

同步互斥和 Linux 内核模块

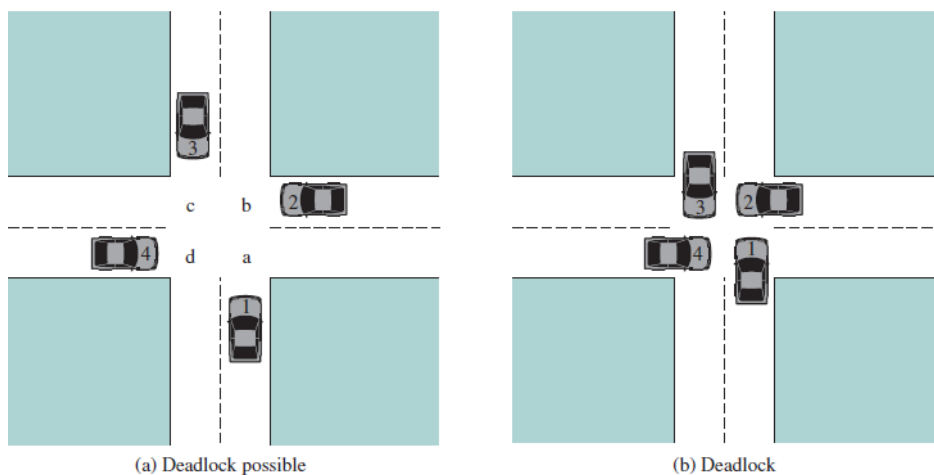
实验目的

- 学习使用 Linux 的系统调用和 pthread 线程库编写程序。
- 充分理解对共享变量的访问需要原子操作。
- 进一步理解、掌握操作系统进程和线程概念，进程或线程的同步与互斥。
- 学习编写多线程程序，掌握解决多线程的同步与互斥问题。
- 学习 Linux 模块的实现机理，掌握如何编写 Linux 模块。
- 通过对 Linux 系统中进程的遍历，进一步理解操作系统进程概念和进程结构。

实验内容

1. 有两条道路双向两个车道，即每条路每个方向只有一个车道，两条道路十字交叉。假设车辆只能向前直行，而不允许转弯和后退。如果有4辆车几乎同时到达这个十字路口，如图（a）所示；相互交叉地停下来，如图（b），此时4辆车都将不能继续向前，这是一个典型的死锁问题。从操作系统原理的资源分配观点，如果4辆车都想驶过十字路口，那么对资源的要求如下：

- 向北行驶的车 1 需要象限 a 和 b；
- 向西行驶的车 2 需要象限 b 和 c；
- 向南行驶的车 3 需要象限 c 和 d；
- 向东行驶的车 4 需要象限 d 和 a。



我们要实现十字路口交通的车辆同步问题，防止汽车在经过十字路口时产生死锁和饥饿。在我们的系统中，东西南北各个方向不断地有车辆经过十字路口（注意：不只有4辆），同一个方向的车辆依次排队通过十字路口。按照交通规则是右边车辆优先通行，如图(a)中，

若只有car1、car2、car3，那么车辆通过十字路口的顺序是car3->car2->car1。车辆通行总的规则：

- 1) 来自同一个方向多个车辆到达十字路口时，车辆靠右行驶，依次顺序通过；
- 2) 有多个方向的车辆同时到达十字路口时，按照右边车辆优先通行规则，除非该车在十字路口等待时收到一个立即通行的信号；
- 3) 避免产生死锁；
- 4) 避免产生饥饿；
- 5) 任何一个线程（车辆）不得采用单点调度策略；
- 6) 由于使用 AND 型信号量机制会使线程（车辆）并发度降低且引起不公平（部分线程饥饿），本题不得使用 AND 型信号量机制，即在上图中车辆不能要求同时满足两个象限才能顺利通过，如南方车辆不能同时判断 a 和 b 是否有空。

