Лабораторная работа №1

1. **информация о студенте**

Поляков Игорь Андреевич МОАИС 184

1. **Цель работы**

Получить общие представления о потоках, о способах их создания и использования.

1. **информация о компьютере**

i3-2120 CPU 2 ядра 3.30GHz

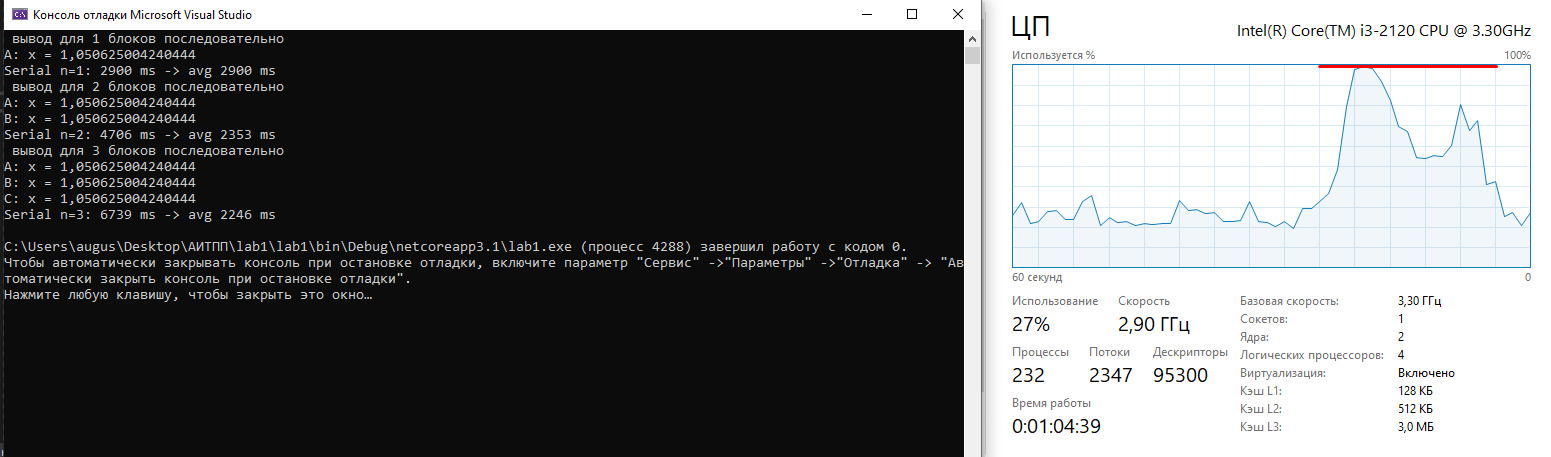
1. **Таблица 1. Сравнение времени последовательных и параллельных вычислений.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Организация алгоритма | Последовательная обработка | | | Параллельная обработка | | |
| общее время работы | время работы одного метода | порядок  вывода | общее время работы | время работы одного метода | порядок  вывода |
| *(1)* | *(2)* | *(3)* | *(4)* | *(5)* | *(6)* | *(7)* |
| 1 блок | 2678 ms | 2678ms | A | 2434 ms | 2343 ms | A |
| 2 блока | 4439 ms | 2219 ms | A, B | 2303 ms | 1151 ms | B, A |
| 3 блока | 6450 ms | 1450 ms | A, B, C | 2445 ms | 815 ms | B, A, C |

