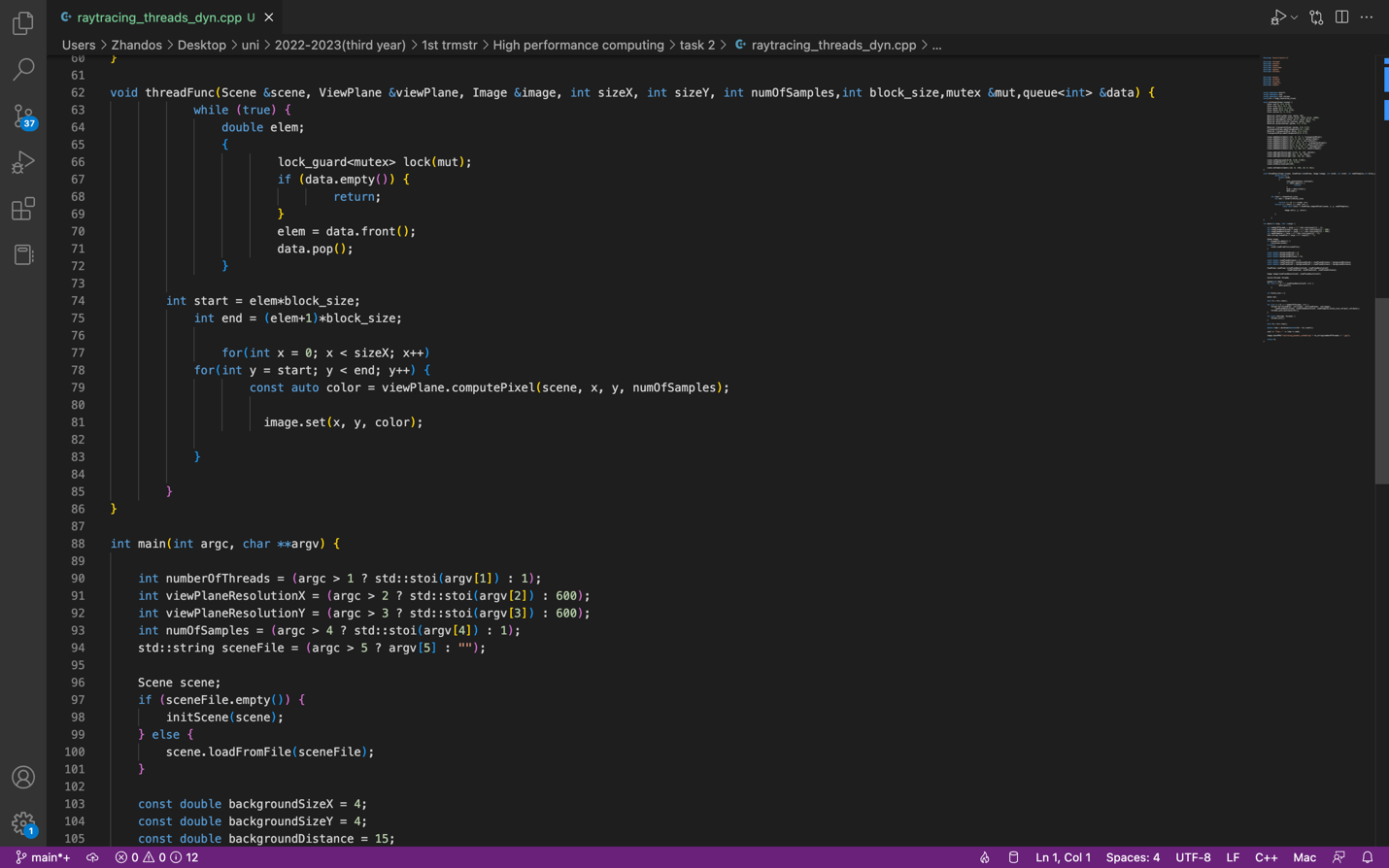
Report on task 2

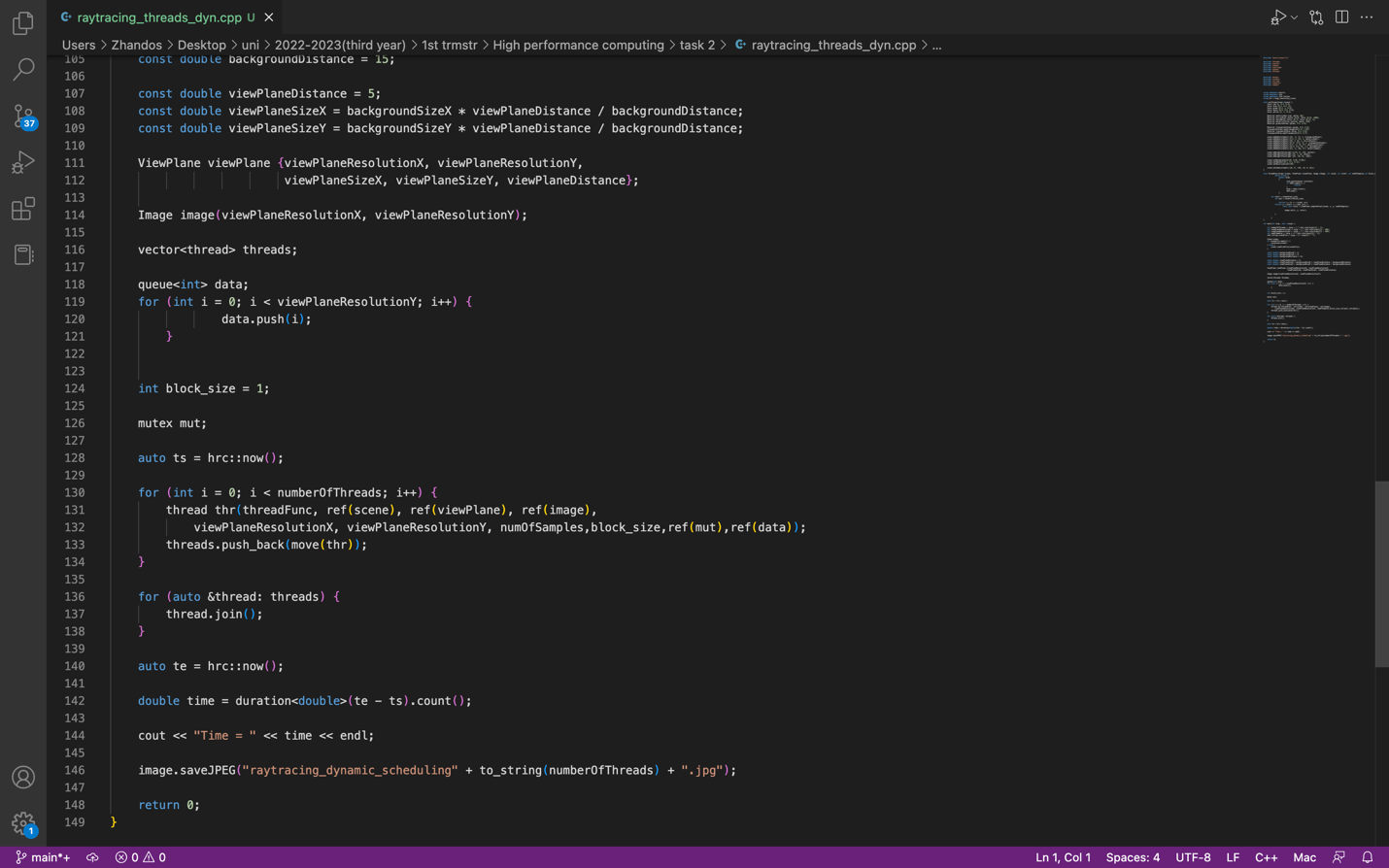
**Name: Abdireshov Zhandos**

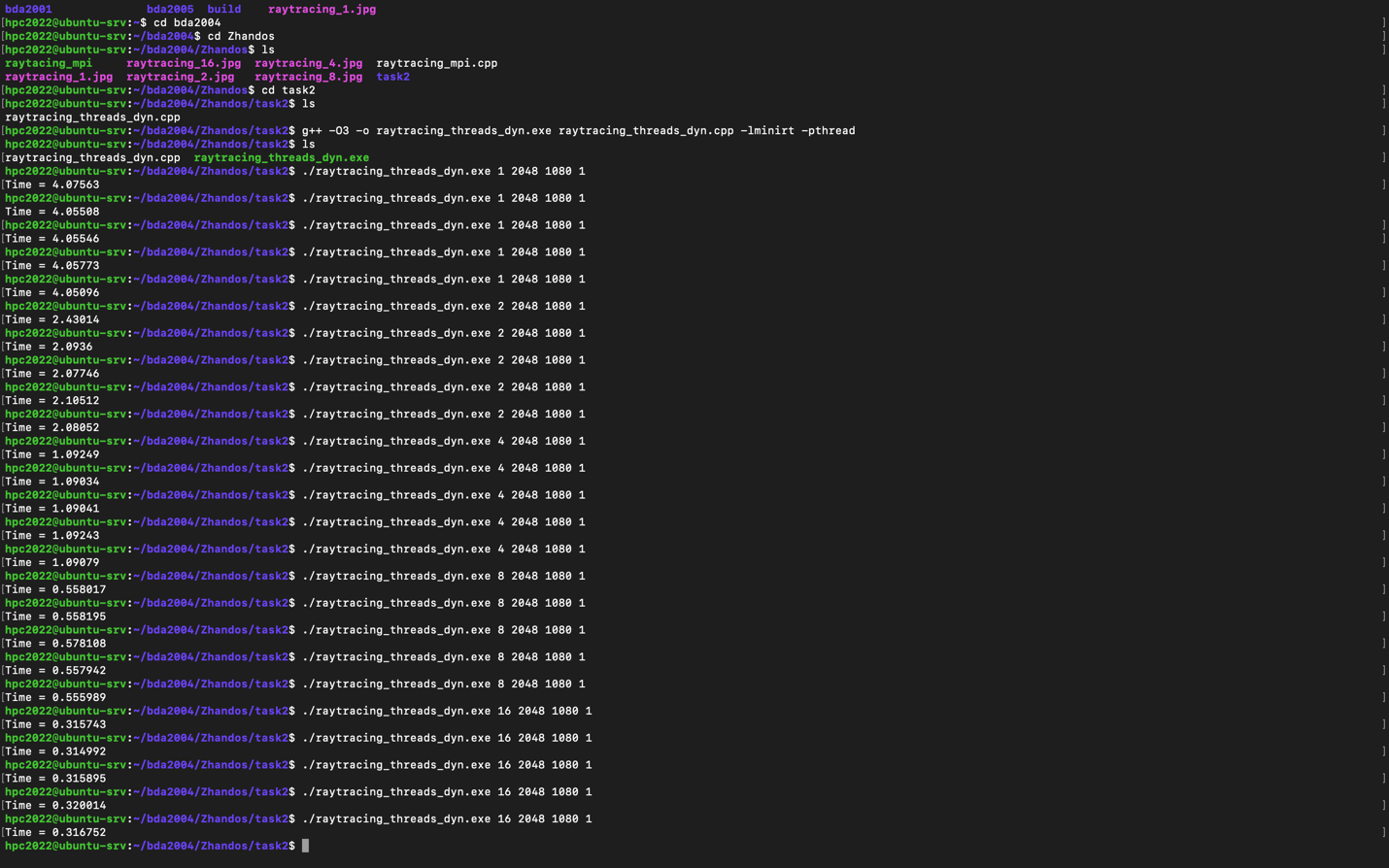
**Group: bd-2004**

**E-mail: 201045@astanait.edu.kz**

Step 4\*: Implement dynamic scheduling with POSIX/C++ threads







|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threads | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | Min. time |
| 1 | 4.07563 | 4.05508 | 4.05546 | 4.05773 | 4.05096 | 4.05096 |
| 2 | 2.43014 | 2.0936 | 2.07746 | 2.10512 | 2.08052 | 2.07746 |
