**User-level Thread Library and Scheduler**

**Partners:**

Steven Nguyen - shn27

Ethan Febinger -

**pThread Functions & Structures:**

rpthread\_create:

allocated a new thread control block (tcb), sets it up, and enqueues it

if the scheduler is not initialized, initialize the scheduler by making the main context a listItem, making a context for the schedule function, and sets up the timer

rpthread\_exit:

marks a thread as finished and continues to scheduler

rpthread\_join:

checks if the thread to be joined is ENDED, if not, continue to the scheduler

when scheduled again, repeat above

rpthread\_mutex\_t:

mutexes store the thread that locked it and a linked list of threads

on an attempt to lock, if already locked, continue through the scheduler to be added to the mutex’s list of threads

schedule:

if not on mlfq mode, sched\_stcf(), else, sched\_stcf()

sched\_stcf:

this is triggered from a context swap, either from the timer or when continuing from a function like rpthread\_exit

if triggered from interrupt or yield or join, adjusts the time the current context has run, then enqueue back into the scheduler

if triggered from mutex, adjust the time the current context has run, then add to the mutex’s list of threads

if triggered from exiting a thread, the context stack is deallocated and dereferenced

afterwards, the timer is restarted and the scheduler context swaps to the next context

sched\_mlfq:

boop

**Benchmark Results:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | STCF (timer = 2.5ms) | MLFQ () | PTHREAD |
| ./vector\_multiply 2 | 24 micro-seconds |  | 140 micro-seconds |
| ./vector\_multiply 55 | 78 micro-seconds |  | 459 micro-seconds |
| ./vector\_multiply 1000 | 55 micro-seconds |  | 520 micro-seconds |
| ./parallel\_cal 2 | 2497 micro-seconds |  | 1394 micro-seconds |
| ./parallel\_cal 55 | 2501 micro-seconds |  | 556 micro-seconds |
| ./parallel\_cal 1000 | 2505 micro-seconds |  | 558 micro-seconds |
| ./external\_cal 2 | 9061 micro-seconds |  | 5194 micro-seconds |
| ./external\_cal 55 | 9103 micro-seconds |  | 3490 micro-seconds |
| ./external\_cal 200 | 9055 micro-seconds |  | 3542 micro-seconds |

Compared to the STCF scheduler, the PTHREAD scheduler is slower in vector\_multiply and faster in the other benchmarks.