# A picture containing text, clipart, vector graphics Description automatically generated

# OPERATING SYSTEMS

## Assignment 03

Name: Muhammad Ahmad Siddique

Reg No. 200901077

BS (CS-01 B)

Submitted To. Ma’am Asia Aman

Date: 12/28/2022

**Multithreaded merge sort Code:**

#include <iostream>

#include <algorithm>

#include <thread>

using namespace std;

const int Cores = 4; // pascal case

// merge function to sort an array

void mergeSort(int\* array\_, int Left, int Right) // snake case, psacal case

{

if (Left < Right)

{

int mid = (Left + Right) / 2;

mergeSort(array\_, Left, mid);

mergeSort(array\_, mid + 1, Right);

inplace\_merge(array\_ + Left, array\_ + mid + 1, array\_ + Right + 1);

}

}

int main()

{

cout << "\t\tSORTING USING MERGE SORT\n\n";

int size;

cout << "Enter array size= ";

cin >> size;

int\* arr = new int[size]; // fixing size of array

cout << "\n";

cout << "Enter the elements (with spaces)= ";

for (int i = 0; i < size; i++)

cin >> arr[i]; // puting elements and incrementing from 0.

int partSize = size / Cores; // dividing array in 4 as cores=4

int start[Cores], end[Cores];

for (int i = 0; i < Cores; i++)

{

start[i] = i \* partSize;

end[i] = start[i] + partSize - 1;

}

end[Cores - 1] = size - 1;

thread threads[Cores]; // array of threads

for (int i = 0; i < Cores; i++) // using for loop for

threads[i] = thread(mergeSort, arr, start[i], end[i]); // sorting each of 4 parts of array using each thread.

for (int i = 0; i < Cores; i++)

threads[i].join();

for (int i = 1; i < Cores; i++)

inplace\_merge(arr, arr + end[i - 1] + 1, arr + end[i] + 1); // merging all 4 parts of array

cout << "Sorted arraya= ";

for (int i = 0; i < size; i++) //printing sorted array

cout << arr[i];

cout << "\n\n\t\tENDED THANKYOU\n\n"; //ended :)

}

**Available Cores in the system:**

**Cores:** 4

**Processors:** 8

**Screenshot:**

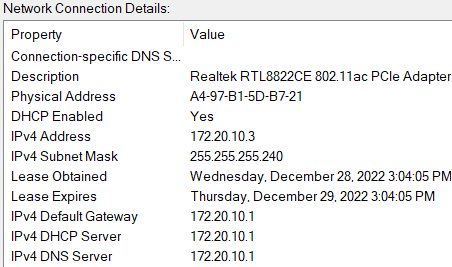
Table

Description automatically generated

**Mac-Address:**

**Address:** A4-97-B1-5D-B7-21

**Screenshot:**



Links:

<https://www.softwaretestinghelp.com/merge-sort/#:~:text=C%2B%2B%20Merge%20Sort%20Technique.,to%20form%20a%20unified%20solution>.

<https://www.freecodecamp.org/news/snake-case-vs-camel-case-vs-pascal-case-vs-kebab-case-whats-the-difference/>

<https://www.geeksforgeeks.org/merge-sort-using-multi-threading/>