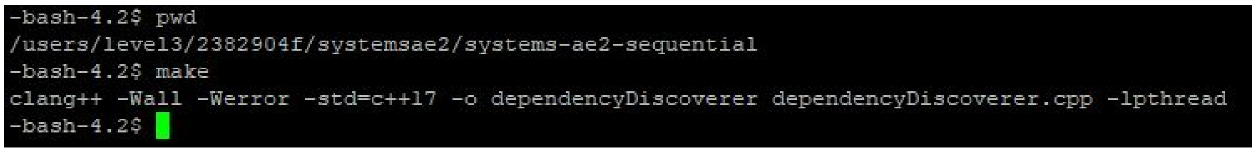
Georgios Flouri 2382904F

SP Exercise 2 Report

Status of solution

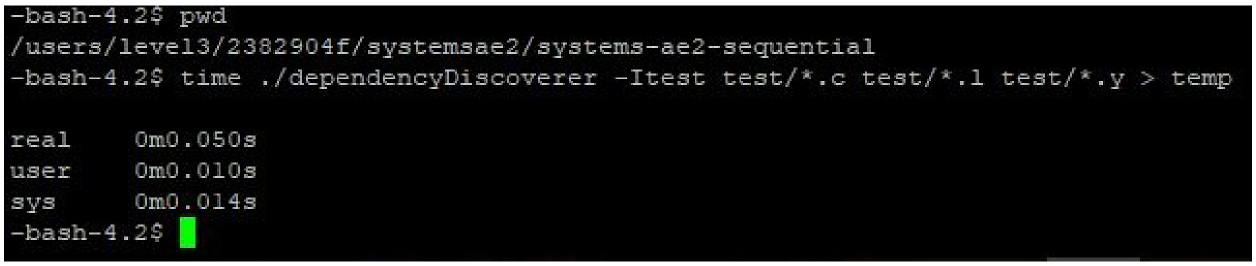
In my solution I implemented the multi threaded version of the exercise. The **dependencyDiscovered.cpp** file compiles with no errors. The compiled **dependencyDiscovered** program returns an identical output to that of the **output** file when run from inside the test folder using **$ ../dependencyDiscoverer *.y *.l *.c** . The command **$ ../dependencyDiscoverer *.y *.l *.c | diff - output** produces no output as requested.

Sequential Build



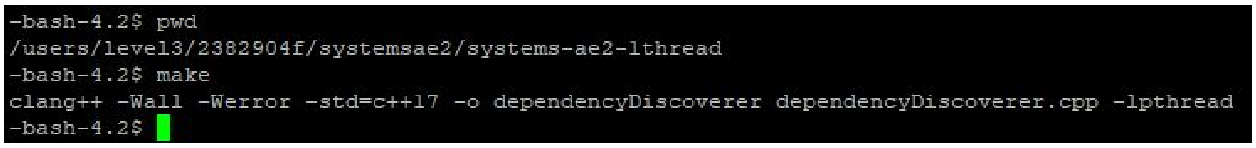
```
-bash-4.2$ pwd
/users/level3/2382904f/systemsae2/systems-ae2-sequential
-bash-4.2$ make
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lpthread
-bash-4.2$
```

Sequential Runtime



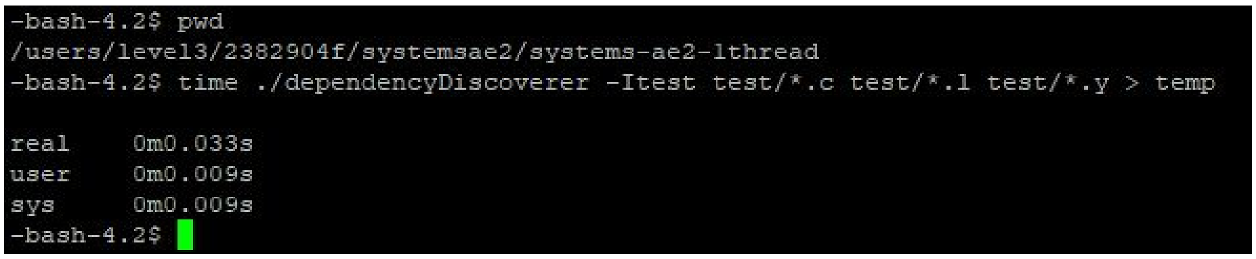
```
-bash-4.2$ pwd
/users/level3/2382904f/systemsae2/systems-ae2-sequential
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.050s
user    0m0.010s
sys     0m0.014s
-bash-4.2$
```

