OPERATING SYSTEM

Name: Vanya Ahmed Siddiqui.

Section: BSCS-01 (B)

REG# 200901095.

Submit to: Ma'am Asia.

DECEMBER 28, 2022

TASK-1

CODE:

```cpp
#include <iostream>
#include <thread>
#include <vector>

// Function to merge two sorted arrays into a single sorted array
std::vector<int> merge(std::vector<int> left, std::vector<int> right)
{
    std::vector<int> result;
    while (left.size() > 0 && right.size() > 0)
    {
        if (left[0] < right[0])
        {
            result.push_back(left[0]);
            left.erase(left.begin());
        }
        else
        {
            result.push_back(right[0]);
            right.erase(right.begin());
        }
    }

    while (left.size() > 0)
    {
        result.push_back(left[0]);
        left.erase(left.begin());
    }

    while (right.size() > 0)
    {
        result.push_back(right[0]);
        right.erase(right.begin());
    }

    return result;
}

void mergeSort(std::vector<int>& arr, int start, int end)
{
    if (start < end)
    {
```

```cpp
        int mid = (start + end) / 2;
        std::vector<int> left(arr.begin() + start, arr.begin() + mid + 1);
        std::vector<int> right(arr.begin() + mid + 1, arr.begin() + end + 1);

        std::thread t1(mergeSort, std::ref(left), 0, left.size() - 1);
        std::thread t2(mergeSort, std::ref(right), 0, right.size() - 1);

        t1.join();
        t2.join();

        arr = merge(left, right);
    }
}

int main()
{
    std::cout << "Enter the size of the array: ";
    int size;
    std::cin >> size;

    std::cout << "Enter the elements of the array: ";
    std::vector<int> arr(size);
    for (int i = 0; i < size; i++)
    {
        std::cin >> arr[i];
    }

    // Perform the merge sort algorithm in a separate thread
    std::thread t(mergeSort, std::ref(arr), 0, arr.size() - 1);
    t.join();

    std::cout << "Sorted array: ";
    for (int i = 0; i < size; i++)
    {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}
```
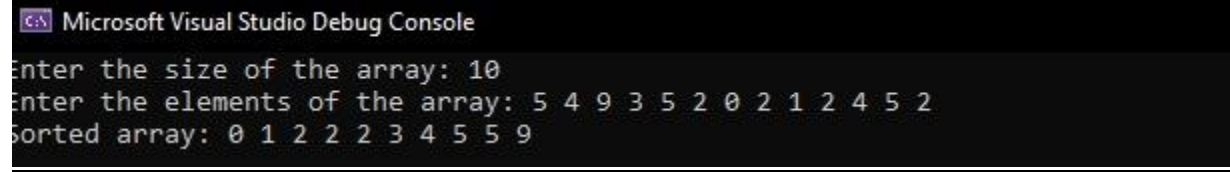
OUTPUT:


```
Microsoft Visual Studio Debug Console

Enter the size of the array: 10
Enter the elements of the array: 5 4 9 3 5 2 0 2 1 2 4 5 2
Sorted array: 0 1 2 2 2 3 4 5 5 9
```

EXPLANATION

INTRODUCTION:

In this assignment we have used the concept of multithreading with the help of the algorithm of merge sort.

MERGE SORT AND MULTIPLE THREADS WHY?

