**My\_list-forming\_0.c**

-binds threads to CPU

-Doesn’t use a local list; attempts try\_lock before adding nodes to list

**My\_list-forming\_1.c**

-binds threads to CPU

**-Makes local list prior** to attempting try\_lock and adding to global list

**My\_list-forming\_2.c**

-binds threads to CPU

-Doesn’t have a try lock; **only locks and unlocks**; adds right to a global list

**My\_list-forming\_3.c**

**Doesn’t bind threads to specific CPU**

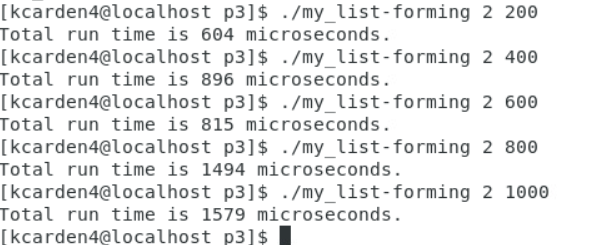
**Does use try lock** before adding

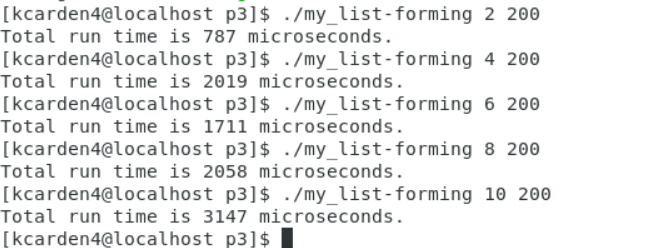
Doesn’t use a local list

My\_List-Forming\_0.c Runs (first number is K or # of nodes, second number is # of threads)

-binds threads to CPU

-Doesn’t use a local list; attempts try\_lock before adding nodes to list



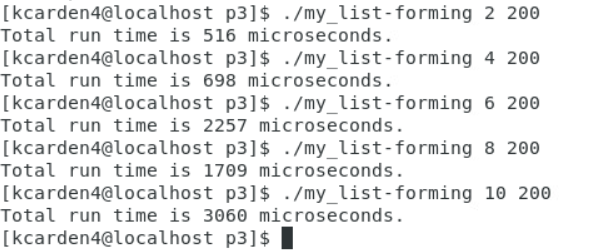


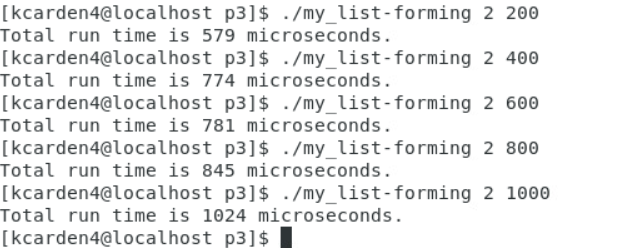
The more threads running, the longer the process to add nodes to the shared global variable

My\_List-Forming\_1.c Runs

-binds threads to CPU

**-Makes local list prior** to attempting try\_lock and adding to global list



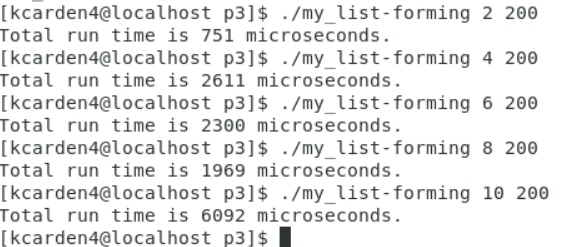


Same as previous case – running less threads does increase performance time. Also, my\_list-forming\_1.c runs faster overall because of using a local list and adding to global list once threads are done executing. This is faster than creating continually accessing the CR and adding more nodes to the global list.

My\_List-Forming\_2.c Runs

-binds threads to CPU

-Doesn’t have a try lock; **only locks and unlocks**; adds right to a global list





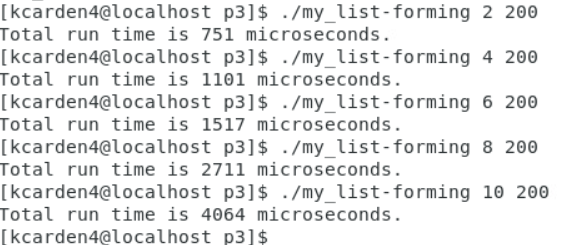
As previously, more threads leads to higher times. When threads are constant, it’s a little slower than 1.c but about the same time to finish as 0.c

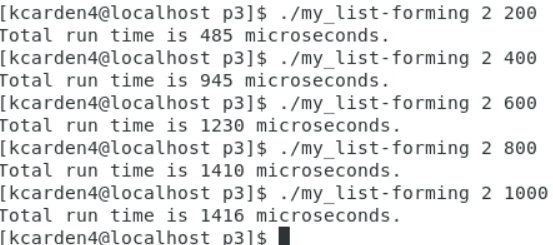
My\_List-Forming\_3.c Runs

**Doesn’t bind threads to specific CPU**

**Does use try lock** before adding

Doesn’t use a local list





Has the fastest running time of all iterations with a baseline of 2 threads and increasing K values only.

Using try\_lock function and not binding threads to CPUs does make the baseline 2 threads the fastest. However, the only some runs with increasing number of threads is mostly faster than other iterations, minus the my\_list-forming\_0.c running with some faster run times.