

oslab4 实验报告

姚荣春 151110079

1. gcc 版本及联系方式

Ubuntu 4.9.3-13ubuntu2

15895873878@163.com

2. 本次实现的功能

- (1) 本次要求的所有功能。
- (2) 增加了用户的静态库，所有用户程序都可以方便的使用这个库，用户文件夹变得简洁。

3. 测试方法

生产者消费者问题在 `test` 程序中演示，只需要 `make run` 就会自动运行 `test` 和 `game` 两个进程。如果不希望运行 `game`，只需要把 `151110079/src /kernel /kernel.c` 中的 `load_PCB(102400,"game");`注释掉即可。

3.实验心得

- (1) 创建线程的时候要完成参数的传递，因此要改写线程新分配的用户

栈。第一次实现的时候忘记 `lcr3` 获得新的页表，导致参数传错位置。

(2) 写着写着突然发现某个 `.c` 文件不见了，之前却全然不知。看来要定时 `make clean` 一下清除旧的 `.o` 来排除对编译的干扰。

4.思考题

(1) 回忆一下整个 `os` 实验，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？

内核代码及其映射是所有进程共享的，可以通过在进程的内核中开一个空间来实现进程的匿名信号量。也可以内核在用户区维护一个共享区来实现。

(2) 普通的全局变量是不是可以被用户修改？`linux` 进程和线程下匿名信号量的实现有什么本质区别？结合你的 `os`，你该怎样实现进程/线程的具名信号量？（可以纸上谈兵）

普通的全局变量是可以被用户修改的。`linux` 下进程和线程匿名信号量的实现的区别在于：因为线程共享代码段和数据段，因此线程的匿名信号量相当于线程的全局变量，占用进程的 `.data` 空间，而进程的信号量存放空间需要内核辅助创建（在内核中或单独的共享区中）。

线程/进程具名信号量：由内核开辟空间并维护（保存在内核中），通过进程所在组的/进程本身的权限来判定是否有权限对具名信号量进行操作。

5.makefile 使用介绍

`make run` 编译运行

`make gdb` 编译且打开 `gdb`

`make clean` 清除.o 等文件

`make run` 之后运行 `terminal.sh`(需要加权限) 可以连接 `qemu`

`src/usrlib` 下 `make lib` 编译用户库 (`make run` 也会编译 `lib`)