Netaji Subhas University of Technology



High Performance Computing Lab File

Submitted By:

Name:- Sreeram Vipparla

Roll No:- 2021UCA1860

Batch :- CSAI-1

INDEX

|  |  |  |
| --- | --- | --- |
| S.NO | PRACTICAL | PAGE NO |
| 1) | Write a program in C to multiply two matrices of size 10000 x 10000 each and find it's execution-time using "time" command. |  |
| 2) | Write a parallel program to print “Hello World" using MPI. |  |
| 3) | Write a parallel program to find sum of an array using MPI. |  |
| 4) | Write a C program for parallel implementation of Matrix Multiplication using MPI. |  |
| 5) | Write a C program to implement the Quick Sort Algorithm using MPI. |  |
| 6) | Write a multithreaded program to generate Fibonacci series using pThreads. |  |
| 7) | Write a program to implement Process Synchronization by mutex locks using pThreads. |  |
| 8) | Write a "Hello World" program using OpenMP library also display number of threads created during execution. |  |
| 9) | Write a C program to demonstrate multitask using OpenMP. |  |
| 10) | Write a parallel program to calculate the value of PI/Area of Circle using OpenMP library. |  |
| 11) | Write a C program to demonstrate default, static and dynamic loop scheduling using OpenMP. |  |
|  |  |  |

MPI

MPI, which stands for Message Passing Interface, is a theory for parallel programming. It focuses on dividing a large computational problem into smaller, independent tasks. These tasks are then distributed among multiple processors or computers in a network, allowing them to work on the problem simultaneously.

The core idea behind MPI is communication. Each processor, called a rank in MPI terminology, has its own local memory and can only directly access its own data. To collaborate, ranks need to exchange data by sending and receiving messages. MPI provides a set of functions that define how these messages are structured, sent, received, and synchronized. This allows ranks to work on their assigned tasks, share necessary data with others, and ultimately combine their results to solve the overall problem.

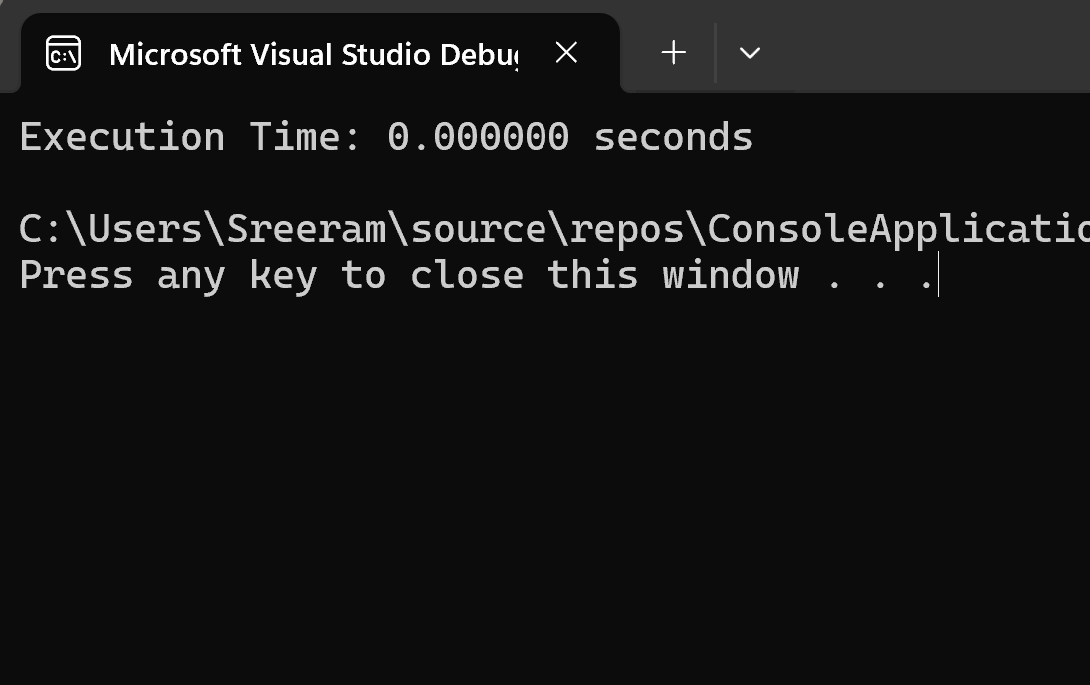
MPI offers several advantages. Firstly, it enables significant speedups by utilizing the combined processing power of multiple machines. Secondly, it promotes modularity by breaking down complex problems into smaller, easier-to-manage tasks. Finally, MPI is portable, meaning programs written with MPI can be run on various computer architectures with minimal changes to the code. This makes it a widely adopted standard for parallel programming in scientific computing and high-performance applications.

