Threading Library

We have implemented a threading library with the fuctions thread_create(), which accepts a function pointer and an argument, and thread_join(), in which the calling thread will wait on one of its children, or return -1 if it has none. We also created a user-level spinlock for threads.

Our design was to simply create a new process that shares and address space, current working directory, and file descriptors with its parent. This meant that much of the code was similar to fork(), except that the stack and EIP are set according to the arguments passed. We struggled to find a way to pass page-sized and page-aligned memory blocks to the syscall while ensuring that the same pointer would be returned. Our solution was to pass a 2 page block of memory, in order to guarantee that a page aligned block would be contained in it, find the upper section of this block, and store the original pointer.