

Lab7 实验报告

第一部分

这次的实验，实现每 CPU 空闲列表，首先需要修改 kmem 数据结构，为每个 CPU 分配锁和空闲列表，同时，由于我修改了 kmem，对于 kinit、kalloc、kfree 函数中对内存的操作都需要做出相应修改。这里遇到的主要问题是如何处理一个 CPU 没有空闲列表，但另一个 CPU 的列表有空闲内存的情况下，对 CPU 的切换，切换的期间注意使用 push_off() 和 pop_off() 来开关中断。然后第二个实验与第一个实验关系不大，为了让 CPU 可以并行工作，优化 cache 锁，利用对质数取余实现哈希函数，可以使用固定数量的存储桶，而不动态调整哈希表的大小。使用主要数量的存储桶来减少散列冲突的可能性。第二个实验最后在组员的帮助下完成的，还是不太理解。

```
== Test running kallocetest ==
$ make qemu-gdb
(72.0s)
== Test kallocetest: test1 ==
kallocetest: test1: OK
== Test kallocetest: test2 ==
kallocetest: test2: OK
== Test kallocetest: sbrkmuch ==
$ make qemu-gdb
kallocetest: sbrkmuch: OK (8.2s)
== Test running bcachetest ==
$ make qemu-gdb
(3.7s)
== Test bcachetest: test0 ==
bcachetest: test0: OK
== Test bcachetest: test1 ==
bcachetest: test1: OK
== Test usertests ==
$ make qemu-gdb
usertests: OK (90.5s)
== Test time ==
time: OK
Score: 70/70
```

第二部分

遇到的问题

- 报错 panic: freeing free block。

解决方案

- NBUCKET 需要小一点 因为锁的数目有限 而 binit 至少需要 NBUCKET+ NBUCKET 个锁 这次的实验用时两天。

第三部分

- 这次实验让我对锁的理解与使用更加深刻娴熟了。
- 这次实验让我对于多 CPU 下程序的优化有了更深的理解。