

Lab6 实验报告

第一部分

这次的实验，我先是阅读了源码，学会了 pthread 库的使用，根据提示一步步编程，完成了实验。

```
== Test uthread ==
$ make qemu-gdb
uthread: OK (3.7s)
== Test answers-thread.txt == answers-thread.txt: OK
== Test ph_safe == make[1]: Entering directory '/home/chy/lab/lab6/xv6-labs-2020'
make[1]: 'ph' is up to date.
make[1]: Leaving directory '/home/chy/lab/lab6/xv6-labs-2020'
ph_safe: OK (9.6s)
== Test ph_fast == make[1]: Entering directory '/home/chy/lab/lab6/xv6-labs-2020'
make[1]: 'ph' is up to date.
make[1]: Leaving directory '/home/chy/lab/lab6/xv6-labs-2020'
ph_fast: OK (21.6s)
== Test barrier == make[1]: Entering directory '/home/chy/lab/lab6/xv6-labs-2020'
make[1]: 'barrier' is up to date.
make[1]: Leaving directory '/home/chy/lab/lab6/xv6-labs-2020'
barrier: OK (12.7s)
== Test time ==
time: OK
Score: 60/60
```

第二部分

遇到的问题

- 不了解 thread 切换上下文需要保存什么变量。
- 不知道如何编写锁。

解决方案

- 借鉴进程保持的上下文来定义线程的上下文。
- 调用 pthread 库来解决。

这次的实验用时一天。

第三部分

- 不太了解 context 与 trapframe 的区别。
- 了解到了 thread 是如何创建与控制并发的。