| 4 | 1.09249 | 1.09034 | 1.09041 | 1.09243 | 1.09079 | 1.09034 |
| 8 | 0.558017 | 0.558195 | 0.578108 | 0.557942 | 0.555989 | 0.555989 |
| 16 | 0.315743 | 0.314992 | 0.315895 | 0.320014 | 0.316752 | 0.314992 |

The execution time (to demonstrate how it depends on the number of threads)

Speedup: Speedup(N) = Time(1) / Time(N), N - number of threads

Efficiency: Efficiency(N) = Speedup(N) / N

|  |  |  |  |
| --- | --- | --- | --- |
| Threads | Time | Speedup | Efficiency |
| 1 | 4.05096 | 1 | 1 |
| 2 | 2.07746 | 1.94995 | 0.974975 |
| 4 | 1.09034 | 3.715318 | 0.928829 |
| 8 | 0.555989 | 7.28604 | 0.910755 |
| 16 | 0.314992 | 12.86051 | 0.803782 |

Commit and push your changes to the Gitlab server

Conclusion in a free form

Static separation of threads works sequent (in order), and dynamic separation is taken for the next block at the end of its own. It means that there are threads take the work that is in the queue and take it from there as they perform their own. Meanwhile in the static, the corresponding queue has already been defined for each threads.