

CSE 451 Sample Project Proposal

Group member names/netids:

Dwight K. Schrute (schrutefarms@uw.edu) and Pam Beesley (beastlybeesly@uw.edu)

Brief overview of topic you have chosen:

We have decided to implement a user thread library, modeled after the linux pthread library. We will implement the basic thread control functionality (create, join, exit, etc.) as well as user-level synchronization.

Initial Risk Analysis/Potential Problems:

Outside of create/join/exit we aren't sure which thread control functions are "essential" for us to implement and which we should add as stretch goals.

We also aren't sure what kinds of user-level synchronization is needed. It seems like some sort of mutex/lock would be good to have as well as maybe conditional variables.

We are also a bit concerned about handling scheduling, such as setting up some sort of preemption mechanism in order to switch between threads that are available to run.

Helpful resources you've found to research topic (at least three):

<http://man7.org/linux/man-pages/man7/pthreads.7.html>

<https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>

<http://ospp.cs.washington.edu/preview/toc.pdf>

<https://computing.llnl.gov/tutorials/pthreads/>

Minimum Viable Goal:

Our minimum viable goal would be a user thread library with fully functional create, join, exit methods. We want to test and demonstrate the correctness of our implementation by writing a user-level program that can interact with the user and create / destroy threads based user input. We would be using a yield function instead of relying on a timer interrupt (as this would require a signal implementation within the operating system). Basic synchronization tools will be provided, such as a mutex. A small testing suite will also be implemented.

Stretch Goal:

Our stretch goal would be to implement kernel threads, in addition to user threads, build more synchronization tools and expand the testing infrastructure.

User threads will also be preemptive, so they will not require a yield call.