操作系统实验一

体验Nachos下的并发程序设计



小组成员：

覃龙虎22920172204029

夏心蕾22920172204240

王宇昕22920172204228

沈丽姗22920172204196

日期：2020.3.23

一、实验内容：

(1) 安装nachos

(2) 实现双向有序链表（50%）

如对c++不很熟悉，可以参考nachos-3.4/c++example/中的有关实现。

(3) 体验nachos线程系统（50%）

需要做的更改有：

(a) 将dllist.h, dllist.cc, dllist-dirver.cc等文件拷贝到nachos-3.4/code/threads/目录中。

(b) 修改Makefile.common中的THREAD\_H、THREAD\_C、THREAD\_O以保证新的文件确实被编译了。

(c) 根据实验内容，main.cc，threadtest.cc等文件可能需要改动。

二、实验过程：

2.1 Nachos安装与源码学习

2.1.1源码阅读

本次实验主要学习并发编程，并研究并发编程时的错误。

（1）thead.cc文件：

Fork(VoidFunctionPtr func,int arg)：func是新线程运行的函数，arg是func函数的入参，Fork的实现包括分为几步：分配一个堆栈，初始化堆栈，将线程放入就绪队列。

Finish()：不是直接收回线程的数据结构和堆栈，因为当前仍在这个堆栈上运行这个线程。先将threadToBeDestroyed的值设为当前线程，在Scheduler的

Run()内切换到新的线程时在销毁threadToBeDestroyed。Yield()、Sleep()。这里实现的方法大多是都是原子操作，在方法的一开始保存中断层次关闭中断，并在最后恢复原状态。

Yield()：当前线程放入就绪队列，从scheduler就绪队列中的找到下一个线程上cpu，以达到放弃CPU的效果。

（2）utility.cc

此外，为了调试，我们阅读相关debug错误说明.

实际操作中，我们主要采用ABORT和ASSERT居多；

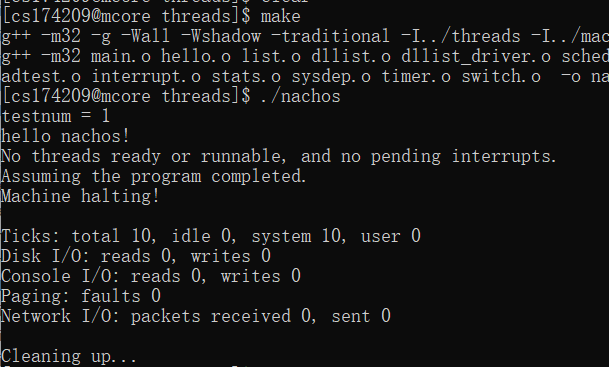
以及使用DEBUG()结合命令行打印相应信息。

（3）threadtest.cc

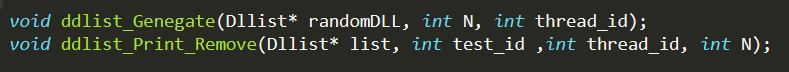
nachos内核线程测试部分，Fork两个线程，交替调用Yield()主动放弃CPU，执行循环体，会发现线程0和线程1刚好是交替执行.

