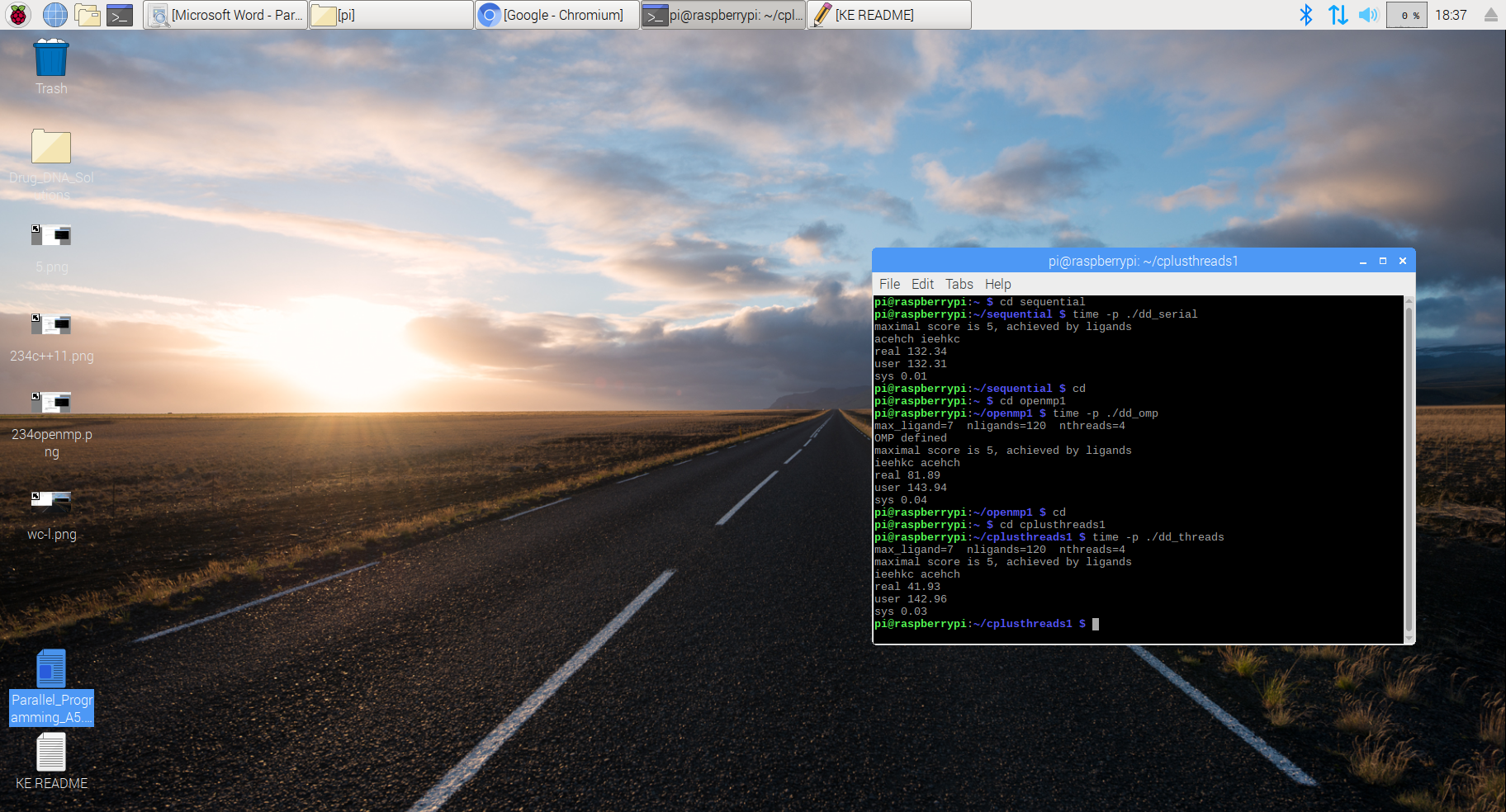
