

通过套接字传输一个大文件。使用共享内存的方法重新实现相同的应用程序。你觉得哪个版本性能更好？为什么？使用你写好的代码和不同的文件大小进行性能的测量。你观察到了什么？你认为在Linux内核中发生了什么导致这样的行为？

51. 实现一个基本的用户级线程库，该线程在Linux的上层运行。库的API应该包含函数调用，如`mythreads_init`、`mythreads_create`、`mythreads_join`、`mythreads_exit`、`mythreads_yield`、`mythreads_self`，可能还

有一些其他的。进一步实现这些同步变量，以便用户能使用安全的并发操作：`mythreads_mutex_init`、`mythreads_mutex_lock`、`mythreads_mutex_unlock`。在开始前，清晰地定义API并说明每个调用的语义。接着使用简单的轮转枪占调度器实现用户级的库。还需要利用该库编写一个或更多的多线程应用程序，用来测试线程库。最后，用另一个像本章描述的Linux2.6 O(1)的调度策略替换简单的调度策略。使用每种调度器时比较你的应用程序的性能。