

2022年秋季操作系统实验（四）

实验任务（7分）

【读者写者问题】现有6个一直存在的进程执行循环读（写）操作，其中A是普通进程，B、C、D是读者进程，E、F是写者进程。

- B阅读消耗2个时间片，C、D阅读消耗3个时间片。
- E写消耗3个时间片，F写消耗4个时间片。
- 允许 n 个读者同时读一本书，有读者时写者不能写（ $n = 1, 2, 3$ ）。
- 只允许1个写者写，此时读者不能读。
- 读（写）完后休息 t 个时间片（ $t \geq 0$ ）。

具体要求

参考《Orange's 一个操作系统的实现》第六章，在之前搭建的NASM+Bochs实验平台上实现读者写者问题的模拟，具体要求如下：

- 阅读并理解书中已有的代码，尤其是和本次实验相关的代码，检查时也会有相关内容的提问。
- 添加一个系统调用，其功能是接受一个 `int` 型参数 `milli_seconds`，调用此系统调用的进程会在数 `milli_seconds` 毫秒内不被分配时间片。

第六章代码已经有 `mills_delay` 函数，但它仍为进程分配了时间片，进程进入空循环。

- 添加一个系统调用，其功能是接受一个 `char *` 型参数 `str`，打印出 `str`。

第六章代码已有 `disp_str` 函数，但这是内核函数。本次实验要求通过系统调用模式打印字符串。

- 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。普通进程A每个时间片输出每个读者写者的状态，格式为：`[序号][B][C][D][E][F]`，如 `1 0 0 0 X X`，每个状态用对应的符号加上对应的颜色表示。为了方便检查，只输出20次（序号从1~20）。

状态	符号	颜色
正在读	O	绿色
等待读	X	红色
正在写	O	绿色
等待写	X	红色
休息	Z	蓝色

- 同时读的数量 n 要求 $n = 1, 2, 3$ 均要实现，要求能够现场修改；读（写）完后休息的时间 $t(t \geq 0)$ 可自定，每个进程休息时间可不同，要求能够现场修改。
- 请分别实现读者优先和写者优先策略，要求能够现场修改。
- 请想办法解决此问题中部分情况下的进程饿死问题（不能通过调整读写后的休息时长来解决，即即便 $t = 0$ 时也要想办法解决）。

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 一个操作系统的实现》第六章源代码的基础上完成，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。

评分标准

本次不设置附加题，完成上述所有要求得7分。

实验问题（3分）

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若干题目进行提问，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？
10. 什么是时间片？
11. 结合实验代码解释什么是内核函数？什么是系统调用？