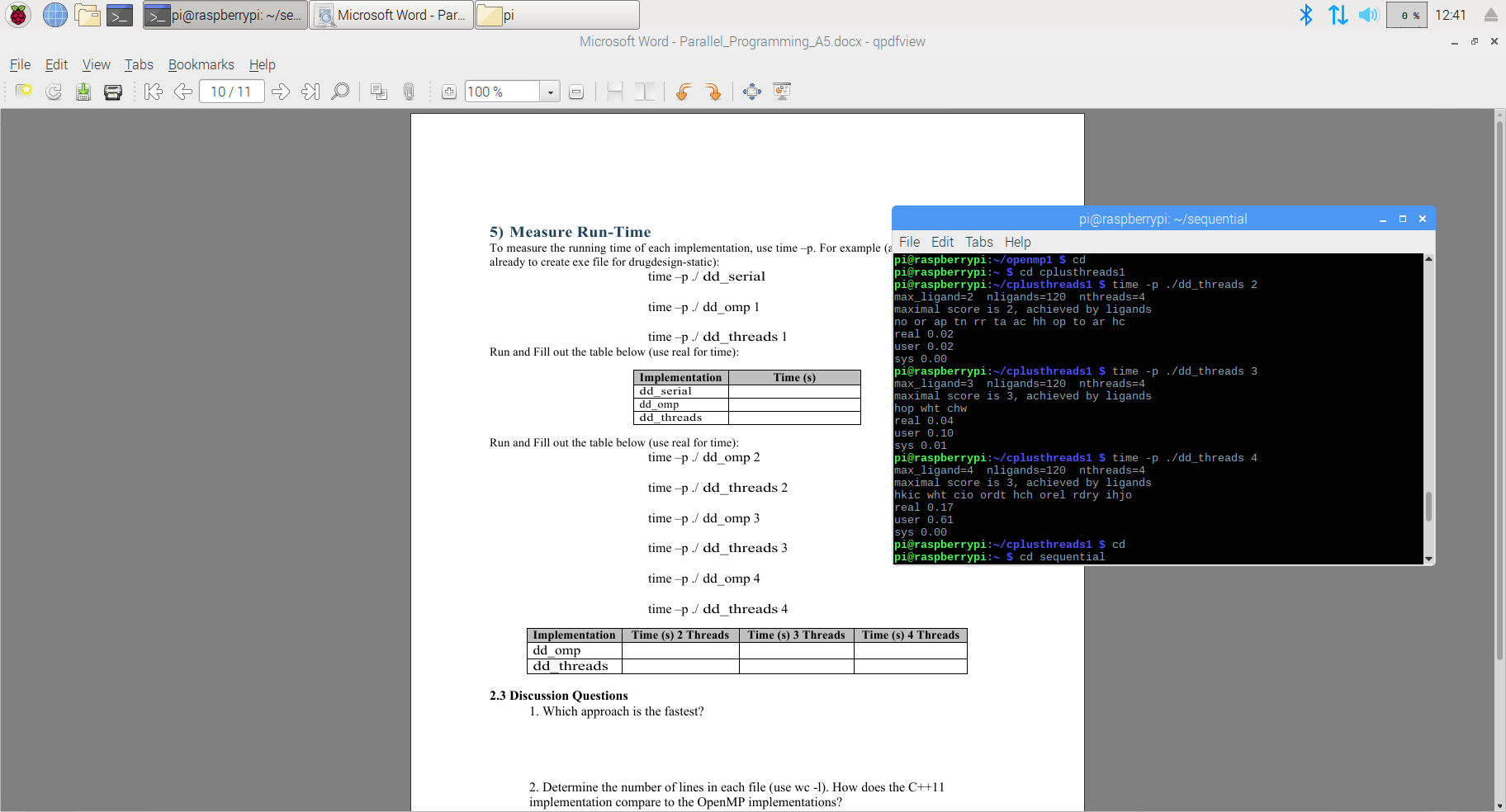
