任务一：实现系统调用 spawn、kill、wait

中国科学院大学 操作系统研讨课

2017.11.15

1. 介绍

本任务的主要目的是实现三个系统调用：spawn、kill、wait，即 kernel.c 中的三个函数：do_spawn、do_kill、do_wait

1.1. 需要了解的部分

- spawn的含义与处理机制
- kill的含义与处理机制
- wait的含义与处理机制

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: 时钟中断处理函数。
- kernel.c: 内核最先执行的文件，放在内核的起始处，本任务需要在该文件中实现三个系统调用do_spawn、do_kill和do_wait。其中，entry_function里面调用了do_spawn(“init”)，启动init进程。init进程调用spawn系统调用进一步启动process1和process2。Makefile里面指定了各个进程的入口地址。
- scheduler.c: 调度器，实现*task*的调度。
- syslib.S: 系统调用函数
- syslib.c: 系统调用接口
- 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。
- mbox.c: 邮箱操作，本次任务暂不需要实现。
- 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。

- file.c: 将所有任务放在File的结构体中，供do_spawn查找使用
- ramdisk.c：提供操作File的一些接口
- settest: 设置需要测试的样例，并编译。
- test_*文件夹：针对不同任务提供不同的测试，本次任务需要测试test_spawn。
- *.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

帮助学生发现设计的错误，及时完成任务。学生需要对这次的作业进行全面考虑，在实现代码之前有清晰的思路。学生讲解设计思路时可以用不同的形式，如伪代码、流程图等，建议使用 PPT。

3.1.1.设计介绍

- spawn的含义是什么?spawn实现中主要完成哪些操作？
- kill实现中主要完成哪些操作？
- wait实现中主要完成哪些操作？

3.2. 开发

3.2.1.要求

本次任务主要是实现三个系统调用，分别是 do_spawn, do_kill 和 do_wait 。

3.2.2.注意事项

本作业中所有任务启动都要经过 do_spawn 函数来实现，因此在该函数中你需要：

- 根据提供的参数，即任务名字，在 File(files.c)结构体中找到对应名字的任务信息，根据这些信息，对任务进行初始化，即下面的步骤
- 初始化 PCB(pcb_t 数据结构可以根据需要修改)
- 将任务加入到就绪队列中

注意考虑当进程处于 sleeping 和 blocking 状态时的处理，如 blocking 状态的进程被 kill 时，保证其相关的 semaphore、condition、barrier 操作不受影响，相关设计思路需要在 design review 时回答。

4. 测试

已提供了 spawn 的测试样例 test_spawn。 kernel.c 的开始函数将创建 init 进程，并在 init 进程中创建了另外两个进程。

kill 和 wait 在任务三中进行测试。