任务二：优先级调度器

中国科学院大学 操作系统研讨课

2017.10.19

1. 介绍

实现一个基于优先级的调度器，即在不同优先级下的进程执行时间应该不同。

2. 初始代码

2.1. 文件介绍

- Makefile: 编译文件。
- bootblock.s: 内核启动程序，**请使用作业一中自己写的代码**。
- Createimage.c: 生成内核镜像的Linux工具，**请使用作业一中自己写的代码**。
- entry.S: 中断和系统调用处理函数，使用任务一完成的代码
- kernel.c: 内核最先执行的文件，放在内核的起始处，使用任务一完成的代码。
- scheduler.c: 调度器，实现*task*的调度，这里需要实现一个基于优先级的调度器，另需实现两个系统调用函数do_getpriority()和do_setpriority()。
- syslib.S: 系统调用函数，进程通过系统调用进入内核，本次任务需要了解。
- interrupt.c: 系统调用和中断处理相关的函数。
- queue.c: 队列处理函数，提供了队列操作的一些接口。
- print*.c：提供一些输出函数，可以用于调试以及显示信息。请在本任务开始前了解该文件。

- sync.c: 一些同步操作。
- util.c: 提供了一些输出函数，可以用于调试以及显示信息。请在本任务开始前了解该文件。
- settest: 设置需要测试的样例。
- test_*文件夹：针对不同任务提供不同的测试，本次作业需要测试test_preemp和test_blocksleep。
- *.h: 相应.c/.S文件的头文件。

2.2. 获取：

课程网站。

2.3. 运行

Makefile 文件提供编译功能。

./settest test_XXX 设置测试对象以及编译

make 编译命令

make clean 对编译产生的文件进行清除

sudo dd if=image of=/dev/sdb 将产生的 image 写进 SD 卡中

在 minicom 中执行 *loadboot* 运行程序

3. 任务

3.1. 设计和评审

怎样设置一个基于优先级的任务调度？不同优先级的 task 占总系统运行比例不同。

3.2. 开发

需要在 scheduler.c 中实现 do_setpriority 和 do_getpriority 函数。

基于优先级的系统调度，可以在 scheduler.c 中的 scheduler 实现或在 queue.c 中实现（此处不限）。

3.2.1.注意事项

基于优先级的进程调度：不同优先级进程占系统总运行时间的比例不同。如优先级为 1 的进程占 70%，优先级为 2 的进程占 30%（只是举例，可以按照这个例子来写）。

4. 测试

测量基于优先级的进程调度：test_prempt, 此时需要修改测试数据，更改每个 task 的优先级，如果不同优先级最终的占用时间比例和你预设的相同，则正确。此处检查时需先说明自己优先级的设定值。

可以自己设计测试样例。