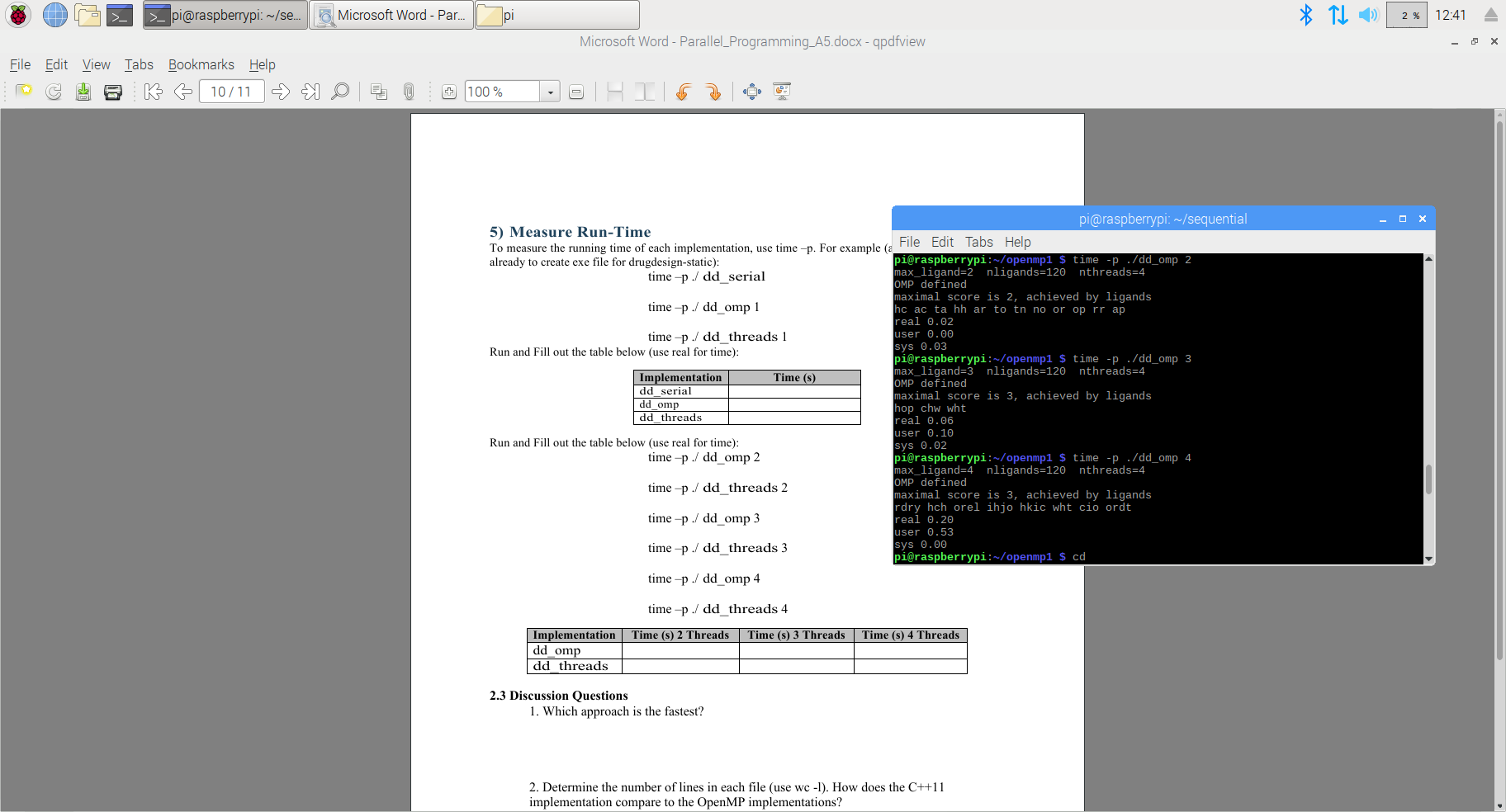
5)measure run time

