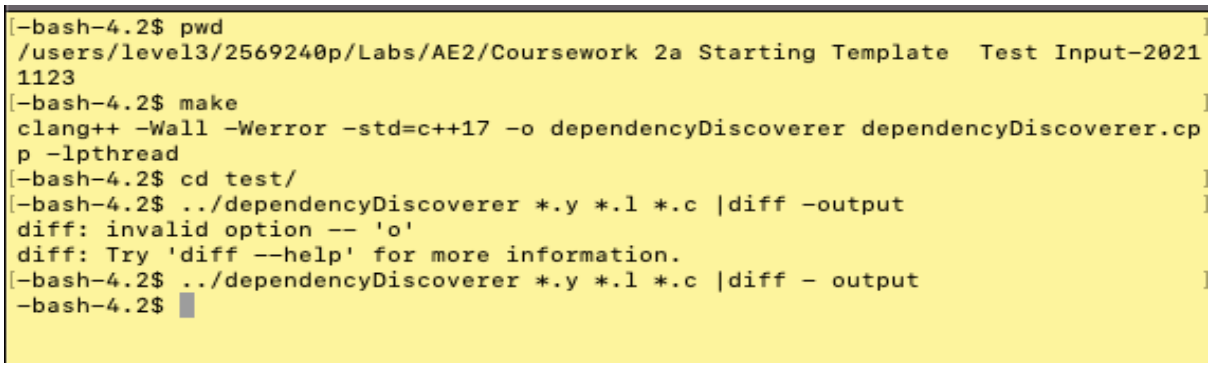
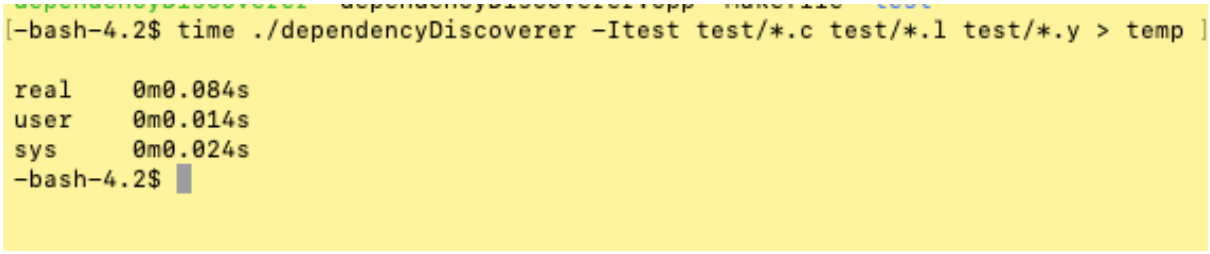
SP Assessed Exercise 2

Status Report: I have provided the multithreaded solution and the code runs as expected without any errors. The code has been tested on the school server using the commands provided in the specification sheet. I have not carried out any additional memory leak checks though. The evidence of testing has been provided below with screenshots.
When running on the server I had to install/source latest version of *clang* before I was able to compile the file using *make* command.

Evidence for the sequential crawler:
The sequence crawler i.e., dependencyDiscoverer.cpp file that was provided was uploaded to server without any modification. Then it was timed for crawling using command provided in the specification sheet.





As we can see that the real time for sequential crawler took 0.084 seconds.

Evidence for threaded crawlers:
Single Threaded Crawler:

```
[-bash-4.2$ pwd
 /users/level3/2569240p/Labs/AW2
[-bash-4.2$ make
 clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cp
 p -lpthread
[-bash-4.2$ cd test
[-bash-4.2$ ../dependencyDiscoverer *.y *.l *.c | diff - output
[-bash-4.2$ cd ..
[-bash-4.2$ export CRAWLER_THREADS=1
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.053s
user    0m0.005s
sys     0m0.022s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.051s
user    0m0.010s
sys     0m0.019s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.052s
user    0m0.010s
sys     0m0.016s
-bash-4.2$
```

As we can see, this screenshot also serves as evidence for compiling the code using make file and testing for the difference in output. The CRAWLER_THREADS has been set to 1 to make it a single tread program.

2 Threaded Crawler:

```
[-bash-4.2$ pwd
 /users/level3/2569240p/Labs/AW2
[-bash-4.2$ export CRAWLER_THREADS=2
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

 real    0m0.045s
 user    0m0.013s
 sys     0m0.026s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

 real    0m0.028s
 user    0m0.010s
 sys     0m0.018s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

 real    0m0.031s
 user    0m0.015s
 sys     0m0.014s
 -bash-4.2$
```

3 Threaded Crawler:

```
[-bash-4.2$ pwd                                                                ]
 /users/level3/2569240p/Labs/AW2
[-bash-4.2$ export CRAWLER_THREADS=3                                           ]
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.025s
user    0m0.013s
sys     0m0.018s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.024s
user    0m0.013s
sys     0m0.017s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.024s
user    0m0.009s
sys     0m0.021s
-bash-4.2$
```

4 Threaded Crawler:

```
[-bash-4.2$ pwd                                                                ]
 /users/level3/2569240p/Labs/AW2
[-bash-4.2$ export CRAWLER_THREADS=4                                           ]
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.029s
user    0m0.007s
sys     0m0.030s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.022s
user    0m0.020s
sys     0m0.012s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.022s
user    0m0.014s
sys     0m0.017s
-bash-4.2$
```

6 Threaded Crawler:

```
[-bash-4.2$ pwd                                                                ]
 /users/level3/2569240p/Labs/AW2
[-bash-4.2$ export CRAWLER_THREADS=6                                           ]
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.022s
user    0m0.015s
sys     0m0.020s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.020s
user    0m0.013s
sys     0m0.020s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.022s
user    0m0.014s
sys     0m0.023s
-bash-4.2$
```

8 Threaded Crawler:

```
[-bash-4.2$ pwd                                                                ]
 /users/level3/2569240p/Labs/AW2
[-bash-4.2$ export CRAWLER_THREADS=8                                           ]
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.019s
user    0m0.013s
sys     0m0.024s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.019s
user    0m0.011s
sys     0m0.025s
[-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp ]

real    0m0.020s
user    0m0.015s
sys     0m0.023s
-bash-4.2$
```

CRAWLER_THREADS	1	2	3	4	6	8
	Time (s)	Time(s)	Time(s)	Time(s)	Time(s)	Time(s)
Execution 1	0.053	0.045	0.025	0.029	0.022	0.019
Execution 2	0.051	0.028	0.024	0.022	0.020	0.019
Execution 3	0.052	0.031	0.024	0.022	0.022	0.020
Median	**0.052**	**0.031**	**0.024**	**0.022**	**0.022**	**0.019**
Mean	**0.052**	**0.035**	**0.024**	**0.024**	**0.021**	**0.019**

Discussion:
As we can see that as the number of threads increases, the time taken for the program to run decreases. The numbers go down from 0.052 seconds while using single thread to 0.019 when using 8 different threads. We can see that when using 4 thread, time we get for execution 1 is slightly higher compared to rest of thread 3 execution. Likewise, when using 2 threads, the execution time goes from 0.045 to 0.028 and then up to 0.031 seconds. These could be result of instability in server. The number of requests that was being handled by the server might have gone up or down hence we see some inconsistency in the reading.

In conclusion, from what we have observed, dividing a task into smaller pieces and assigning different threads (cores) to compute that data allows faster execution of the task.