

Lab4 挑战性任务

在课程所实现的 MOS 系统中，我们通过记录进程控制块、保存现场等方式实现了进程相关的模块。然而，为了发挥操作系统强大的功能，仅有进程这一粒度往往是不够的，我们还需要线程 (thread) 的机制来实现更多场景下的并发服务。

结合操作系统理论课上学习的有关线程的知识，以及我们实验代码中有关进程的现有实现，在本挑战性任务中，我们希望你基于已有的架构进行增量开发，实现**线程相关机制**。同时，为了解决线程并发带来的竞态问题，你还需要实现**信号量 (semaphore) 机制**，以控制线程的同步互斥关系，保证服务的线程安全性。

实现参考：

- 从零建立起一个相对完善的线程机制是较为困难的。我们梳理一下思路：
 - 首先，类似于进程控制块 PCB，对线程也同样需要引入记录线程相关状态的数据结构
 - 不同于**内核中**进程控制块的初始化、创建、销毁等函数，对线程的操作需要一套完整的**用户态**接口函数，供用户进程调用
 - 与进程不同的是，线程之间的地址空间是共享的，需要合理布局页面使得同一进程的不同线程可以相互访问数据
- 信号量相关请参考理论课中讲述的内容

在本挑战性任务中，要求按照 **POSIX 标准**，结合 MOS 操作系统已有的架构实现这两个机制。

POSIX Threads

POSIX 线程 (POSIX Threads) 是由 IEEE Std 1003.1c 定义的一套线程标准。你需要至少实现标准中的以下函数 (省略参数)：

- `pthread_create()`：创建线程
 - 该函数所涉及的 `pthread_attr_t` 的具体功能不要求实现
- `pthread_exit()`：终止线程
- `pthread_cancel()`：撤销线程
- `pthread_join()`：阻塞至线程结束

参考链接：[Linux Tutorial: POSIX Threads \(cmu.edu\)](https://www.cmu.edu/~rjeffrey/linux/tutorial/posix/threads/)

POSIX Semaphore

POSIX Semaphore 是由 IEEE Std 1003.1b 标准定义的一套信号量标准。该标准提供的信号量分为有名和无名信号量，无名信号量用于进程内同步与通信，有名信号量可以用于进程内同步与通信，也可以用于进程间同步与通信。

你需要至少实现标准中的以下函数 (省略参数)：

- `sem_init()`：初始化信号量
- `sem_destroy()`：销毁信号量
- `sem_wait()`：对信号量的 P 操作 (阻塞)
- `sem_trywait()`：对信号量的 P 操作 (非阻塞)
- `sem_post()`：对信号量的 V 操作
- `sem_getvalue()`：读取信号量的值

参考链接：[POSIX Semaphores \(villanova.edu\)](https://www.villanova.edu/~cs/380/posix/semaphores/)

任务要求

线程

- 至少实现以上列出的所有函数，支持线程的创建、撤销、等待、终止操作
- 实现**全用户地址空间**的共享，同一进程的不同线程可以相互访问栈内数据（即不同线程中运行栈的地址不能重合）
- 可以保留少量不在线程间共享的地址空间，以记录当前线程的相关状态（也可实现 POSIX 标准规定的相关接口）

信号量

- 至少实现以上列出的所有函数，完整支持无名信号量

说明

本任务要求结合 POSIX 标准，以及 **MOS 操作系统中的已有机制的实现方法**，在系统中支持线程和信号量两种机制。具体的实现细节不必完全遵循已有的方法，只需正确实现各接口函数的行为即可。评判时会从实现方式、实现效果和工作量几个方面检查。

- 至少实现上述的两组函数，满足所有任务要求
- 可能需要合理扩展系统调用以及相关数据结构
- 实现的其他 POSIX 线程和信号量相关接口会根据难度和效果加分

关于以上 POSIX/Linux 接口的详细行为，建议通过以下方式查询 man pages:

- <https://man7.org/linux/man-pages/>
- <https://linux.die.net/man/>
- 在实验环境中使用 `man` 命令

提交要求

- 请自行建立 lab4-challenge 分支，在该分支完成代码后，push 到个人的远程仓库。代码内需要包含：对于功能的详细测试程序，测试程序本身及运行测试程序得到的运行结果；且应具有足够的可读性。

```
1 git checkout lab4 # 可以自行选择合适的主分支
2 git add .
3 git commit -m "xxxxx"
4 git checkout -b lab4-challenge # 基于所选择的主分支，建立lab4-challenge分支
5 # 完成代码并提交到远程仓库
6 git push origin lab4-challenge:lab4-challenge
```

- 实验报告请提交至 spoc 系统，在书写**实验报告**和准备**申优答辩**时，请加入以下内容：
 - 对于任务的**实现思路**，并配合关键代码进行说明
 - 对于功能的**详细测试程序**，以及运行测试程序得到的**运行结果**
 - 完成挑战性任务过程中**遇到的问题及解决方案**