

《操作系统实现》上机实验

初识 Xinu

20201009

1. 在实验室机器（或自己的电脑，视未来实验环境决定）上的 Linux 系统中安装编译运行 xinu 操作系统
 - a) 在 `compile` 目录中执行 `make` 命令
 - b) 如果是首次编译，第一次 `make` 出错，可以再执行一次 `make` 即可
 - c) 修改 `include` 目录下头文件后，一般需要 `make clean`，再 `make`，才能将修改反应到可执行代码上
 - d) 编译成功后在 `compile` 目录生成 `xinu.elf` 文件
 - e) 执行命令 `qemu-system-i386 -kernel xinu.elf` 启动 QEMU 模拟器运行 xinu，按键 `Ctrl+Alt+3` 进入串口输入输出界面
2. 修改 `system/main.c`，将其中的代码修改为课件第一章“引论”第 34 页或第 36 页的并发执行示例代码，观察输出，判断在初始时间片配置下，一个时间片大约能输出多少个字符，并注意，一个时间片所输出的字符数目是否稳定
 - a) 时间片配置位于 `include/kernel.h`: `#define QUANTUM 2`，表示一个时间片为 2 毫秒
 - b) 初始化时间片设置在 `system/clkinit.c` 中，每个时间片到期后重置时间片并重新调度，在 `system/clkhandler.c` 中
 - c) 可修改其他代码使得观察输出结果更为容易
 - d) 在实验报告中给出观察结果

3. 完成如下要求

- a) 修改时间片长度为 10 毫秒
- b) 在第 4 章“调度和上下文切换”结束时提到, **ready** 的实现方式可能导致当前进程刚进入时间片, 但因为 **create/resume** 一个具有相同优先级的新进程导致重新调度, 请适当修改第 2 题中的代码(可根据需求修改其他部分的代码或数据结构), 观测是否存在此类现象
 - i. 将 **xinu** 完整代码打包, 文件名为 **xinu-1009-1**, 打包格式可为 **zip** 或 **tar.gz** 等
 - ii. 在实验报告中描述所作的修改(思想方法及改动的位置)以及实验结果(如运行结果的屏幕截图和说明)
- c) 如果上一题中存在所描述的现象, 请修改代码, 使得在优先级相同的情况下, 当前进程不会因为使某个进程变为就绪状态的操作, 在时间片尚未用完之前由于重新调度而失去 **CPU** 的控制权
 - i. 将 **xinu** 完整代码打包, 文件名为 **xinu-1009-2**, 打包格式可为 **zip** 或 **tar.gz** 等
 - ii. 在实验报告中描述所作的修改(思想方法及改动的位置)以及实验结果(如运行结果的屏幕截图和说明)

- ### 4. 最终提交一份实验报告和第 3 题中的两个压缩包文件, 整体打包成一个压缩包, 提交到 OBE 系统, 截止日期: 2020 年 10 月 16 日 0 点。