操作系统课程设计任务书

抢占式用户级线程库的设计与实现

一、目标：

本课程设计要求在linux环境下实现一个抢占式的用户级线程库mythread。目的是使得学生理解操作系统的调度机制以及调度算法。

二、设计内容

本课程设计要求学生在linux环境下用C语言设计并实现一个抢占式的用户级线程库，支持线程的6种状态（见下文），支持线程管理的相关函数，并支持线程互斥锁。

线程的调度采用时间片轮转法。每个线程被赋予一个时间片。当线程的时间片用完时，该线程被抢占。线程的抢占是通过时钟信号实现的。时间片的大小以毫秒为单位，在mythread\_init(period)中定义。

使用mythread线程库的程序必须首先调用mythread\_init函数初始化现场环境。其主要功能为：

1. 设置定时器产生定时的周期为period；
2. 捕获定时信号，设置定时信号处理函数；
3. 设置线程库的环境；
4. 设置主函数为0号线程，并设置其环境。

线程调度的时机在定时信号处理函数中进行。

线程库中的每一个线程应当有一个线程控制块uthread，其结构见thread.h。线程创建时，通过malloc动态申请线程控制块，并将其放入适当的链表中；线程结束时，由等待其结束的线程将控制块从链表摘除，并释放空间。

任何时刻，线程的状态均处于如下状态之一：

NEWTHREAD：线程正在创建，还没有被调度运行

RUNNING：线程正在运行

COMPLETE：线程结束（不通过mythread\_exit）

EXITED：线程结束（通过mythread\_exit）

CANCEL：线程被另一个线程杀死

BLOCKED：线程被阻塞

线程库应当实现如下函数：

int mythread\_create(mythread\_t \* thread,void \* (\*start\_routine)(void \*),void \* arg);

创建一个线程。第一个参数是返回的线程ID，第二个参数是线程要执行的函数，第三个参数是线程要执行的函数的参数。成功时返回0，出错时返回-1。

int mythread\_join(mythread\_t thread, void \*\*status);

等待指定的线程结束，并返回线程的结束码。0：成功，-1：出错。

void mythread\_exit(void \* retval);

线程结束，并返回线程函数的返回值到retval

void mythread\_yield(void);

线程让步。

mythread\_t mythread\_self(void);

返回线程自己的ID。

int mythread\_equal(mythread\_t t1, mythread\_t t2);

比较两个线程，非0表示相同，0表示不同

int mythread\_cancel(mythread\_t thread);

发送[终止信号](https://baike.baidu.com/item/%E7%BB%88%E6%AD%A2%E4%BF%A1%E5%8F%B7)给thread[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B/103101)，如果成功则返回0，否则为非0值。

int mythread\_mutex\_init(mythread\_mutex\_t \* mutex);

初始化一个互斥锁。

int mythread\_mutex\_lock(mythread\_mutex\_t \* mutex);

上锁。

int mythread\_mutex\_unlock(mythread\_mutex\_t \* mutex);

释放锁。

三、时间安排

1.第13周：布置课程设计内容；

2.第14周-18周：完成课程设计内容，并提交课程设计报告。

四、设计工作要求

1.独立完成课程设计内容及报告，并在规定的时间内提交。提交内容包括：1）课程设计报告书；2）源代码。

2.你的课程设计不得使用任何POSIX线程库或linux线程库，或者对其进行包装。

3.线程切换可使用（getcontext、makecontext、setcontext、swapcontext）和（siglogjmp、sigsetjmp）两组函数之一，但不可以两者混用，事实上两者混用在技术上也不可行。

4.我提供给你的材料：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 文件名 | 作用 | 用法 |
| 1 |  | 课程设计任务书 |  |
| 2 |  | 课程设计报告书模板 |  |
| 3 | mythread.h | 头文件 |  |
| 4 | mythread.a | 示例库 | 你可以用如下命令将测试程序编译，并与该示例库连接，运行测试程序，理解线程库的功能：  gcc -o thread phi.c mythread.a |
| 5 | phi.c | 测试程序 | 用于测试线程库的测试程序 |
| 6 | Makefile | 生成你的mythread.a的makefile |  |

5.你写的程序命名为mythread.c，并将其编译为mythread.a，替换我给你的mythread.a。你写的程序应包含设计内容中所要求的所有函数。

6.不可以修改提供的测试程序，不得反汇编libthread.a。

7.鼓励在完全完成基本功能的前提下，增加新的功能。新的功能包括：1）增加新的调度算法，即时间片+优先级；2）增加线程同步功能，即条件变量

8.本课程设计编程语言为c语言，编译环境为linux+gcc。

五、成绩评定

1.你的线程库必须保证编译通过（15分），编译命令：

gcc main.c thread.c -o thread

2.完全通过测试程序（50分）

3.良好的代码风格及注释（10分）

4.课程设计报告的质量（25分）

六、参考文献

【1】 http://blog.csdn.net/cyberlabs/article/details/6920138

【2】 http://blog.csdn.net/qq910894904/article/details/41911175

【3】 W.Richard Stevens,unix环境高级编程，人民邮电出版社

【4】 https://linux.die.net/man/