2.1.2几个基本测试

（1）测试hello函数

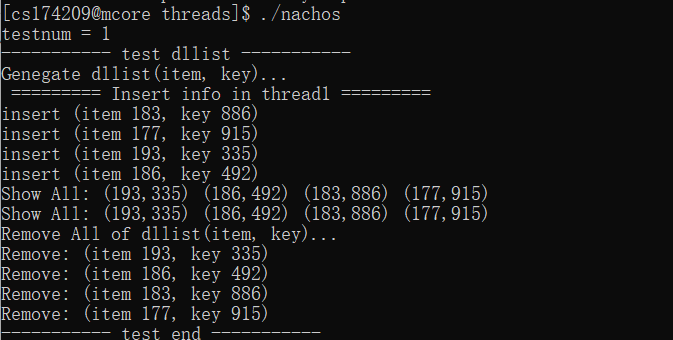


（2）测试双链表的插入和删除（dllist-driver.cc）

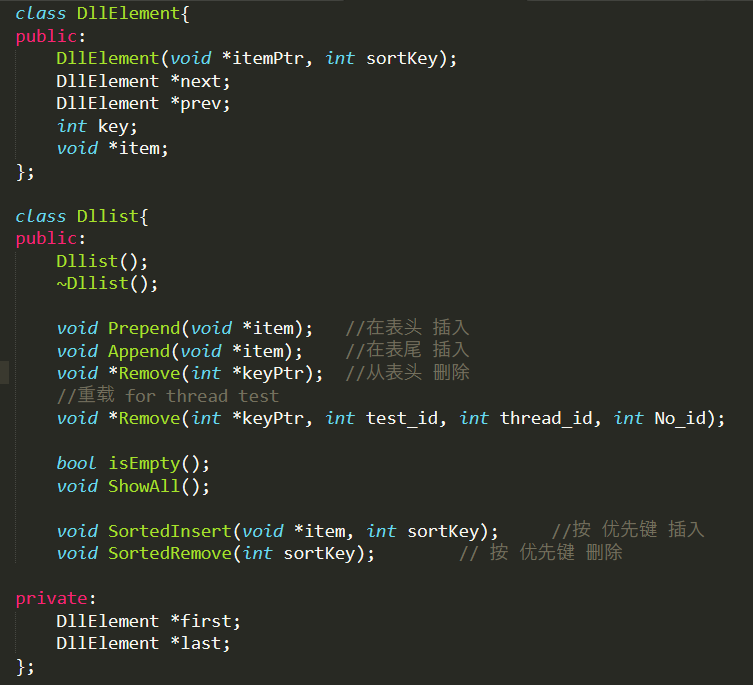


(为了后续线程测试，扩展了参数测试号、操作数量、线程id)

