

2021 操作系统实验（四）

本次实验重点在于掌握：进程的概念、操作系统的系统调用、PV 操作以及进程调度的实现。

1. 实现进程调度

参考《Orange's》第六章，在之前搭建的 *nasm+bochs* 实验平台上实现特定进程调度问题的模拟。

1.1 功能描述

- 添加系统调用，其接受 *int* 型参数 *milli_seconds*，调用此方法进程会在 *milli_seconds* 毫秒内不被分配时间片。
 - 注意，第六章代码已经在 *clock.c* 中有方法 *mills_delay*，这个方法仍然为进程分配了时间片，只不过进程进入空循环。
- 添加系统调用 **打印字符串**，接受 *char** 型参数 *str*
 - 注意，第六章代码已经在 *kliba.asm* 文件中有了 *disp_str* 函数显示字符串，但这是内核函数。请实现并包装成相应的系统调用。
- 添加两个系统调用执行 **信号量 PV 操作**，在此基础上模拟 **读者写者问题**。
 - 共有 6 个一直存在的进程（循环读写操作），A、B、C 为读者进程，D、E 为写者进程，F 为普通进程，其中
 - * A 阅读消耗 2 个时间片
 - * B、C 阅读消耗 3 个时间片
 - * D 写消耗 3 个时间片
 - * E 写消耗 4 个时间片
 - 读者在读的时候，写者不能写，必须等到全部读者读完
 - 同时只能一个作者在写
 - 在写的时候，读者不能读
 - 多个读者可以读一本书，但是不能太多，**上限数字**有 1、2、3，需要都能够支持，并且可以现场修改
 - A、B、C、D、E 进程需要 **彩色** 打印基本操作：读开始、写开始、读、写、读完成、写完成，以及对应进程名字
 - F 每隔 1 个时间片打印当前是读还是写，如果是读有多少人
 - 请分别实现 **读者优先** 和 **写者优先**，需要都能够支持，并且可以现场修改
 - 请想办法解决此问题中部分情况下的进程饿死问题（可参考第六章）

1.2 注意事项

- 使用 make 或类似工具构建整个项目。其中 makefile 必须支持 make run 命令直接启动，不需要其他命令。
- 本次作业可以直接在《orange's》源代码基础上完成，请记录下添加或者修改的地方。
- 请提交代码、Makefile、说明文档和截图。

1.3 评分标准

完成上述全部要求可以获得全部分数。

2. 问题清单

在整个实验过程中，无论是编程还是查资料，请同学们注意思考以下问题，助教检查时会从中随机抽取数个题目进行提问，根据现场作答给出分数。请注意，我们鼓励自己思考和动手实验，如果能够提供自己的思考结果并辅助以相应的实验结果进行说明，在分数评定上会酌情考虑。

1. 进程是什么
2. 进程表是什么
3. 进程栈是什么
4. 当寄存器的值已经被保存到进程表内，esp 应该指向何处来避免破坏进程表的值
5. tty 是什么
6. 不同的 tty 为什么输出不同的画面在同一个显示器上
7. 解释 tty 任务执行的过程
8. tty 结构体中大概包括哪些内容
9. console 结构体中大概包括哪些内容

3. 参考资料

- 《Orange's 一个操作系统的实现》