

操作系统 Lab 4

151242041, 王昊庭, hatsuyukiw@gmail.com

2017 年 5 月 29 日

1 实验内容

本次实验中主要实现了线程间的匿名信号量 (semaphore). 用户库提供 `sem_init`, `sem_wait`, `sem_post`, 和 `sem_destroy` 四个函数.

这四个函数的定义与 POSIX 一致, 不出错时与 POSIX 行为一致.

这个作者实现了简易线程, 通过 `fork_thread` 系统调用可以创建线程. 该系统调用与 `fork` 的行为基本一致, 但是对于子进程在拷贝父进程的内存内容时, 除栈空间进行深拷贝以外, 其他内存页只进行页目录与页表的映射. 因此线程有独立的栈, 但是共享其他内存空间.

这个作者编写了 Producer-Consumer 小程序检验信号量的实现正确性. 程序在 `game/producer_consumer.c` 中. 该程序输出至串口 (终端). Producer 生产从 1 至 100 的整数, 存入大小为 5 的缓存中, Consumer 从缓存中取出整数并输出.

2 问题

1. Q: 什么地方是所有进程共享的呢? 你可以据此实现你的进程匿名信号量么?
A: 内核所在的那个页目录是所有进程共享的. 可以据此实现匿名信号量, 但是实现非常不方便.
2. Q: 普通的全局变量是不是可以被用户修改?
A: 不能. 如果全局变量可以被两个线程访问, 那么可能出现竞争情形; 如果只有一个线程会访问这个全局变量, 那么它本身不应该作为全局变量出现.
3. Q: Linux 进程和线程下匿名信号量的实现有什么本质区别?
A: 这个作者认为没有本质区别, 所谓匿名信号量就是它的使用者需要事先约定好在哪个特定的内存地址访问这个信号量. 线程本身共享内存, 天然地适合使用匿名信号量, 而进程间则需要手动分配共享内存来存放共有的信号量. 总之, 匿名信号量工作的关键是共享内存.
4. Q: 结合你的 OS, 你该怎样实现进程/线程的具名信号量?
A: 在内核维护一个从具名信号量映射到进程的字典结构.