测试结果：



此外，这里对我们的dllist 简要的说明如图：

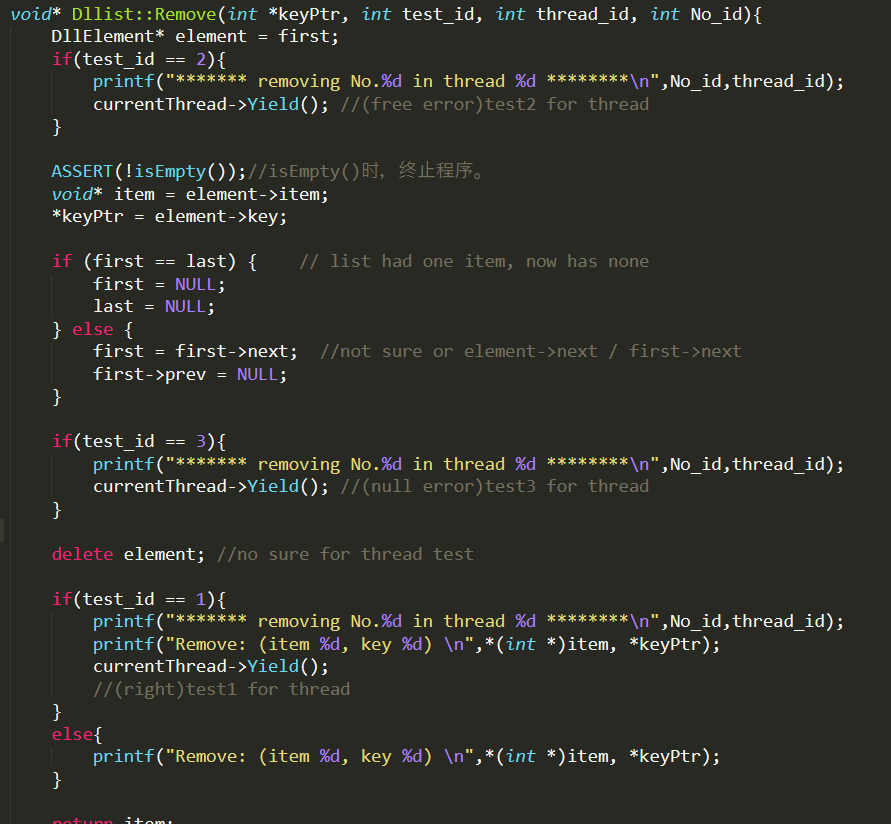


2.2并行问题展示

2.2.1 remove操作问题

我们先让每个线程的所有插入操作（Generate\_DDlist）完成后，在专门研究remove操作的并行问题

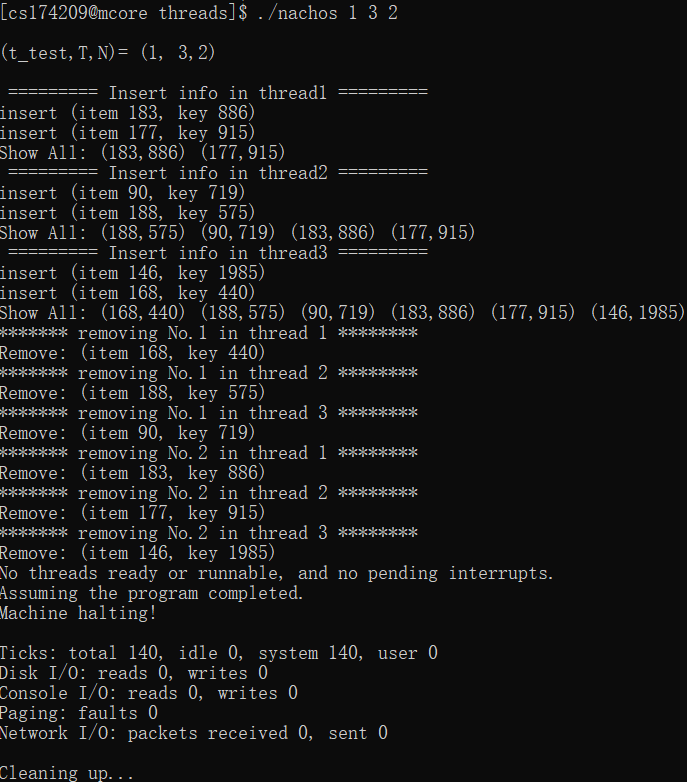
Remove测试函数如下：



（1 正常情况

测试1：每个线程的remove完整的完成后再切换线程

结果：remove正常交替进行

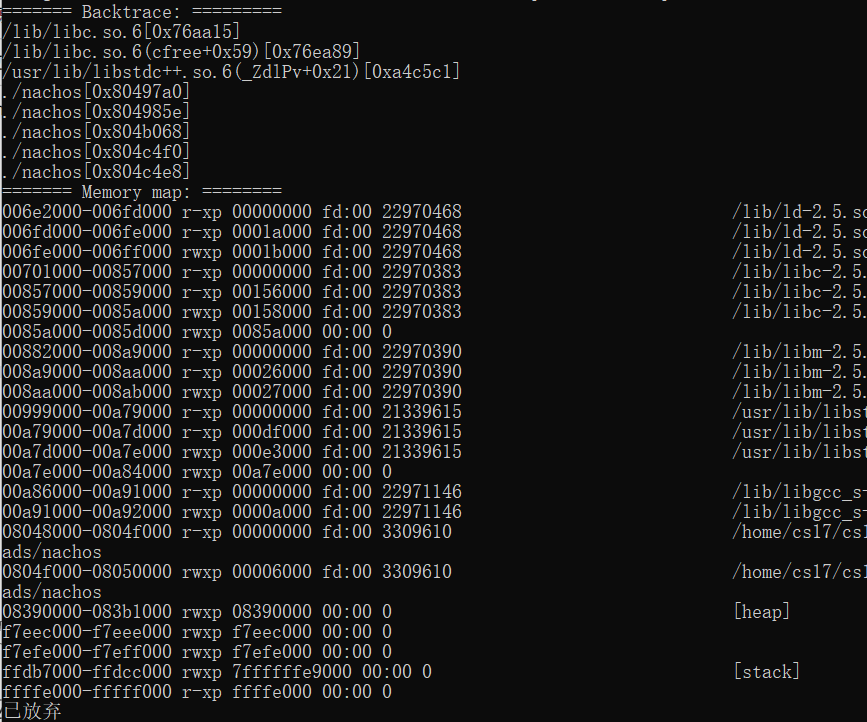
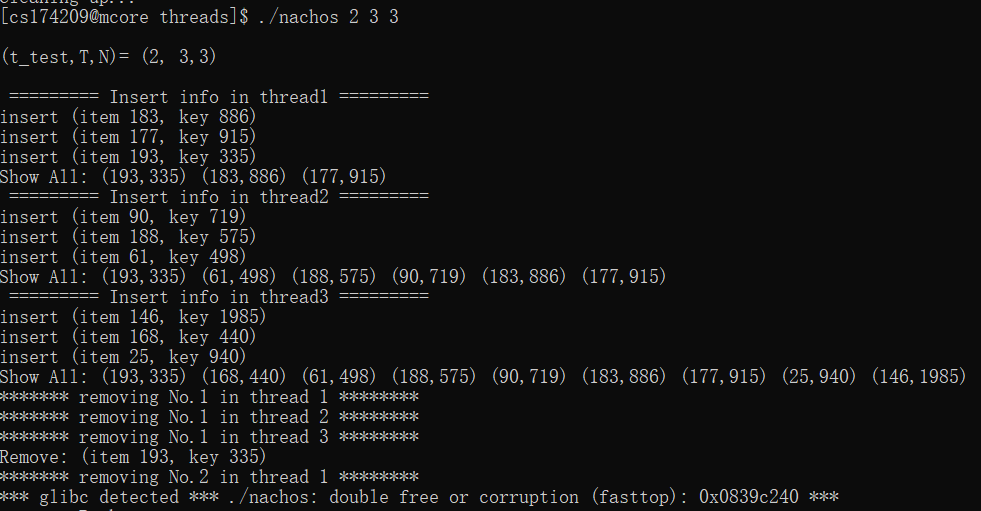