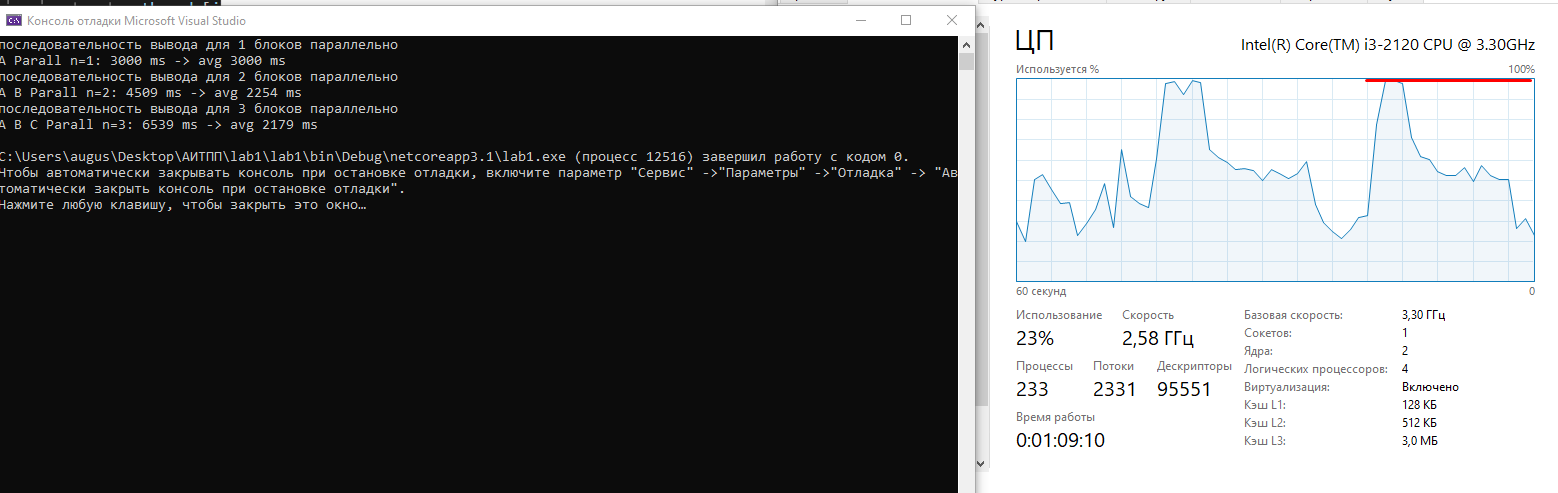
1. **Таблица 2. Влияние приоритета потока на время вычислений.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Число блоков | Среднее время работы одного экземпляра метода | | | | |
| Lowest | BelowNormal | Normal | AboveNormal | Highest |
| *(1)* | *(2)* | *(3)* | *(4)* | *(5)* | *(6)* |
| 1 блок | 3865 ms | 2343 ms | 2199 ms | 2138 ms | 2130 ms |
| 2 блока | 2496 ms | 2287 ms | 2299 ms | 2240 ms | 2239 ms |
| 3 блока | 2258 ms | 2274 ms | 2216 ms | 2143 ms | 2165 ms |

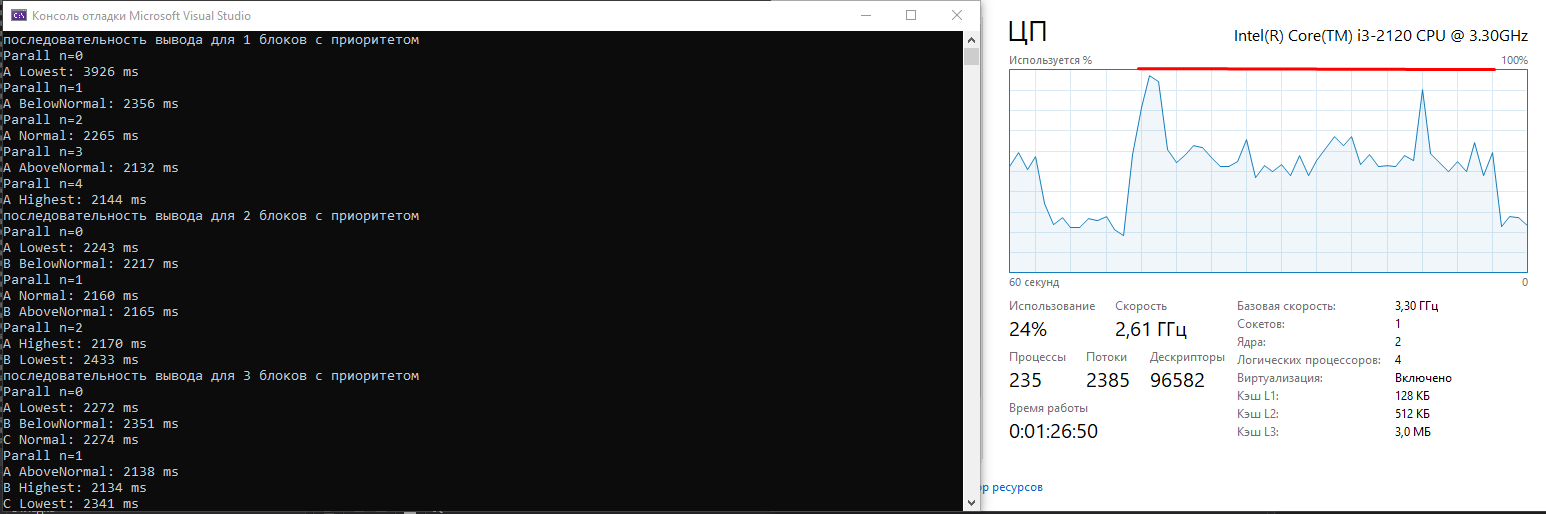
1. Снижается скорость вычислений при превышении количества потоков логических процессоров
2. Потоки с высшим приоритетом выполняются быстрее
3. Нагрузка при последовательном вычислении



