Systems Programming Coursework 2A Report

**Status**

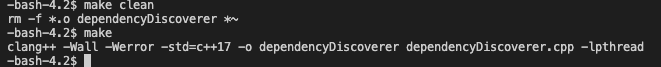
I have provided a multithreaded solution which compiles with thread safe structs. Furthermore, I have implemented my solution so that it avoids busy waiting by using *joinable().*

The program was executed on *ssh stlinux04*.

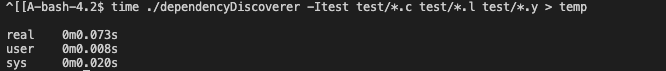
**Build**

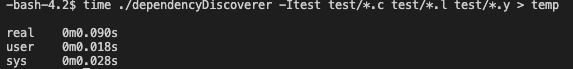
Directory



Compiling

Time

 Sequential

**** Single thread

**Multiple Threads**



Text

Description automatically generatedOne thread

Text

Description automatically generatedTwo threads

Text

Description automatically generatedThree threads

Text

Description automatically generatedFour threads

Text

Description automatically generatedSix threads

Text

Description automatically generatedEight threads

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CRAWLER\_  THREADS | 1 | 2 | 3 | 4 | 6 | 8 |
| Elapsed Time | Elapsed Time | Elapsed Time | Elapsed Time | Elapsed Time | Elapsed Time |
| Execution 1 | 0.090 | 0.073 | 0.093 | 0.100 | 0.088 | 0.136 |
| Execution 2 | 0.062 | 0.062 | 0.076 | 0.074 | 0.176 | 0.103 |
| Execution 3 | 0.061 | 0.061 | 0.062 | 0.071 | 0.085 | 0.104 |
| Median | 0.062 | 0.062 | 0.076 | 0.074 | 0.088 | 0.104 |

**Conclusions**

Executing the program with few additional seems to make the program run faster, however when several threads the program seems to slow down. A possible reason is that more time is spent creating more threads. As for variability there is a noticeable variability, and the trend is that the elapsed times decrease for more repetitions. This may be because instructions are being stored in cache.