

Pthreads Activity
by Gabriel FitzPatrick (gjf7128@rit.edu)
2/26/24

Initial run output:

```
● o pthreads_demo } ; if ($?) { .\pthreads_demo }  
IN MAIN: Creating thread 0.  
IN MAIN: Creating thread 1.  
THREAD 0: Started.  
THREAD 0: Will be sleeping for 2 seconds.  
THREAD 1: Started.  
THREAD 1: Will be sleeping for 2 seconds.  
IN MAIN: Creating thread 2.  
IN MAIN: Creating thread 3.  
THREAD 2: Started.  
THREAD 2: Will be sleeping for 2 seconds.  
THREAD 3: Started.  
IN MAIN: Creating thread 4.  
THREAD 3: Will be sleeping for 2 seconds.  
IN MAIN: All threads are created.  
THREAD 4: Started.  
THREAD 4: Will be sleeping for 2 seconds.  
THREAD 1: Ended.  
THREAD 0: Ended.  
IN MAIN: Thread 0 has ended.  
IN MAIN: Thread 1 has ended.  
THREAD 2: Ended.  
IN MAIN: Thread 2 has ended.  
THREAD 3: Ended.  
IN MAIN: Thread 3 has ended.  
THREAD 4: Ended.  
IN MAIN: Thread 4 has ended.  
MAIN program has ended.  
o PS D:\RITClasses2023andOnward\SWEN342ConccurentSystems\swen-342\pthreads>
```

All threads sleep for 2 seconds with the initial program.

Run after replacing random timing:

```
eads_demo.c -o pthreads_demo } ; if ($?) { .\pthreads_demo }  
IN MAIN: Creating thread 0.  
IN MAIN: Creating thread 1.  
THREAD 0: Started.  
THREAD 0: Will be sleeping for 1 seconds.  
THREAD 1: Started.  
THREAD 1: Will be sleeping for 3 seconds.  
IN MAIN: Creating thread 2.  
IN MAIN: Creating thread 3.  
THREAD 2: Started.  
THREAD 3: Started.  
THREAD 3: Will be sleeping for 7 seconds.  
IN MAIN: Creating thread 4.  
IN MAIN: All threads are created.  
THREAD 4: Started.  
THREAD 4: Will be sleeping for 9 seconds.  
THREAD 0: Ended.  
IN MAIN: Thread 0 has ended.  
THREAD 1: Ended.  
IN MAIN: Thread 1 has ended.  
THREAD 2: Ended.  
IN MAIN: Thread 2 has ended.  
THREAD 3: Ended.  
IN MAIN: Thread 3 has ended.  
THREAD 4: Ended.  
IN MAIN: Thread 4 has ended.  
MAIN program has ended.  
o PS D:\RITClasses2023andOnward\SWEN342ConccurentSystems\swen-342\pthreads>
```

All threads sleep incrementally longer as they are created.

Run after implementation of reverse command line arg:

```
IN MAIN: Creating thread 0.  
IN MAIN: Creating thread 1.  
THREAD 0: Started.  
THREAD 0: Will be sleeping for 10 seconds.  
THREAD 1: Started.  
THREAD 1: Will be sleeping for 8 seconds.  
IN MAIN: Creating thread 2.  
THREAD 0: Ended.  
THREAD 1: Ended.  
IN MAIN: Creating thread 3.  
THREAD 2: Started.  
IN MAIN: Creating thread 4.  
THREAD 3: Started.  
THREAD 2: Will be sleeping for 6 seconds.  
IN MAIN: All threads are created.  
IN MAIN: Thread 0 has ended.  
THREAD 3: Will be sleeping for 4 seconds.  
THREAD 2: Ended.  
THREAD 4: Started.  
THREAD 4: Will be sleeping for 2 seconds.  
THREAD 3: Ended.  
IN MAIN: Thread 1 has ended.  
THREAD 4: Ended.  
IN MAIN: Thread 2 has ended.  
IN MAIN: Thread 3 has ended.  
IN MAIN: Thread 4 has ended.  
MAIN program has ended.  
PS D:\RITClasses2023andOnward\SWEN342ConccurentSystems\swen-
```

running with the reverse command line argument does just as you would expect “reverse” to do. The threads created first sleep for the longest rather than the other way around.

Lessons Learned:

I don't think I really learned a whole lot from this assignment compared to our previous assignment. This mostly served as a refresher for me on C programming which I rarely do.