实验思路

具体实现

三种替换策略

实验总结

收获

在代码能力方面得到了较大提升，体会较大的是位运算和面向对象的思想，以及要注意编写单测。

有时候位运算可能并不能得到自己预想的结果，尤其是在cache.cpp文件当中用到了大量的位运算来实现，例如常量数字并不会自己变成我以为的uint64\_t类型，需要进行强制类型转换，否则如果用一个常量数字1来直接和一个uint64\_t类型的数据进行按位与，它会自动向长度较短的类型进行转换，也就是说uint64\_t会丢失前32位。这也是为什么最开始我在取出flag时会丢失不少的数据。

面向对象的思想要感谢刘丰源同学对我的帮助，在替换策略上利用多态可以使代码结构更加清晰，也能减少很多代码量。此外我了解了纯虚析构函数必须自己定义函数体，否则会出现链接错误。

编写单测会非常有助于定位bug，会极大地减少工作量。例如lru和plru的替换策略我都在正式应用前编写了一些测试，简单检验了正确性，因此在加入到程序框架当中之后并没有在替换策略这方面发现问题。而cache我没有进行测试，因此在正式运行时几乎所有的bug都来自cache的位运算。

在对cache的理解方面也得到了较大的帮助。一方面是全相联之前了解很少，通过实验得到了进一步理解；另一方面是写直达和写回策略对cache实现的影响。

和同学之间的交流

感谢计71班的刘丰源同学在代码实现上对我的帮助，在代码实现框架和细节处理方面都帮助很大，包括但不限于：

在lru和plru策略实现的头文件当中使用静态变量以节省空间，

lru部分栈的位运算拆分函数

尤其是在cache的替换策略部分的面向对象思想

这一点上对我启发很大，代码结构因此变得较为清晰。

感谢计85班杨雅儒同学在cache的相关知识方面对我的帮助，例如帮助我理解了cache的写直达和写回。