**Thread Programming**

**Author : SREENIDHI KRISHNA**

**Date : 10/16/2014**

**SUMMARY**

This simple program written in C language for UNIX/LINUX systems takes in an integer N and number of threads as user input and verifies whether the sum of factors of N, excluding N adds up to the number.

**PROGRAM MODULES**

**1. void \* Factorize(void \*arg)**

This Function is called by each thread to evaluate the factors of the number and add to the sum *mysum* variable. It uses a mutex *mutexsum* for the shared data i.e. mysum. The function adds all factors of the number (user input) and returns to main function.

**2.** **int main(int argc,char \*\*arv)**

The main function takes an integer value N and number of threads from the user. It makes sure that the number of threads aren’t excess/insufficient and allows a maximum of square root of N threads, where N is the number to be checked. The main function calls the function Factorize to calculate the sum of factors. It then checks whether the given number is perfect or not by comparing the *mysum* with the number itself, and prints the result. The main function creates joinable threads and makes use of the system call pthread\_join() , to ensure thread synchronization.

**FOR COMPILATION & EXECUTION**

Type

**$ gcc -pthread ./compute\_perfect\_num.c -lm**

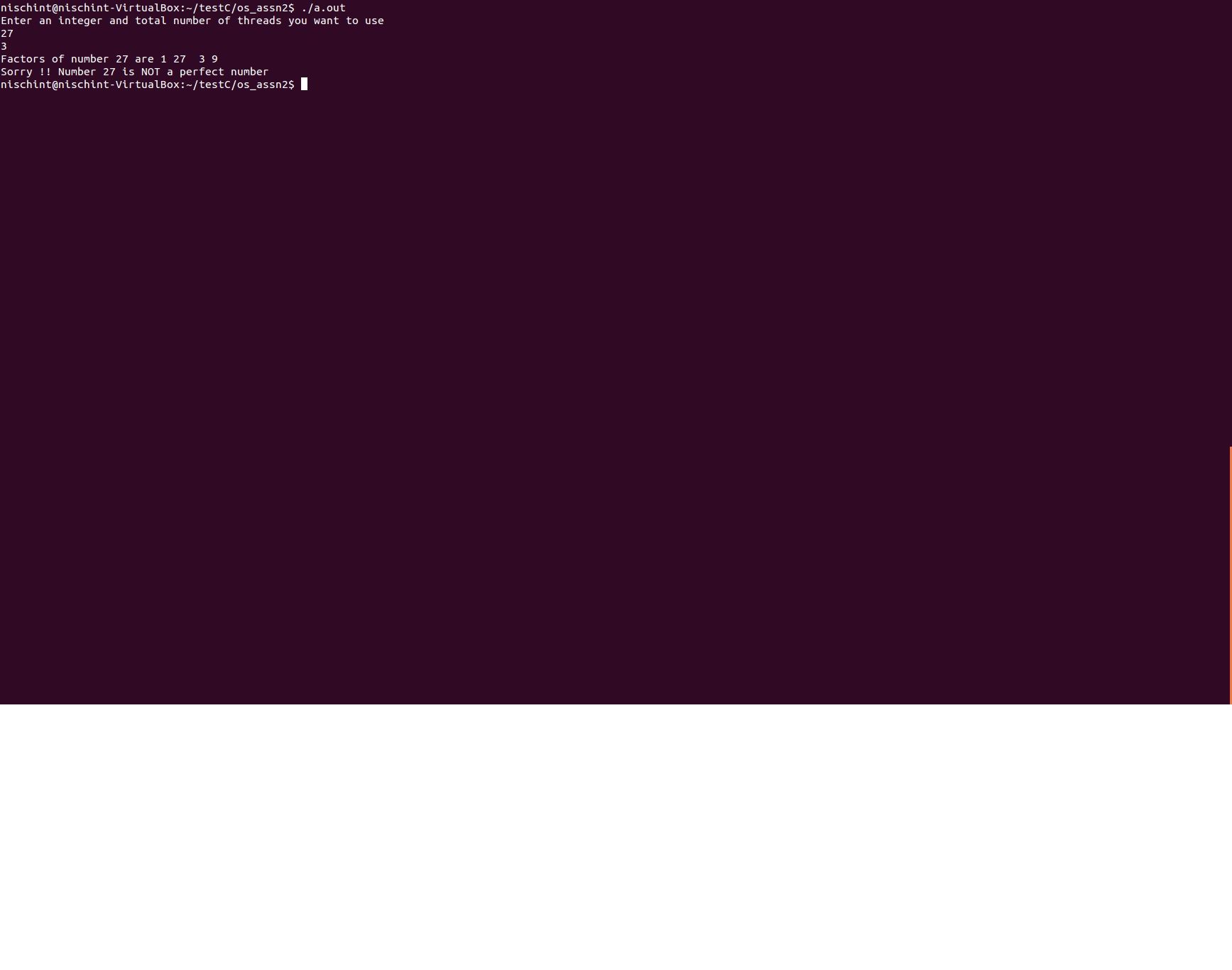
**$ ./a.out**

**SAMPLE OUTPUT**

**1.The output for the following input is as shown.**

**Number : 27**

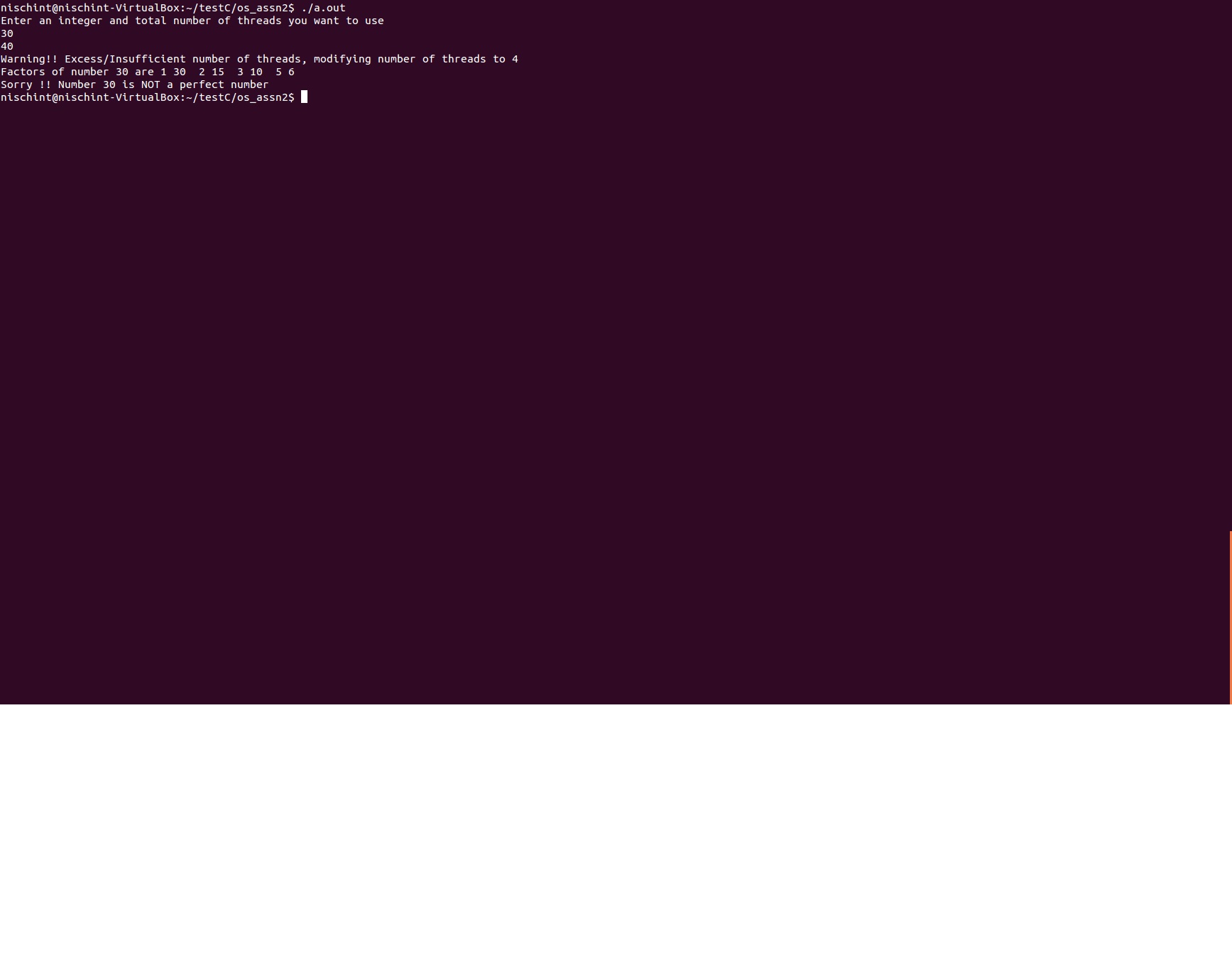
**Threads : 3**

****

**2.The output for the following input is as shown.**

**Number : 30**

**Threads : 40**

****

**3.The output for the following input is as shown.**

**Number : -5**

**Threads : 2**

