三 | 0.04 | 0.614 | 19.444 | 75.227 |
| 方法四 | 0.032 | 0.604 | 19.007 | 74.454 |

以實驗一來看，氣泡排序的執行效率為 O(n^2)來看，切割成20份，方法二執行時間比應該是20\*(N/20)^2，和實驗結果大概相同(方法一和方法二的速度差大概20倍)。雖然方法三、方法四與方法二只差大概2倍，但還是有增加其執行效率。

實驗二. 不同K值同筆資料: 單位s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N = {100w} | k = 20 | k = 100 | k = 200 | k = 400 |
| 方法一 | 1813.263 | 1813.263 | 1813.263 | 1813.263 |
| 方法二 | 134.944 | 24.622 | 10.597 | 4.372 |
| 方法三 | 75.227 | 12.605 | 5.586 | 2.463 |
| 方法四 | 74.454 | 12.562 | 5.579 | 2.283 |

切成不同份，對單一一個方法也有差，以方法二來看，切成20份需要約135秒，切成100份需要約25秒，兩者約差5倍，與切的份數剛好差5倍；切成100份與切成200份的速度也大概差2倍。

其他發現: 原本我虛擬機用1核8192MB記憶體，方法二、三、四執行結果很接近，後來改成2核8192MB記憶體後，方法二與方法三、四才有所差距。後來拿別人在他自己電腦上的程式碼來跑，他的環境是6核，在相同的份數(k=3)、資料量(10w)、執行的任務(3)下，跑出來的結果也與他自己電腦的不符合(與他的結果差大概3倍)。