

SP Exercise 2

Status:

I have implemented the multithreaded version, as described in the Specifications. The files compile and run with no warnings/errors. The file is set to have 1 thread, it can be changed from the variable `num_threads`.

Build and sequential & 1-Thread runtimes:

```
-bash-4.2$ clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer dependencyDiscoverer.cpp -lpthread
-bash-4.2$ clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer2 dependencyDiscoverer2.cpp -lpthread
-bash-4.2$ clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer3 dependencyDiscoverer3.cpp -lpthread
-bash-4.2$ clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer4 dependencyDiscoverer4.cpp -lpthread
-bash-4.2$ clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer6 dependencyDiscoverer6.cpp -lpthread
-bash-4.2$ clang++ -Wall -Werror -std=c++17 -o dependencyDiscoverer8 dependencyDiscoverer8.cpp -lpthread
-bash-4.2$ clear

-bash-4.2$ time ./dependencyDiscoverer -Itest test/*.c test/*.l test/*.y > temp

real    0m0.088s
user    0m0.014s
sys     0m0.026s
-bash-4.2$ time ./dependencyDiscoverer2 -Itest test/*.c test/*.l test/*.y > temp

real    0m0.041s
user    0m0.013s
sys     0m0.020s
-bash-4.2$ time ./dependencyDiscoverer3 -Itest test/*.c test/*.l test/*.y > temp

real    0m0.024s
user    0m0.011s
sys     0m0.018s
-bash-4.2$ time ./dependencyDiscoverer4 -Itest test/*.c test/*.l test/*.y > temp

real    0m0.029s
user    0m0.016s
sys     0m0.019s
-bash-4.2$ time ./dependencyDiscoverer6 -Itest test/*.c test/*.l test/*.y > temp

real    0m0.020s
user    0m0.010s
sys     0m0.023s
-bash-4.2$ time ./dependencyDiscoverer8 -Itest test/*.c test/*.l test/*.y > temp

real    0m0.023s
user    0m0.018s
sys     0m0.023s
-bash-4.2$ pwd
/users/level3/2315792c/Download/Coursework 2 - Cpp Template-20201119
-bash-4.2$ ls
dependencyDiscoverer      dependencyDiscoverer3.cpp  dependencyDiscoverer6.cpp  Makefile
dependencyDiscoverer2    dependencyDiscoverer4     dependencyDiscoverer8     temp
dependencyDiscoverer2.cpp dependencyDiscoverer4.cpp  dependencyDiscoverer8.cpp  test
dependencyDiscoverer3    dependencyDiscoverer6     dependencyDiscoverer.cpp
-bash-4.2$
```

Runtime with Multiple Threads:

CRAWLER_THREADS	1	2	3	4	6	8
	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time	Elapsed Time
Execution 1	0.088s	0.041s	0.024s	0.029s	0.020s	0.023s
Execution 2	0.079s	0.041s	0.024s	0.030s	0.021s	0.025s
Execution 3	0.049s	0.031s	0.024s	0.021s	0.019s	0.024s
Median	0.072s	0.038s	0.024s	0.027s	0.020s	0.024s

- Discussion:

a) From this experiment I have concluded that with more threads the execution times decreases by a lot. However I noticed this only makes a big difference only when using 3 or more threads. I assume with more data, more threads would be beneficial as they would minimize execution time.

b) With the lab machine barely loaded the times for 2 or more threads were constant but for the single thread the times show a slight difference. Assuming with a more loaded machine times would be less constant.