

1. What are thread pools and what are they used for?

A thread pool threads that have been created beforehand. They are idle at first and are waiting to be given work. They are used because creating new threads requires a lot of overhead when there are a lot of short tasks to be completed. If the threads are already created, then you can use them repeatedly.

2a. You could store the keyboard input into a global variable. Then you lock the thread so that you don't have any conflicts. You can use `getchar()` to read the input in your thread. You know that the keyboard sends a signal to the thread that calls `getchar()`.

2b. The signal will go to the thread with the largest possible implementation range. If there is more than one thread that will allow a specific signal, then the kernel will just choose one.

2c. You could map specific regions of the file to certain threads. Each thread can only modify its specific region. This keeps different threads from overlapping with each other. Alternatively, you could lock the file so that only one thread can access it at once.

3. What is thread safe?

Code is considered thread safe if multiple threads can use it at the same time without problems occurring.

3b. If you are going to write a thread safe library what do you have to do?

You can use mutex locks to make updates to indexes atomic operations. Alternatively, you can write code so that has re-entrancy. This means that the code can be partially executed by a thread or simultaneously by another and still function. This requires all variables to be on the stack so that they are local.