# 操作系统实验报告

171860695 梁宇方
计算机科学与技术系

## L0：amgame(2019/3/24)

**实验内容：**

编写了一个贪吃蛇小游戏。优化了游戏的运行逻辑。完成初始化后的正常运行过程中，每帧至多只需要绘制 3 个矩形。操作方法为 WASD 分别对应上左下右。

**发现问题：**

在 native 上运行时可能会出现无法响应的状况，此时可以按住 Ctrl 键配合 WASD 进行控制。

## L1：kernel(2019/4/4)

**实验内容：**

实现函数 kalloc()和 kfree()，并设计测试框架进行测试。

按照 first-fit 算法实现了关于 mem_block 的链表的管理。在 alloc 的内存都恰好 free 一次的情况下可以完美回到初始状态。源代码附带文档，并且存在足够的 assert 语句帮助理解逻辑。

测试模块嵌入在 os.c 文件内，为了实现并发数据结构增加了 lock.c 文件，所有 assert 语句在最后一次测试中均已成功。

**发现问题：**

若在获得锁失败后执行空语句可能会使虚拟机宕机。支持最多 4096 个内存块的分配。

## L2：kernel(2019/5/19)

**实验内容：**

实现 kmt 模块的一系列成员函数，包括线程的操作 create、teardown、自旋锁的操作 spinlock-lock、spinlock-unlock、信号量的操作 sem-wait、sem-signal 等。自行设计测试框架对代码进行测试。

使用 4-线程的生产者消费者模型进行测试，有较低的概率卡死。可以使用 tty 模块，并在上交的代码中保留了对 tty 模块的调用。

**发现问题：**

自旋锁并不是最原子的锁。自旋锁需要 pushcli 和 popcli，理论上来说应该只有对应的 cpu 可以改自己的 cli 的数值，但实践发现并非如此。于是创建了一种更原子锁用来保护自旋锁的数据。

tty 的加载稍微有点卡，需要十几秒的样子。有些代码比较适合放在 tty 的目录下因此对 tty.c 做了少许修改。

这个实验真坑，不过挺好玩。