# nachos

## 目录



## Lab1 线程机制

### 内容一：总体概述

> 【用简洁的语言描述本次lab的主要内容；阐述本次lab中涉及到的重要的概念，技术，原理等，以及其他你认为的最重要的知识点。这一部分主要是看大家对lab的总体的理解。
> 要求：简洁，不需要面面俱到，把重要的知识点阐述清楚即可。】

### 内容二：任务完成情况

**内容完成列表**

| | Exercise1 | Exercise2 | Exercise3 | Exercise4 |
|---|---|---|---|---|
| Part 1 | Y | Y | Y | Y |

**具体内容完成情况**
**Part 1**
**Exercise 1**

**Exercise 2**

**Exercise 3 扩展线程的数据结构**
**增加“用户ID，线程ID”两个数据成员，并在Nachos现有的线程管理机制中增加对这两个数据成员的维护机制**

用户ID：在Thread类中增加一个私有变量userId，使用nachos宿主系统的用户id作为nachos的用户id。在Thread.cc中的Thread的构造函数中使用getuid()来获取当前系统的uid并初始化线程的userId,并在Thread.h中添加一个成员函数getUid()返回userId。

线程ID：同样在Thread类中增加一个私有变量threadId。
新创建一个线程时，分配系统中没有被分配的最小的tid给新的线程（在system.h和system.cc中声明和定义AllocateTid()函数），在Thread()构造函数中调用AllocateTid()。
当一个线程被回收时，把该线程持有的tid释放（在system.h和syetem.cc中定义函数FreeTid()),并在~Thread()中调用这个函数。

**Exercise 4 增加全局线程管理机制**
**在Nachos中增加对线程数量的限制，使得Nachos中最多能够同时存在128个线程；**

在system.h和system.cc中增加对线程数量的限制，定义一个宏THREAD_POOL_SIZE设置最大线程数，并用一个bool数组tids[]来表示每个tid的分配状态，tid的分配释放规则在上一个问题中已经说明。

**仿照Linux中PS命令，增加一个功能TS(Threads Status)，能够显示当前系统中所有线程的信息和状态。**

**要求**

【对于阅读代码类的exercise，请对其中你认为重要的部分（比如某文件，或某个类、或某个变量、或某个函数……）做出说明。
对于要编程实现的exercise，请对你增加或修改的内容作出说明。如果增加或修改了某个类，请写出这个类写在那个文件中，它的的功能是什么，你自己添加或修改的成员变量以及成员函数有哪些，它们的功能是什么；特别的，对于复杂的函数，请说明你的实现方法。不需要贴具体的实现代码。
要求：表述清楚即可， 可采用图表来辅助说明，不需要贴代码】

## 内容三：遇到的困难及解决办法

**困难1**
**困难2**
**要求**

【描述你在实习过程中遇到的困难，是与实习直接相关的技术方面的难题。突出说明你是如何攻克这些难关的。
要求：只需写一下有较深体会的困难。如果觉得整个过程都比较简单的话此部分可不用写。】

## 内容四：收获及感想

**要求**

【自己的收获，任何关于实习的感想，可以是技术方面的或非技术方面的，可随意发挥。
要求：内容不限，体裁不限，字数不限，多多益善，有感而发。】

## 内容五：对课程的意见和建议

## 内容六：参考文献

**要求**

【我们希望大家不要使用复制粘贴来拼凑你的报告。详细地列出你在完成lab的过程中引用的书籍，网站，讲义，包括你咨询过的大牛们。
要求：诚实，格式尽量按照论文的要求，请参考“论文参考文献格式.doc”】

# 备注

各种的宏定义在对应部分的Makefile中的DEFINE中定