

操作系统实验（四）

南京大学软件学院

2015.5

实验重点

本次作业重点：操作系统的进程，系统调用，PV 操作以及进程调度的实现。

1 实验内容

1.1 实现进程调度

参考《Orange's》，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，具体要求如下。

- 添加一个系统调用 `sys_process_sleep`，其功能是接受一个 `int` 型参数 `mill_seconds`，调用此 System Call 的进程会在 `mill_seconds` 毫秒内不被进程调度函数分配时间片。
- 添加一个系统调用 `sys_disp_str`，其功能接受一个 `char* str` 参数，打印出字符串。
- 添加两个系统调用, `sys_sem_p` 和 `sys_sem_v`，即信号量的 PV 操作，在此基础上模拟睡眠的理发师问题。
- 共有五个进程，A 进程普通进程，B 进程是理发师，C 进程是顾客，D 进程是顾客，E 进程是顾客。要求有一把理发椅，并支持等待椅子的数目分别为 1、2、3（必须都能够支持，并且可以现场修改，助教检查时在其中随机选择数目），开始时理发师处于沉睡状态。理发师理发消耗两个时间片。
- 普通进程、理发师进程和顾客进程用不同颜色打印，其中顾客要打印递增的顾客 ID，并打印基本操作比如理发师剪发，顾客得到服务，顾客到来并等待，顾客离开等。

1.2 注意事项

- 第六章代码已经有 `sys_get_ticks` 系统调用和基于此的 `mills_delay` 函数, 似乎已经有了 `sleep` 的功能可以把进程延迟, 但它本质上还是给予分配了时间片的, 只不过在分配的时间片里在 `mills_delay` 函数中什么也没做。我们的 `sys_process_sleep` 是不分配时间片的。你可能有疑惑要是所有进程都 `sleep` 了时间片给谁了? 这确实是个问题, 如果四个进程都调用的 `sleep`, 在目前《Orange'S》的代码上要完美解决可能的改动很大, 有难度, 所以我们规定 A 进程是不调用 `sleep` 的 (相当于不可以被 `sleep` 的系统进程), 检查作业时不会要求 A 进程调用 `sleep`。
- 第六章的代码已经在 `kliba.asm` 文件中有了 `disp_str` 函数显示字符串, 但注意这是内核函数, 写在 `main.c` 中的 `testA, testB, testC` 能够调用只是因为它们虽然是用户进程但仍然写死在了内核中。所以本次实验要求加入一个系统调用, 通过系统调用模式打印字符串。
- 在阅读源代码时可能需要经常:
 1. 查看某段代码第一次在教材中出现的地点。
 2. 使用 `grep` 命令查找某个函数或变量在哪儿定义和哪儿使用到了。比如 `grep -r "some_function"`, 其中 `-r` 表示递归查找所有文件, 会有很详细的解释。
- 请在 TSS 上面提交你的实验代码, 并提交一个以你学号命名的 pdf 文档, 在其中说明你改动的部分和思路, 并将各个椅子数目下的实验结果截图插入其中。
- 按照往年经验, 此次实验完全自己完成的同学 (指**只参考了 Orange's 书, 没有参考任何学长学姐代码, 没有参考任何同学代码, 没有参考任何相同作业的资料**), 可以选择在 TSS 上的加分项目里提交你的所有源代码 (必须支持可以 `make run` 显示结果) 以及一个以你学号命名的 pdf 文档, 文档中只需要**概括叙述**为了完成实验, 需要参考 `oranges` 一书中的哪些内容 (具体到每一个小节), 可选的包括实验中遇到的困难和解决方案。这部分同学, **必须在特定的助教面前检查**, 并应对一切可能的问题和最高的标准, 一旦被认可并且通过复核, 将获得最高 2 倍的分数。即使不认可, 也不能在其他助教处检查作业, 以最高标准下的评分为准。

2 问题清单

在整个实验的过程中, 无论是编程还是查资料, 请各位同学注意思考以下问题, 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 为什么输出输出不同的画面在同一个显示器上?
7. 解释 tty 任务执行过程?
8. tty 结构体中大致包含哪些内容?
9. console 结构体中大致包含哪些内容?

3 参考资料

1. 《Orange'S: 一个操作系统的实现》