Instructor Notes

5. Imagine you are in charge of implementing thread_create() in a new user-level thread library implemented on top of scheduler activations. What does your implementation need to do? How might the kernel respond? When might you avoid interacting with the kernel?

thread_create() in app:

library: 1) add new thread to ready list (user-level structure)

2) maybe tell kernel "add more procs"

response of kernel: could ignore

OR

upcall to lib "add this processor"

lib: pick ready thread, load state into SA, let run

Avoid?

- lib does not ask for more procs if outstanding request for more
- (avoid interactions if possible for overhead)

6. Imagine you must implement thread_exit(). What does your implementation need to do? When will you interact with the kernel?

thread_exit() in app:

library: clean up old thread in user-level thread list

put new thread from ready list on SA

If none on ready q? tell kernel "processor idle"

~~When can lib avoid interaction?~~

~~If app~~