Merge Sort is an algorithm in data structures that works on divide and conquer rule. The reason for using merge sort with multi-threading is because multiple threads are used to performed different task, they are lightweight and is a way to improve parallelism by running the threads simultaneously in different cores of your processor. While merge sort divides an array or list into 2 halves and when it reaches to its depth it starts to merge the array in a sorted way. The time complexity of Merge sort is O(nlogn).
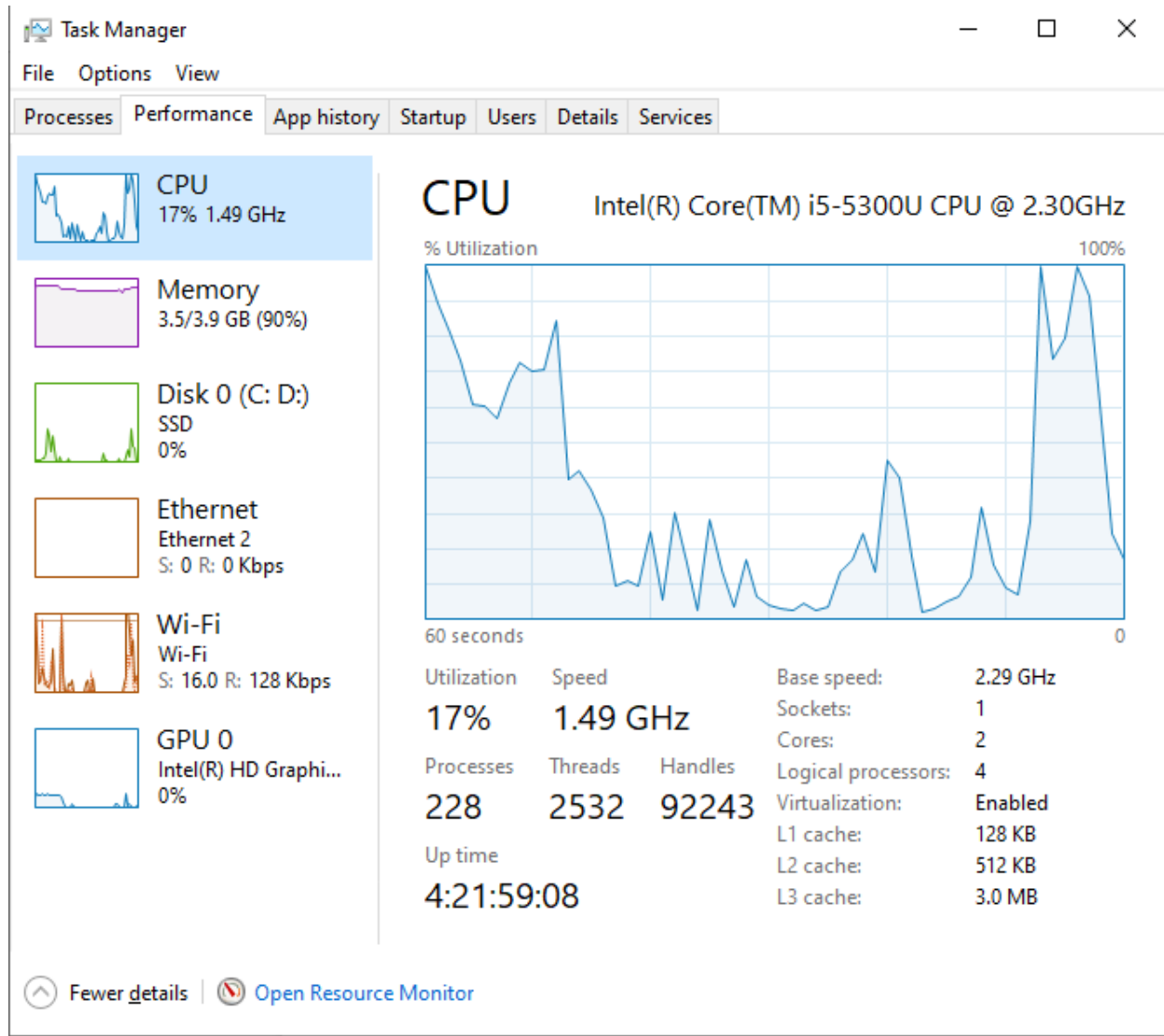
The multiple thread works by sub dividing the program into small chunks of executable threads. Each thread has its programming element that is needed to execute the main program and, each thread is executed one at a time.
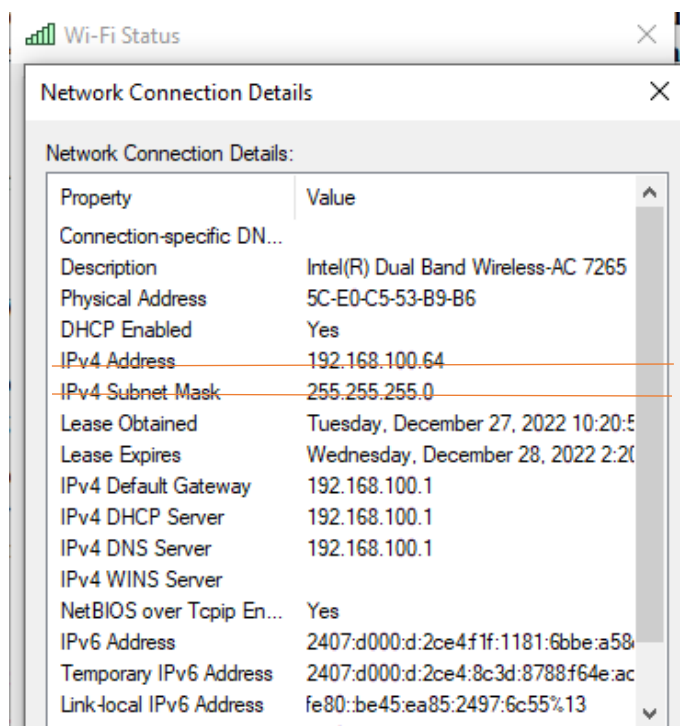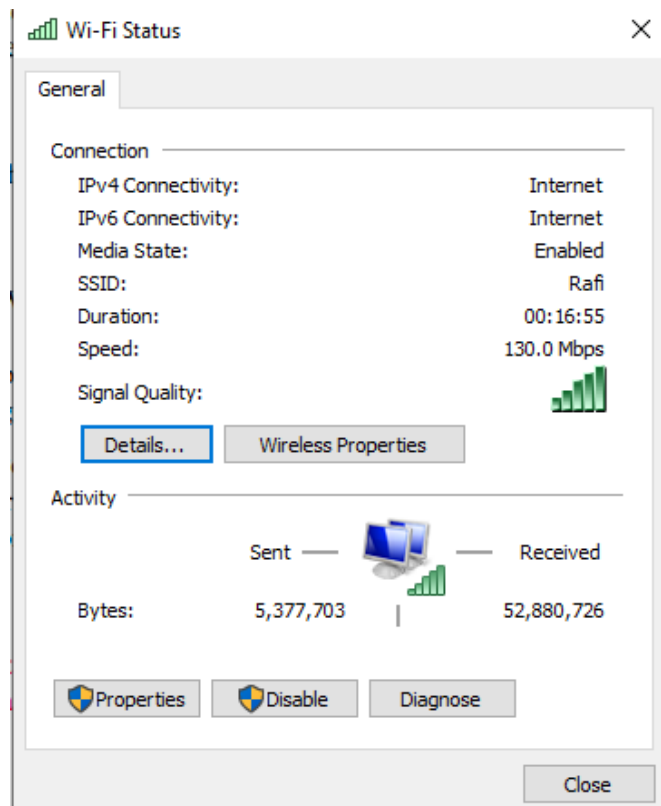
PROGRAM:

In this program we have taken the array from the user and sort it using multithreading with the help of merge sort algorithm. we have taken the core of the CPU (with the help from the **Task Manager)** as thread or threads (the threads of the CPU are different for every user). we have implemented the code as mentioned above with respect to our threads. for example, the cores of my CPU were 2 so I implemented the code by creating the **func ()** of multithreading which sort the array with the concept of merge sort and join the 2 threads. The code will be implemented according to the use of 2 threads only. Suppose if you have 4 threads then you would use the concept of 4 multiple threads and join them with the help of merge sort to sort the array.

SCREENSHOTS:

CORES AND LOGICAL PROCESSOR:

MAC ADRESS:

MAC ADRESS:

5C-E0-C5-53-B9-B6

REFRENCE:

https://openai.com/

GIT_HUB LINK:

https://github.com/vanyaahmed20?tab=repositories

……