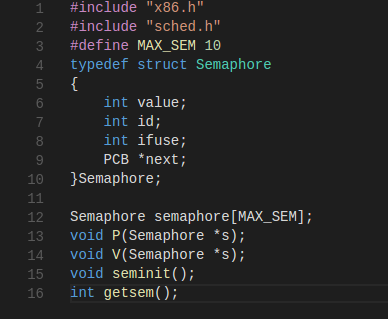
Lab4 进程同步

1. 在内核中定义semaphore结构体

在kernel/include中新建semaphore.h文件，文件内容为：



Value为信号量的值，id为信号量id，ifuse表示信号量是否正在使用，next为当前阻塞在该信号量上的进程链表。

定义了大小为10的信号量数组，等待用户使用。

定义操作P，V。

定义函数seminit，作用为初始化所有信号量。

定义函数getsem，作用为获取一个可用信号量id

1. 具体函数实现

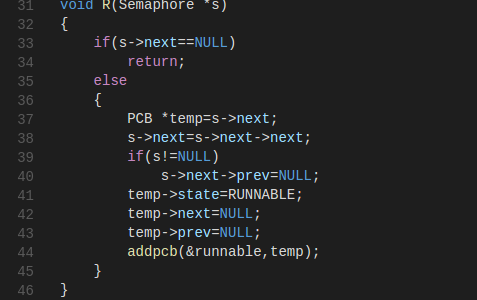
PV操作均在课程指导上。

P操作中调用的W：



作用为将当前进程阻塞在信号量s上

V操作调用的R函数：



作用为释放s信号量上阻塞的一个进程。

1. 实现系统调用

在lib/types.h中定义结构sem\_t：

struct sem\_t

{

int value;

int id;

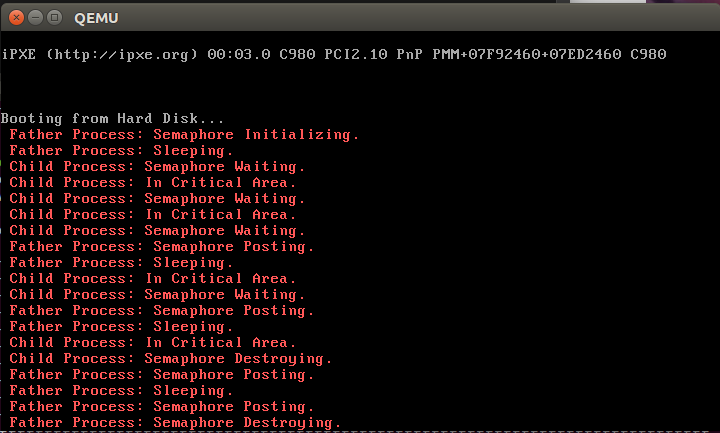
};

typedef struct sem\_t sem\_t;

分别调用syscall实现sem\_init,sem\_wait,sem\_post,sem\_destroy函数并在lib.h中声明。

在内核的irqHandle函数中实现上述系统调用。分别调用相关函数实现。

1. 实验结果：



Ubuntu版本：16.04

Gcc 版本：5.4.0 20160609

Qemu版本：

QEMU emulator version 2.5.0 (Debian 1:2.5+dfsg-5ubuntu10.14)