5.1



|  |  |
| --- | --- |
| Implementation | Time(s) |
| dd\_serial | 132.34 |
| dd\_omp | 81.89 |
| dd\_threads | 41.93 |

5.2



|  |  |  |  |
| --- | --- | --- | --- |
| Implementation | Time(s) 2 Thread | Time(s) 3 Thread | Time(s) 4 Thread |
| dd\_omp | 0.02 | 0.06 | 0.20 |
| dd\_thread | 0.02 | 0.04 | 0.17 |

2.3 Discussion Questions

1.C++11 Thread is the fastest.

2.

|  |  |
| --- | --- |
| Implementation file(.cpp) | # of lines |
| dd\_serial | 170 |
| dd\_omp | 193 |
| dd\_thread | 207 |

As we can see C++11 implementation has more lines than openmp implementation. This is mostly due to the extra method in the C++11 implementation, which is the “do\_Maps”.

**void MR::do\_Maps(void) {**

**string lig;**

**tasks.pop(lig);**

**while (lig != SENTINEL) {**

**Map(lig, pairs);**

**tasks.pop(lig);**

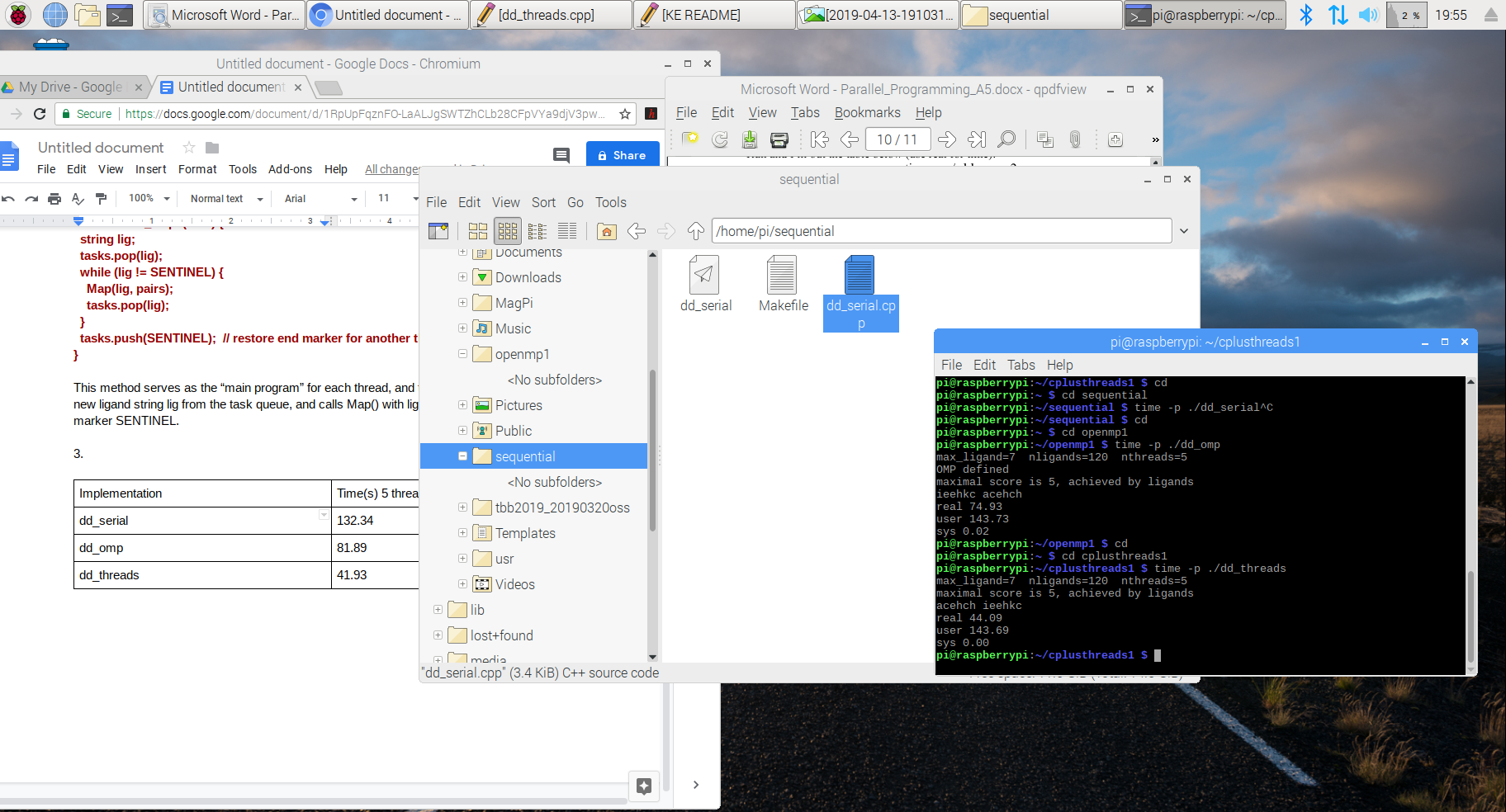
**}**

**tasks.push(SENTINEL); // restore end marker for another thread**

**}**

This method serves as the “main program” for each thread, and that method repeatedly pops a new ligand string lig from the task queue, and calls Map() with lig until it encounters the end marker SENTINEL.

