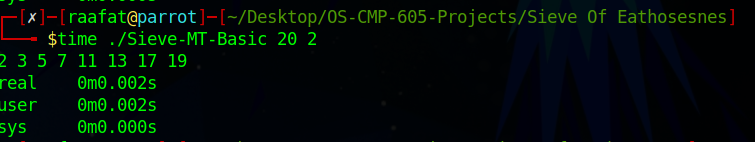
# Basic Multithreading

We run the program as ./Sieve-MT-Basic N thread\_count

N=20, Thread Count = 2



N=20, Thread Count = 4

Text

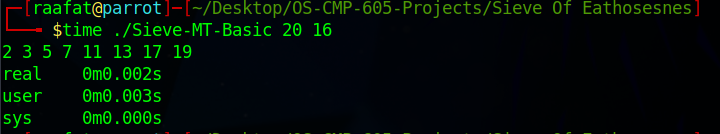
Description automatically generated with medium confidence

N=20, Thread Count = 8

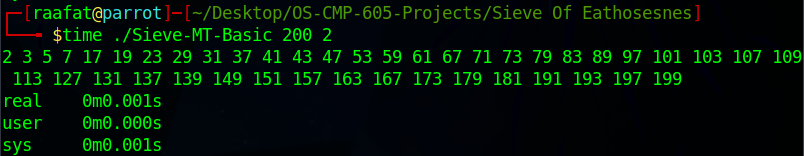
Graphical user interface

Description automatically generated

N=20, Thread Count = 16



N=200, Thread Count = 2

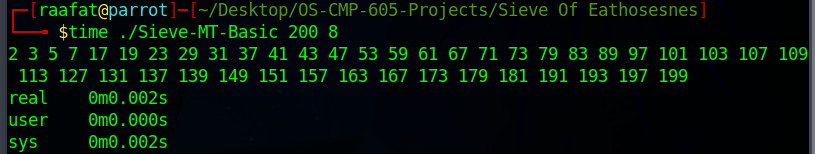


N=200, Thread Count = 4

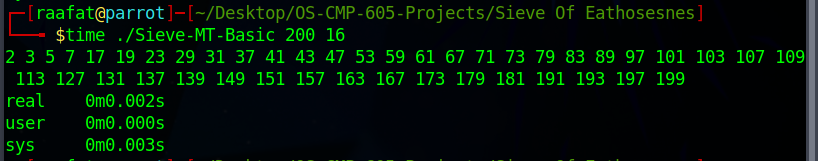
Text

Description automatically generated

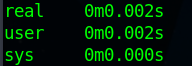
N=200, Thread Count = 8



N=200, Thread Count = 16



N=2000, Thread Count = 2

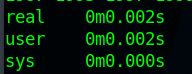


N=2000, Thread Count = 4

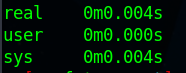
A picture containing text, device, meter

Description automatically generated

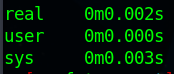
N=2000, Thread Count = 8



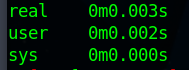
N=2000, Thread Count = 16



N=20000, Thread Count = 2



N=20000, Thread Count = 4



N=20000, Thread Count = 8

Graphical user interface, text

Description automatically generated

N=20000, Thread Count = 16

A picture containing text, device, meter

Description automatically generated

## Discussion

In Basic Multithreading, we get an increased overhead of thread initialization and context switching when number of threads is big (8,16) and the numbers are small (20,200)

# Advanced Multithreading

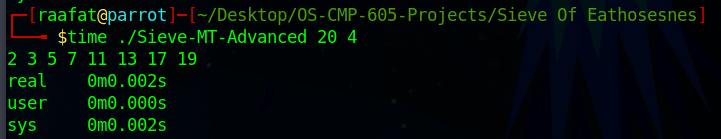
We run the program as ./Sieve-MT-Advanced N thread\_count

N=20, Thread Count = 2

Text

Description automatically generated with medium confidence

N=20, Thread Count = 4



N=20, Thread Count = 8

Graphical user interface

Description automatically generated with medium confidence

N=20, Thread Count = 16

A screenshot of a computer

Description automatically generated with medium confidence

N=200, Thread Count = 2

Text

Description automatically generated

N=200, Thread Count = 4

Graphical user interface

Description automatically generated with medium confidence

N=200, Thread Count = 8

Graphical user interface, text

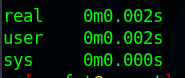
Description automatically generated

N=200, Thread Count = 16

Graphical user interface

Description automatically generated

N=2000, Thread Count = 2

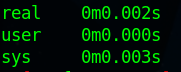


N=2000, Thread Count = 4

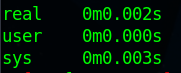
A picture containing text, clock, device, meter

Description automatically generated

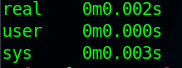
N=2000, Thread Count = 8



N=2000, Thread Count = 16



N=20000, Thread Count = 2



N=20000, Thread Count = 4

Graphical user interface, text

Description automatically generated

N=20000, Thread Count = 8

Graphical user interface, text

Description automatically generated

N=20000, Thread Count = 16

A picture containing text, clock, device, meter

Description automatically generated

## Discussion

In Advanced Multithreading almost all test cases have the same conclusion. From 4 threads onwards it doesn’t make any appreciable difference. Most of the time is spent by the system handling the threads.

# Multiprogramming

We run the program as ./Sieve-MP N thread\_count

N=20, Thread Count = 2

A picture containing graphical user interface

Description automatically generated

N=20, Thread Count = 4

A picture containing graphical user interface

Description automatically generated

N=20, Thread Count = 8

A picture containing graphical user interface

Description automatically generated

N=20, Thread Count = 16

A picture containing text

Description automatically generated

N=200, Thread Count = 2

Text

Description automatically generated

N=200, Thread Count = 4

Graphical user interface, text

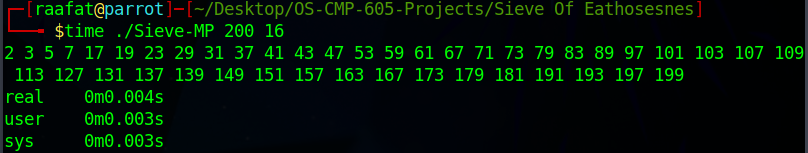
Description automatically generated

N=200, Thread Count = 8

A screenshot of a computer

Description automatically generated with medium confidence

N=200, Thread Count = 16



N=2000, Thread Count = 2

Text

Description automatically generated with medium confidence

N=2000, Thread Count = 4

A picture containing text, device, meter

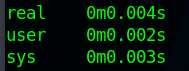
Description automatically generated

N=2000, Thread Count = 8

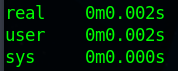
A picture containing text, device, meter

Description automatically generated

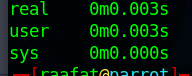
N=2000, Thread Count = 16



N=20000, Thread Count = 2



N=20000, Thread Count = 4



N=20000, Thread Count = 8

A picture containing text, device, meter

Description automatically generated

N=20000, Thread Count = 16

A picture containing text, clock, device, meter

Description automatically generated

## Discussion

In Multiprogramming due to the heavy nature of forking processes. In all but the test case of 20000 showed slower performance due to the management of forked processes in cases of 8, and 16 processes and 4 processes when dealing with low numbers like 20.