# **PRACTICAL 1**

Write a program in C to multiply two matrices of size 10000 x 10000 each and find it's execution-time using "time" command. Try to run this program on two or more machines having different configurations and compare execution-times obtained in each run. Comment on which factors affect the performance of the program



# Result

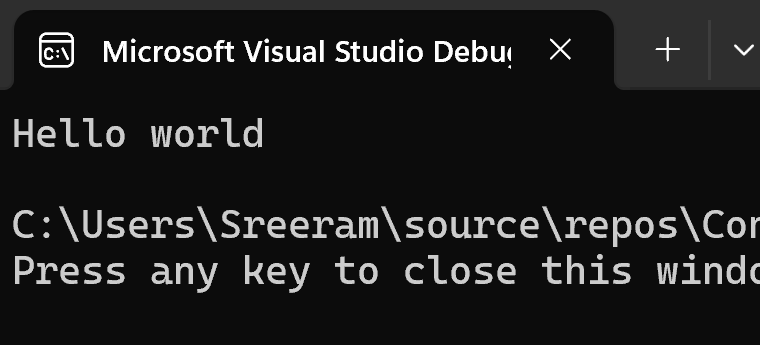


# **PRACTICAL 2**

Write a parallel program to print “Hello World" using MPI.



# Result



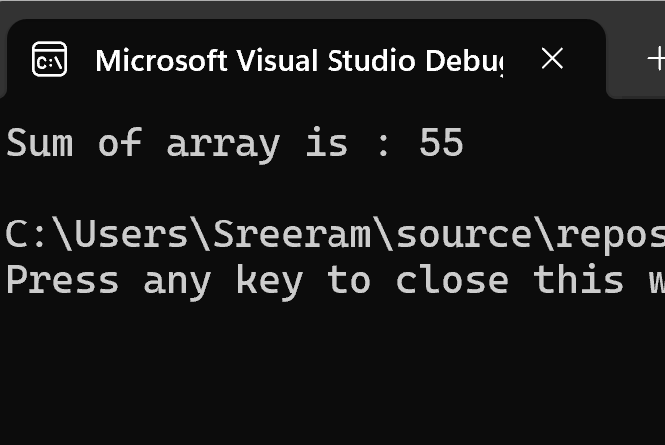
# **PRACTICAL 3**

Write a parallel program to find sum of an array using MPI



# 

# Result



# **PRACTICAL 4**

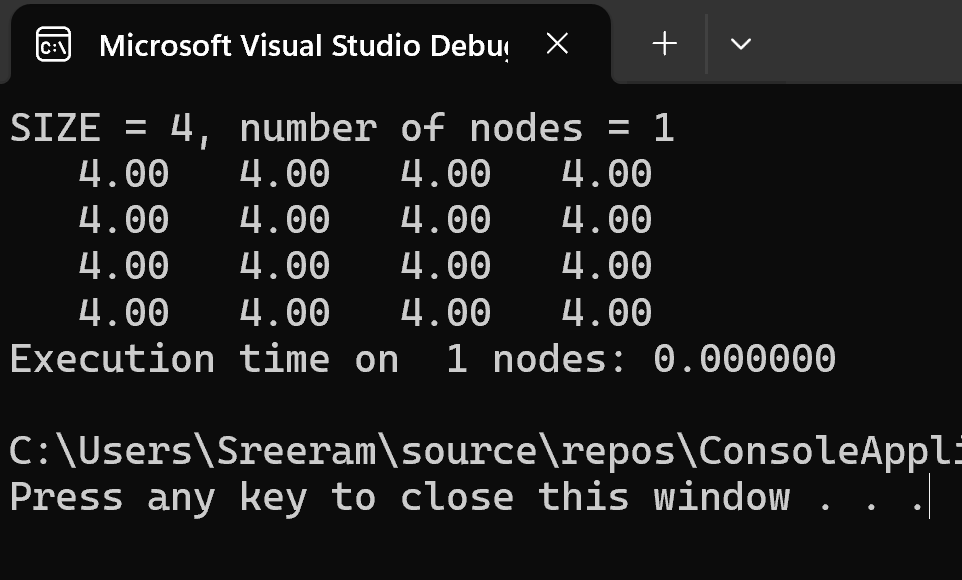
Write a C program for parallel implementation of Matrix Multiplication  
using MPI.