При параллельном вычислении



При параллельном вычислении с приоритетом



**using** **System**;

**using** **System.Diagnostics**;

**using** **System.Threading**;

**namespace** **lab1**

{

**class** **Program**

{

**static** **void** **Work**(**int** n) {//функция для последовательного вычисления

**double** x = **1.0**;

**for**(**int** i=**0**;i<**5000**;++i)

**for**(**int** j = **0**; j < **5000**; ++j)

{

x = Math.Sqrt(x);

x += **0.000000001**;

x = Math.Pow(x, **2**);

}

Console.WriteLine($"{(char)n}: x = {x}");

}

**static** **void** **Work1**(**object** item){//функция для потоков

**double** x = **1.0**;

**for** (**int** i = **0**; i < **5000**; ++i)

**for** (**int** j = **0**; j < **5000**; ++j)

{

x = Math.Sqrt(x);

x += **0.000000001**;

x = Math.Pow(x, **2**);

}

Console.Write($"{Convert.ToChar(item)} ");

}

**static** **void** **parallelCalc**(**int** n) {

Thread[] threads = **new** Thread[n];

**for** (**int** j = **0**; j < n; ++j)//назначение функции потокам

threads[j] = **new** Thread(Work1);

**for** (**int** j = **0**; j < threads.Length; ++j){

threads[j].Start(**65** + j);//65 код символа

threads[j].Join();

}

//foreach (var thread in threads)

// thread.Join();

}

**static** **void** **parallelCalcPriority**(**int** n){//функция для вычисления с приоритетом

Stopwatch sw;

Thread[] threads = **new** Thread[n];

**int** priority = **0**;

**for** (**int** i = **0**; i <= **4**/n; ++i){//цикл приоритетов Priority enum {0-4}

Console.WriteLine($"Parall n={i}");

**for** (**int** j = **0**; j < threads.Length; ++j){//по всем элементам массива потоков

sw = Stopwatch.StartNew();

threads[j] = **new** Thread(Work1);

threads[j].Priority = (ThreadPriority)priority;

threads[j].Start(**65** + j);//65 код символа для имени потока

threads[j].Join();

sw.Stop();

Console.WriteLine($"{(ThreadPriority)priority++}: {sw.ElapsedMilliseconds} ms");//вывод с изменением приоритета

priority %= **5**;//значение приоритета не больше 4

}

}

}

**static** **void** **Main**(**string**[] args){

Stopwatch sw;

**const** **int** N = **3**;//N+1 где N количество ядер

//последовательно вычисление

**for** (**int** i = **1**; i <= N; ++i){

Console.WriteLine($" вывод для {i} блоков последовательно");

sw = Stopwatch.StartNew();

**for** (**int** j = **0**; j < i; ++j)

Work(**65** + j);

sw.Stop();

Console.WriteLine($"Serial n={i}: {sw.ElapsedMilliseconds} ms -> avg {sw.ElapsedMilliseconds / i} ms");

}

//параллельное вычисление

**for** (**int** i = **1**; i <= N; ++i){

Console.WriteLine($"последовательность вывода для {i} блоков параллельно");

sw = Stopwatch.StartNew();

parallelCalc(i);

sw.Stop();

Console.WriteLine($"Parall n={i}: {sw.ElapsedMilliseconds} ms -> avg {sw.ElapsedMilliseconds / i} ms");

}

////параллельное вычисление с приоритетами

**for** (**int** i = **1**; i <= N; ++i){

Console.WriteLine($"последовательность вывода для {i} блоков с приоритетом");

parallelCalcPriority(i);

}

}

}

}