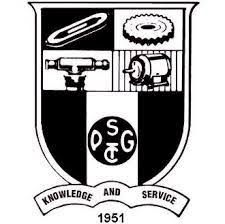
SORTING USING MULTITHREADS AND SUDOKU VALIDATOR

A REPORT ON PACKAGE

SUBJECT: OPERATING SYSTEMS



DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES, PSG COLLEGE OF TECHNOLOGY,COIMBATORE - 641004

SUBMITTED BY

J B BHAVADHARANI - 18PT09

M HARINI - 18PT12

**TABLE OF CONTENTS**

**Abstract**

**1.1 Introduction**

**1.2 Description**

**1.3 System Calls Used**

**1.4 Tools and Technologies**

**1.5 Workflow**

**1.6 Results and Discussion**

**1.7 Conclusion**

**1.8 Bibliography**

**Abstract**

The package aims at understanding and implementing MULTI-THREADS for various purposes. This package consists of two application namely for sorting and sudoku validator. Precisely, we can use this project to understand how parallelism can be achieved using multi-threads and how it helps us to reduce the execution time .

* 1. **Introduction**

when we execute a given program sequentially in a multiprocessing environment we underutilise the resources. Because sequential execution requires only one processor. To maximize the utilisation of processors we opt for parallel processing where the work carried out by the each processors are independent of each other.

We have two scenario where we have implemented this concept:

1. multithreaded merge sort
2. sudoku validator using multithreads

This report will make the reader understand as to how our implementation is efficient compared to the traditional methods.

**1.2 Description**

* what is sorting?

sorting is arrangement of objects in some particular order.a sorting algorithm used to rearrange a given array or list elements according to a comparison operator on the elements. the comparison operator is used to decide the new order of element in the respective data structure.

example:

arr: (before sort) 3,5,2,1,4

arr: (after sort) 1,2,3,4,5

* what is merge sort ?

it is a kind of divide and conquer algorithm. the elements of any data structure are divided into two halves(partition) till it cant be divided further.later each half is sorted and then merged.

* why merge sort?

merge sort is selected because it’s a stable sort . retains element position

eg: 3 1 2 3

after sorting : 1 2 3 3

the time complexity: o(nlogn)

* what are threads?

it is a light weight process . multi-threads used to achieve parallelism. communication between threads is also easier and faster.

eg: in ms –word one thread used to format text and the other is used to get the input.

* rules of sudoku

sudoku is a game of logic and reasoning, number-placement **puzzle.**

* each row, column, and nonet can contain each number (typically 1 to 9) exactly once.
* the sum of all numbers in any nonet, row, or column must match the small number printed in its corner. for traditional sudoku puzzles featuring the numbers 1 to 9, this sum is equal to 45

**1.3 System Calls Used**

Some system calls that were used include:

1. Memcpy() – to copy the data stored in a variable to a particular memory location

**1.4 Tools and Technologies**

1. Ubuntu LTS 18.04LTS Application
2. GCC 7 compiler
3. Required header files were added

**1.5 Workflow**

* **Sorting using multithreads**

**~ main()**

In the main function we input the number of elements to be sorted is by the user . Since its not feasible to input large number of inputs from user we use the random function to randomly generate the numbers in the range of 0 to 9999.

Since the number of elements to be sorted varies we use dynamic memory and is allocated using malloc. we also maintain a duplicate array of the elements to verify and compare the results of the sorted array.

**~merge\_sort()**

It is being invoked from main to abstract other merge function.

**~merge()**

This merge function is used to merge the two counter partitions which have been already sorted .

**~merge\_sort\_mt()**

This function is used to create the threads and initialise its parameters and calls the function merge\_sort\_thread().

**~merge\_sort\_thread()**

The main purpose of this function is to acieve recurrison.so this function again calls the merge\_sort\_mt().

* **sudoku validator**

~**main()**

We have created 27 threads and initialised .we have 9 threads for each row and 9 threads for each column and 9 threads for each smaller 3\*3 nonet . The threads call their respective function during creation.

**~isRowValid()**

This check if the respective row called by the thread is valid i.e by checking if each number between 1 to 9 appears only once. If the given row is valid the result is updated in the global array named valid as 1.

**~isColumnValid()**

This check if the respective column called by the thread is valid i.e by checking if each number between 1 to 9 appears only once. If the given column is valid the result is updated in the global array named valid as 1.

**~is3x3Valid()**

This check if the respective 3x3 nonet called by the thread is valid i.e by checking if each number between 1 to 9 appears only once. If the given 3x3 nonet is valid the result is updated in the global array named valid as 1.

The result is being generated by checking the values in the array valid which has 27 values . it stores the value returned by the 27 threads. if the validity was right 1 is returned or else 0.first 7 position correspond to the threads of the 3x3 nonet , the next 9 to the rows and last 9 for the column. if all the 27 value is 1 the solution is valid or else invalid.

**1.6 Result and Discussion**

**Multi-threading** is way to improve parallelism by running the threads simultaneously in different cores of your processor . Since each partition in the merge sort is independent of each other we can achieve parallelism and also reduce time complexity . we can also use quicksort instead of merge sort . In sudoku validator , when we check for the validity of I number it doesn’t affect the validity of any j number where I and j are not equal.so parallelism is achieved even in this concept.

**1.7 Conclusion**

We have successfully completed this assignment and have gained a deep insight into the usage of using multithreads. We have also understood and demonstrated the various ways of using the multithreads and presenting them in a user friendly way. Therefore, sorting using multithreads and sudoku validator has been successfully demonstrated.

**1.8 Bibliography**

**1.81 Books**

* The UNIX System S.R. Bourne Addison-Wesley, 1983.
* Advanced UNIX Programming M.J. Rochkind Prentice-Hall, 1985.
* Programming Utilities and Libraries SUN Microsystems, 1990.
* Unix man pages (man function).

**1.82 Websites**

* <https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm>
* <https://www.geeksforgeeks.org/introduction-of-system-call/>
* <http://man7.org/linux/man-pages/man3/>
* http://codewiki.wikidot.com/c:system-calls:read
* https://linux-tips.com/t/hard-link-files/125/2
* https://c-for-dummies.com/blog/?p=3004
* www.stackoverflow.com