# 

# 

# Result

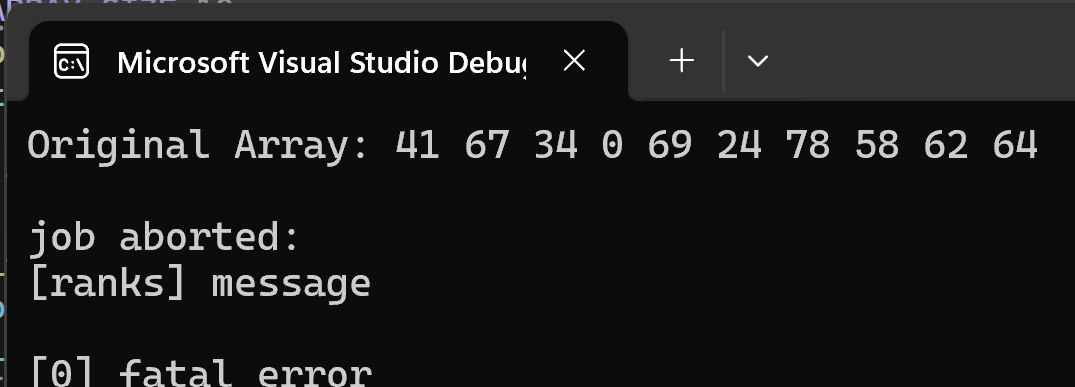


# **PRACTICAL 5**

Write a C program to implement the Quick Sort Algorithm using MPI**.**



# Result



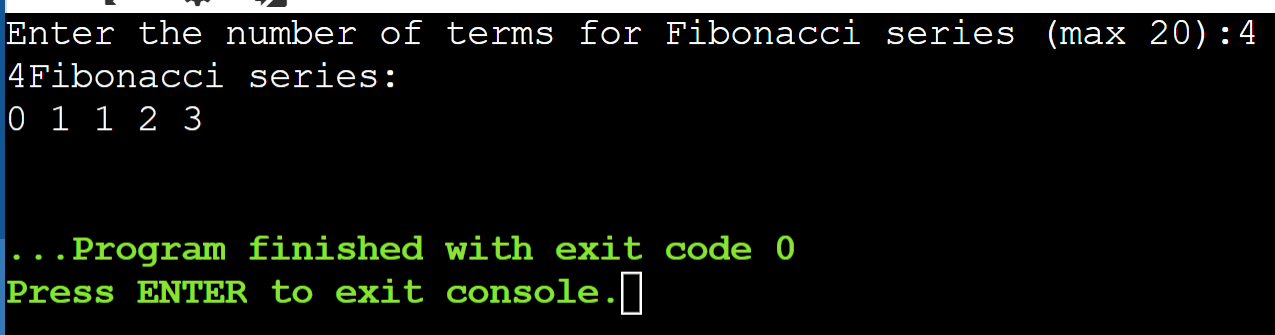
# **PRACTICAL 6**

Write a multithreaded program to generate Fibonacci series using  
pThreads.



# 

# Result

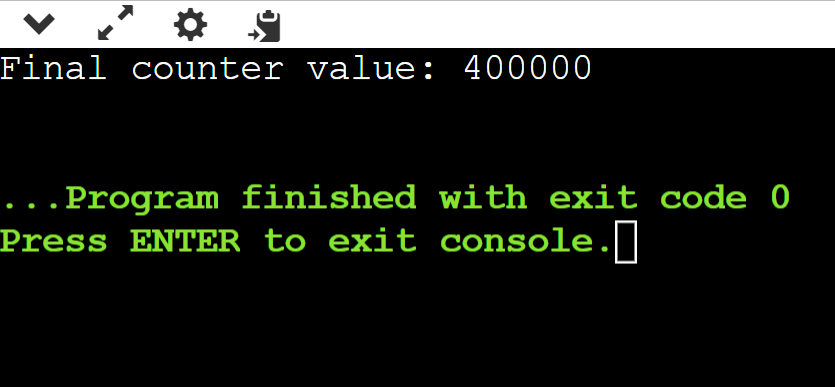


# **PRACTICAL 7**

Write a program to implement Process Synchronization by mutex locks using pThreads.