（2 并发错误

测试2：remove操作时（removing），在first = first -> next前，切换线程

结果：可以看到，在第二次remove还没有完成时，出错

出错原因：当线程2回到remove操作时，delete掉的指针是线程1已经delete后的指针，所以出错（double free or curruption）



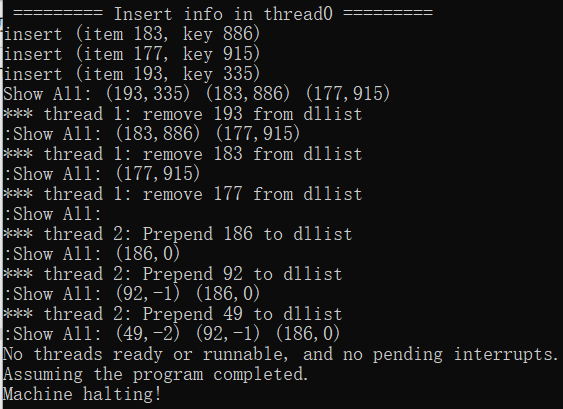
2.2.2 “生产/消费”错误

除了在一个小操作中并行程序可能会出错，即使remove为一个原子操作也会出错

( 1 逻辑错误

测试4（正确）、5（错误）

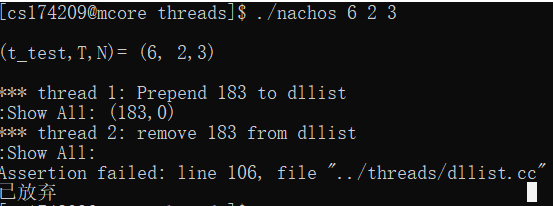
若原本有N个元素，现在要先删除这N个，再新增N个，那么并行就会得到非预期结果



（2 过度消费（列表空）

测试6:

在进行一次插入一次删除后，在接着的一条删除操作时，由于列表为空，ASSERT错误信息。



三、困难以及解决

困难1：编译错误

开始时，make编译出错，发现是makefile.common中少添加了新建的.cc/.h文件项。

以及后来由于版本原因，丢失相关文件。

困难2：编程时，C函数库和已有定义冲突

通过报错信息，找到冲突冲突项

使用类似如下方式解决

extern "C"

{

#include <sys/time.h>

void srand(unsigned seed);

int rand(void);

void \*malloc(unsigned int num\_bytes);

}

困难3: 多次出现段错误

这里注意编程时，段错误通常是访问存储错误、指针问题等，本次遇到的有段错误的原因包括编码逻辑错误，和多线程并发错误。

困难4: 如何方便的展示几种测试

使用命令行参数来处理。