Comp 8005 A1: Documentation

By Ian Lee

Design Work

Objective: compare the efficiency of processes versus threads using a computationally intensive task.
Computational task: Prime Factor Decomposition
Method: Brute force. Only figure out the prime factors of a number, not multiples of one
ex: input: 8result: 2
instead of
input:8 result: 2 2 2

Language:

cpp

Usage:

ipfd -[pt] number

command line arguments:

number	number to be decomposed
-p	use processes
-t	use threads

Pseudo Code

```
Main
    • parse command line arguments
    • prepare ranges for computational work
    • create files for threads/processes to put results and for timing results
    • save beforetime
    • create threads/processes
    • wait for threads/processes
    • save aftertime
    • save beforetime - aftertime to file

Work function that both processes and threads call
    • for each number in range
        ○ if this number is a factor of input number
            ■ check if this number is a prime number
    • save results to file
```

Test Results

Tested on a Lab 323 computer

primes.txt

```
12312121    67 183763
12312121    67 183763
421531642    2 7 1361 22123
421531642    2 7 1361 22123
791645159
791645159
1212121212    2 3 41 271 9091
1212121212    2 3 41 271 9091
3254325432    2 3 739 183487
3254325432    2 3 739 183487
65656565652    2 3 5471380471
65656565652    2 3 5471380471
```

process.txt

12312121	1s -954326usec	=	**0.045674 s**
421531642	1s 149517usec	=	**1.149517 s**
791645159	2s 35434usec	=	**2.035434 s**
1212121212	4s -926870usec	=	**3.073130 s**
3254325432	8s 398216usec	=	**8.398216 s**
65656565652	213s -699666usec	=	212.300334 s

thread.txt

12312121	0s 47104usec	=	0.047104 s
421531642	1s 111079usec	=	1.111079 s
791645159	2s 65758usec	=	2.065758 s
1212121212	3s 116660usec	=	3.116660 s
3254325432	9s -311727usec	=	8.688273 s
65656565652	210s -19075usec	=	**209.980925 s**

From the data above, most tests showed that the processes took less time than the Posix threads. However, the benefits seem marginal (5-10 %) and on the largest duration test number, the threads were actually faster by approximately 2 seconds (1%). This was potentially caused by a program running in the background during the process test and not for the thread test, but I was not monitoring for such things at the time.

More tests that provide an average time for each number to be tested should be performed so that the data can be verified.