1-Thread Build



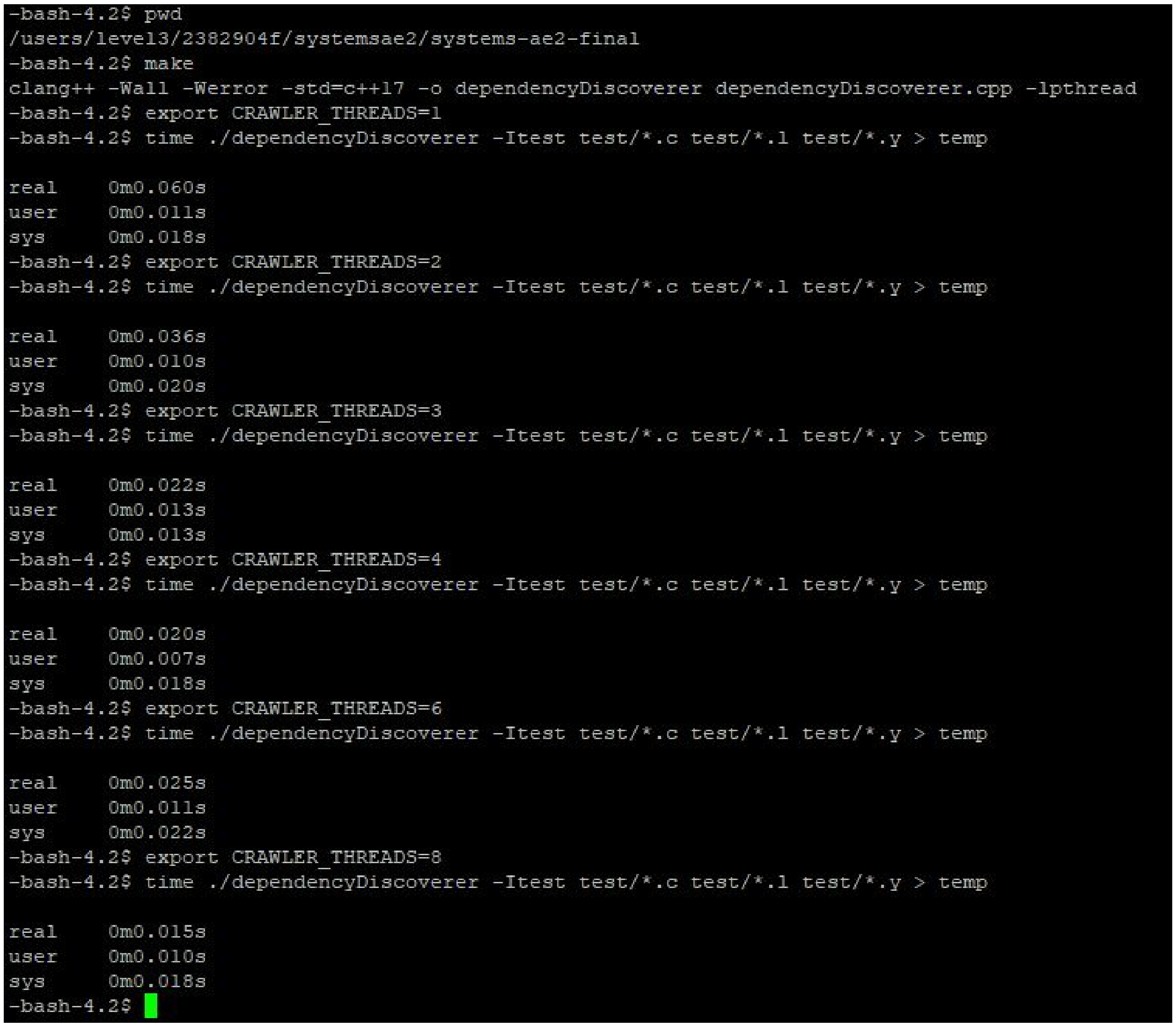
```
-bash-4.2$ pwd
/users/level3/2382904f/systemsae2/systems-ae2-1thread
-bash-4.2$ make
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lpthread
-bash-4.2$
```

1-Thread Runtime



```
-bash-4.2$ pwd
/users/level3/2382904f/systemsae2/systems-ae2-1thread
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.033s
user    0m0.009s
sys     0m0.009s
-bash-4.2$
```

Runtime with Multiple Threads Screenshot



```
-bash-4.2$ pwd
/users/level3/2382904f/systemsae2/systems-ae2-final
-bash-4.2$ make
clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lpthread
-bash-4.2$ export CRAWLER_THREADS=1
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.060s
user    0m0.011s
sys     0m0.018s
-bash-4.2$ export CRAWLER_THREADS=2
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.036s
user    0m0.010s
sys     0m0.020s
-bash-4.2$ export CRAWLER_THREADS=3
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.022s
user    0m0.013s
sys     0m0.013s
-bash-4.2$ export CRAWLER_THREADS=4
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.020s
user    0m0.007s
sys     0m0.018s
-bash-4.2$ export CRAWLER_THREADS=6
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.025s
user    0m0.011s
sys     0m0.022s
-bash-4.2$ export CRAWLER_THREADS=8
-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.015s
user    0m0.010s
sys     0m0.018s
-bash-4.2$
```

Runtime with Multiple Threads Experiment

CRAWLER_ THREADS	1	2	3	4	6	8
	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time
Execution 1	0.058	0.036	0.029	0.020	0.016	0.016
Execution 2	0.035	0.026	0.021	0.018	0.017	0.015
Execution 3	0.040	0.025	0.021	0.019	0.025	0.018
Median	0.040	0.026	0.021	0.019	0.017	0.016

Runtime with Multiple Threads Discussion

a) As can be clearly observed from the graph, additional cores lead to decreased elapsed time as expected. The benefit of additional cores is that execution becomes faster.

b) For each number of threads the three executions have variable elapsed times, however these values do not stray too far from each other. The small variation could be due to other processes running on the system affecting the execution time of the program.