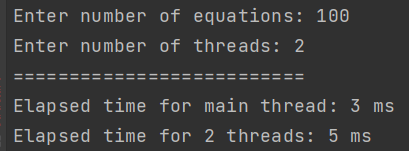
**Висновки по ЛР №10**

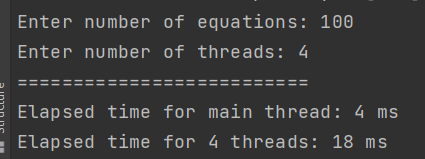
У цій лабораторній роботі, а саме роботі №10 я реалізував алгоритм Гаусса на мові програмування С++. Моя програма показала доволі хороші результати, навіть попри те, що цей алгоритм доволі погано паралелиться. Як і завжди, у звіті я розгляну по три випадки (з малою, середньою і великою кількістю потоків) для трьох (мала, середня і велика кількість систем лінійних рівнянь) різних за розміром систем рівнянь. Отже, перейдемо ближче до кожного випадку:

1. Мала кількість рівннянь (100) і мала кількість потоків(2). У цьому випадку алгоритм відпрацьовує швидко, як в головному потоці так і в мультитредінгу. Зумовлено це тим, що на вхід подається мала кількість рівнянь і мала кількість потоків, через що майже не витрачається час на їх створення. Різниця тут між часом роботи мейн потоку і мультитредів несуттєва.

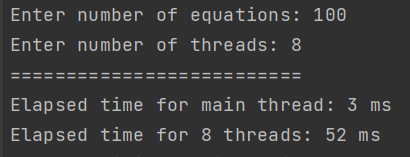


1. Мала кількість рівнянь (100) і середня кількість потоків (4).

Результат виконання мого алгоритму нижче на картинці. Тут головний потік, як і очікувалось, обраховує швидше за мультитреди, оскільки введено малу кількість рівнянь і в мультритредінгу більше часу йде на створення самих потоків, аніж на обчислення в них. Різниця в часі суттєва.

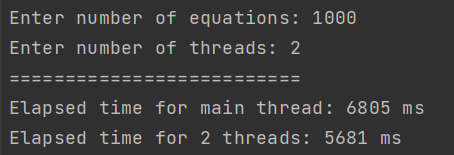


1. Мала кількість рівнянь (100) і велика кількість потоків (8).

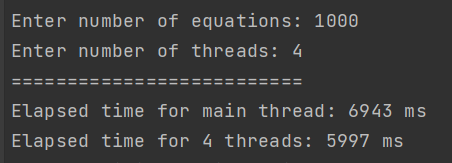
Результат гірший для мультитредів ніж у попередніх двох випадків, що і очікувалось. Причина та ж – на створення потоків у мультитредінгу у порівнянні з самими обчисленнями в них йде ну дуже багато часу. Результат на скріні. Різниця в часі роботи більше ніж 10 разів.

Перейдемо до середньої кількості систем лінійних рівнянь.

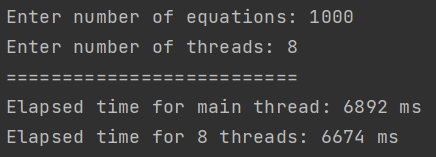
1. Середня кількість рівнянь (1000) і мала кількість потоків (2). У цьому випадку головний потік вже не може так швидко обрахувати ту кількість даних, яку ми йому вводимо, як це було у випадку з малою кількістю систем лінійних рівнянь. Тут мультитреди показують себе не сильно краще, але все ж краще, оскільки, як я вже казав на початку висновків, цей алгоритм доволі таки погано паралелиться.



1. Середня кількість рівнянь (1000) і середня кількість потоків (4). Ситуація трохи схожа на попередню, навіть час виконання в потоках майже такий самий як і на попередньому скріні, але трохи більший, оскільки пішов додатковий час на створення самих потоків. Проте, навіть в такому випадку, алгоритм в мультитредах працює швидше, ніж в одиночному(головному) потоці. Результат, в принципі, непоганий.

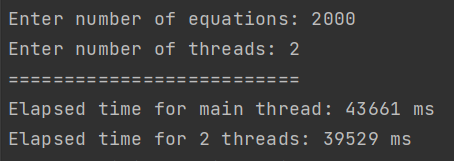


1. Середня кількість рівнянь (1000) і велика кількість потоків (8). Ситуація, в принципі, очікувана. Час у мультитредінгу майже зрівнявся з часом в головному потоці, проте все ще працює швидше за нього.

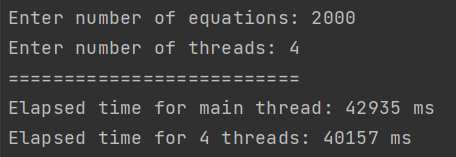


Далі перейдемо до великої кількості систем лінійних рівнянь.

1. Велика кількість систем (2000) і мала кількість потоків (2). Час виконання як і в головному потоці, так і в мультитредінгу сильно збільшився в порівнянні з 1000 систем. Час виконання у мультитредінгу, навіть попри те, що алгоритм погано паралелиться, все одно кращий ніж час виконання моєї програми у головному потоці



1. Велика кількість систем (2000) і середня кількість потоків (4). В принципі, результат сильно схожий до результату на попередньому скріні. Мультитредам знадобилось трохи більше часу на створення самих себе. Більше сказати мені у цьому випадку, в принципі, нічого. Результат виконання програми:



1. Ну і останній, 9 варіант, це велика кількість систем (2000) і велика кількість потоків (8). Результат виконання в цьому випадку, як я і думав, вийшов схожий на два попередніх. Виходить, приблизно на 0.2 сек більше затратилось для мультитредінгу. Гадаю, це час для створення додаткових тредів. Результат виконання для мультитредів все ще кращий, ніж результат виконання у головному потоці.