# Result

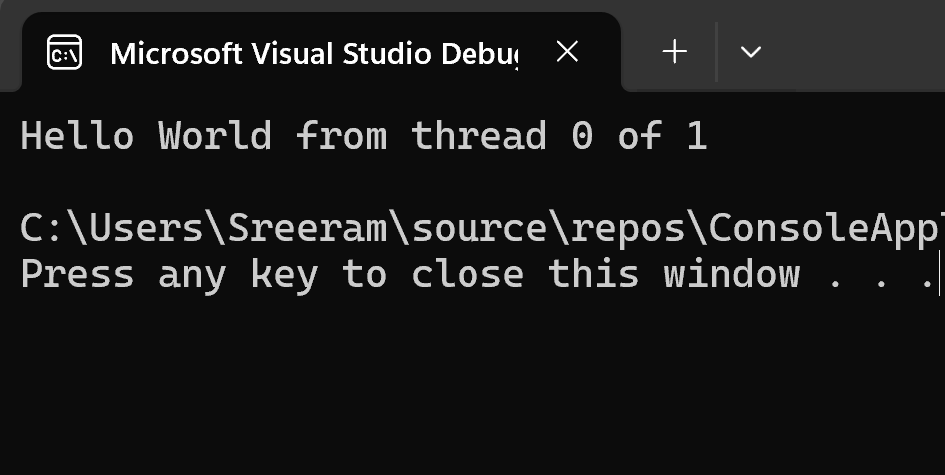


# **PRACTICAL 8**

Write a "Hello World" program using OpenMP library also display number of threads created during execution.



# Result

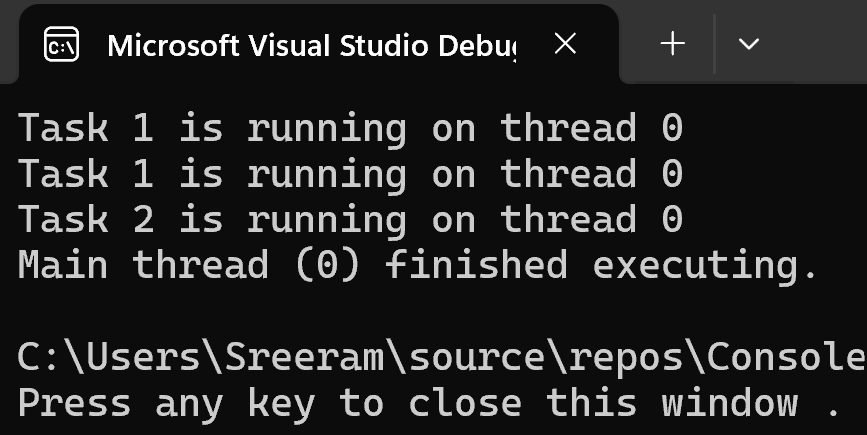


# **PRACTICAL 9**

Write a C program to demonstrate multitask using OpenMP.



# Result

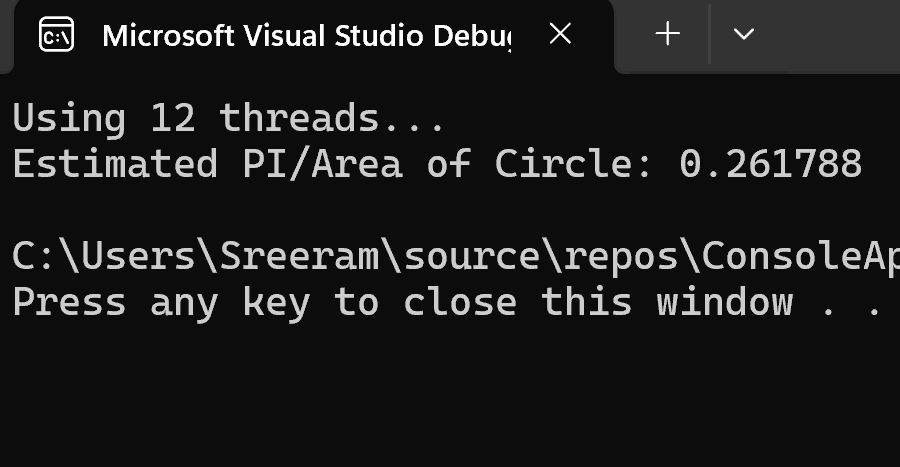


# **PRACTICAL 10**

Write a parallel program to calculate the value of PI/Area of Circle using OpenMP library.



# Result



# **PRACTICAL 11**

Write a C program to demonstrate default, static and dynamic loop  
scheduling using OpenMP.



# Result