OS Program Exercise II

* 上傳壓縮檔(以學號為檔案名)，壓縮檔內含有文件檔與程式檔。
* 文件內容:

1. 姓名、學號 2. 實作題目 3. 實作之程式語言與平台
4. 實作之程式**檔案** 5. 程式主流程 6. 重要的程式碼或副程式
7. 執行結果 (截圖) 8. 心得

** 以下任選一題實作

- 選 4.21 最高分 = 75
- 選 4.26 最高分 = 90
- 選 4.28 最高分 = 100

4.21 Write a **multithreaded program** that **calculates various statistical values** for a list of numbers. This program will be **passed a series of numbers on the command line** and will then **create three separate worker threads.** One thread will determine the **average** of the numbers, the second will determine the **maximum** value, and the third will determine the **minimum** value. For example, suppose your program is passed the integers

90 81 78 95 79 72 85

The program will report

The average value is 82

The minimum value is 72

The maximum value is 95

The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the **parent thread will output the values once the workers have exited.** (We could obviously expand this program by creating additional threads that determine other statistical values, such as **median** and **standard deviation**.)

4.26 The Fibonacci sequence is the series of numbers 0,1,1,2,3,5,8,....

Formally, it can be expressed as:

$fib_0 = 0$; $fib_1 = 1$; $fib_n = fib_{n-1} + fib_{n-2}$

Write a **multithreaded program that generates the Fibonacci sequence**. This program should work as follows: On the command line, the user will enter the number of Fibonacci numbers that the program is to generate. The program will then **create a separate thread that will generate the Fibonacci numbers, placing the sequence in data that can be shared by the threads** (an array is probably the most convenient data structure). When the thread **finishes** execution, the **parent thread**

will **output** the sequence generated by the child thread. Because the parent thread cannot begin outputting the Fibonacci sequence until the child thread finishes, the parent thread will have to wait for the child thread to finish. Use the techniques described in Section 4.4 to meet this requirement.

4.28 A chat room, coded with multi-thread programming. Modify previous echo-server by multi-thread programming and echo message back to all connected clients.