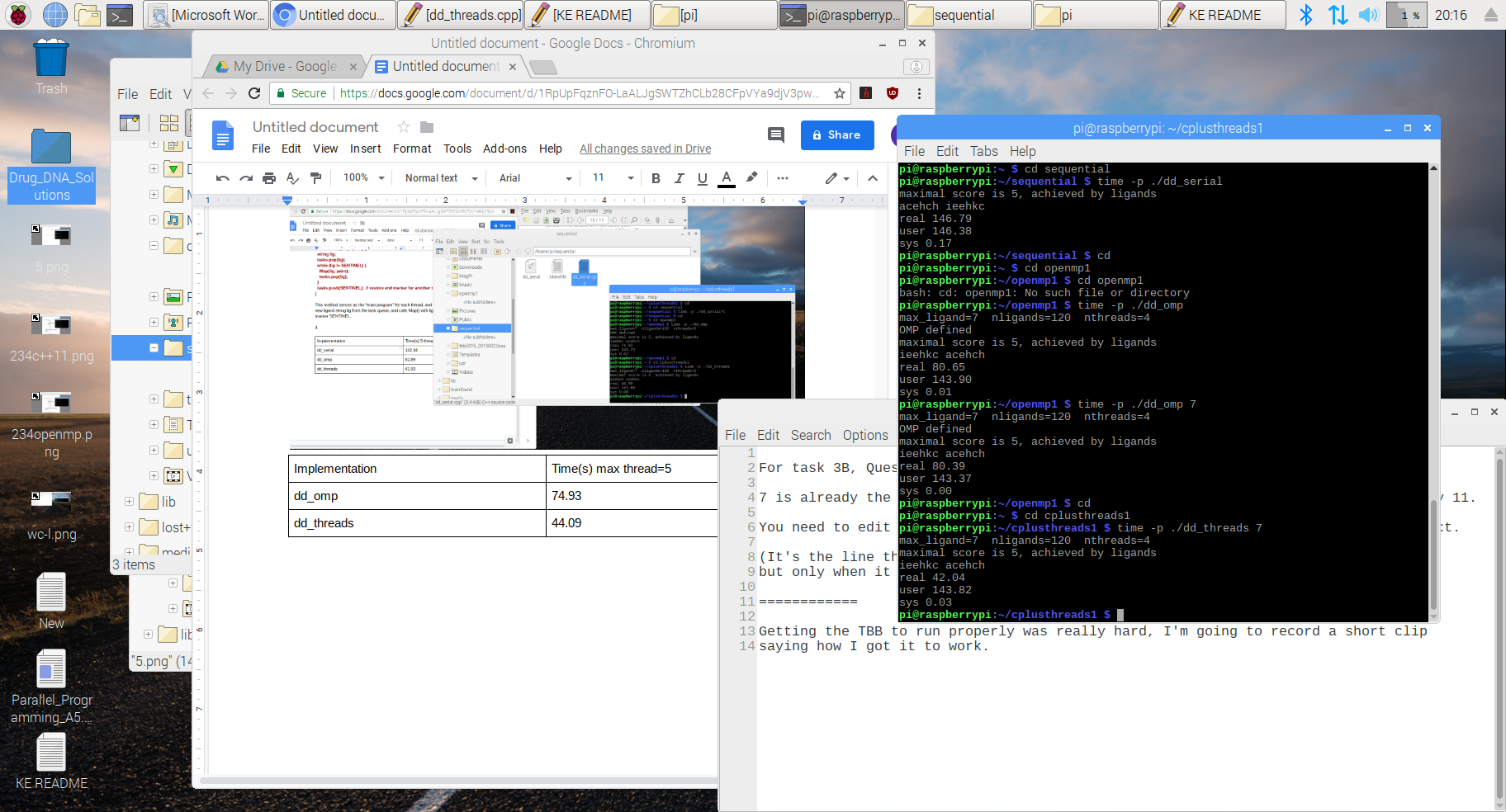
3.



|  |  |
| --- | --- |
| Implementation | Time(s) max thread=5 |
| dd\_omp | 74.93 |
| dd\_threads | 44.09 |

dd\_serial has no # of thread

4.



|  |  |
| --- | --- |
| Implementation | max\_ligand=7 |
| dd\_serial | 146.79 |
| dd\_omp | 80.39 |
| dd\_thread | 42.04 |

When i changed the max\_ligand to 11, i was unable to finish running the program, especially when i ran the openmp and C++11 thread implementations. My pi was overheated after the memory was in 100% of usage after long time and still didn’t finish running the program. I couldn’t finish this question, because i think it will destroy the pi.