编写程序实现避免产生死锁和饥饿的车辆通过十字路口方案，并给出详细的设计方案，程序中要有详细的注释。

实验提示：

- 1) 每一辆车的行为设计为一个单独的线程。由于有 4 个不同方向的车辆，需要 4 种不同类型的线程。
- 2) 使用 pthread 的互斥锁和条件变量解决车辆的同步与互斥。
- 3) 对 4 个不同方向的车辆，要设置车辆队列条件变量如： queueNorth、queueEast、queueSouth、queueWest。比如说，当一辆车从北方来的时候已经在过十字路口，另一辆从北方驶来的车就要等在 queueNorth 队列中。每一个方向都需要一个计数器来跟踪等待排队的车辆数量。
- 4) 按照右边车辆优先通行规则，当一辆车在等待通过路口而它右边不断有车辆到达时，这辆车及这个方向车辆队列会导致饥饿。为了防止饥饿，我们要让刚刚通过路口的 A 车辆发一个信号给它左边等待的 B 车辆，接下去让 B 车辆通行。需要设置下次通行车辆的条件变量 firstNorth, firstEast, firstSouth, firstWest
- 5) 每一车辆到达十字路口时，要检测是否有死锁发生，当发生死锁时，死锁检测线程必须发出一种信号，例如：从北方来的车辆先行。
- 6) 假设我们设计的可执行程序文件名为 p1-1，可以用'e'、'w'、's'、'n'来标识东西南北 4 个方向驶来的车辆，程序 p1-1 运行时有如下显示(你的程序不一定是这样相同的输出)：

```
$ ./ p1-1 nsewwewn
car 4 from West arrives at crossing
car 2 from South arrives at crossing
car 1 from North arrives at crossing
car 3 from East arrives at crossing
DEADLOCK: car jam detected, signalling North to go
car 1 from North leaving crossing
car 3 from East leaving crossing
car 2 from South leaving crossing
car 4 from West leaving crossing
car 6 from East arrives at crossing
car 5 from West arrives at crossing
car 8 from North arrives at crossing
```

car 5 from West leaving crossing
car 6 from East leaving crossing
car 8 from North leaving crossing
car 7 from West arrives at crossing
car 7 from West leaving crossing

2. 编写一个Linux的内核模块，其功能是遍历操作系统所有进程。该内核模块输出系统中：每个进程的名字、进程pid、进程的状态、父进程的名字；以及统计系统中进程个数，包括统计系统中TASK_RUNNING、TASK_INTERRUPTIBLE、TASK_UNINTERRUPTIBLE、TASK_ZOMBIE、TASK_STOPPED等（还有其他状态）状态进程的个数。同时还需要编写一个用户态下执行的程序，显示内核模块输出的内容。
要求：程序中每行代码都要有注释

实验指导

1. Linux 线程创建和同步的编程参考资料：
 - 1) 操作系统原理教材“Operating System Concepts”，第4章的4.3.1节，第6章的“Project”。
 - 2) 短学期教材“Linux 程序设计”，浙江大学出版社
2. Linux 操作系统提供了 pthread 线程库，它是符合 POSIX 标准的函数库。线程控制方面的函数定义在 pthread.h 文件中，信号量控制方面的函数定义在 semaphore.h 文件中。
 - 1) 线程控制方面的函数有：pthread_attr_init、pthread_create、pthread_join、pthread_exit
 - 2) 互斥锁机制函数：pthread_mutex_init、pthread_mutex_lock、pthread_mutex_unlock、pthread_mutex_destroy
 - 3) 条件变量函数：pthread_cond_init、int pthread_cond_signal、int pthread_cond_wait、int pthread_cond_destroy、pthread_cond_broadcast
 - 4) 信号量线程控制函数：sem_init、sem_wait、sem_post、sem_getvalue、sem_destroy
3. 使用 gcc 编译器编译 C 语言程序。gcc 编译器的使用请参考“边干边学—Linux 内核指导”教材第7章。编译第1、2题的程序时，在 gcc 中加入选项 **-lpthread**，链接 pthread 线程库。
4. 第2题中，如何编写内核模块程序及编译、安装内核模块，可以参考“边干边学—Linux 内核指导”教材第13章。
5. 第2题中，每个进程的进程名字、pid、进程状态、父进程的指针等在 task_struct 结构的字段中。在内核中使用 printk 函数打印有关变量的值。遍历进程可以使用 next_task 宏，init_task 进程为0号进程。task_struct 结构参阅“边干边学—Linux 内核指导”教材11.2节；遍历进程方法可以参阅“边干边学—Linux 内核指导”教材11.6节。
6. 第2题中，用户态下的程序是从/var/log/kern.log (ubuntu)文件中读出内核模块输出的内容。

撰写实验报告的要求

1. 按照实验报告模板格式撰写实验报告。
2. 需要提交程序实现的设计思路，必要的设计文档，源程序及注释，程序运行结果截图。
3. 必须撰写实验讨论（即心得体会），内容为实验过程中遇到的问题及解决方法等。否则扣除本实验 20% 分数。
4. 实验提交的内容包括实验报告+源代码（以文本格式单独存放在一个或几个文件中），所有的文件打包后再上传到“作业系统”中。