任务 2：context switch 开销测量

中国科学院大学 操作系统研讨课

2017.09.27

1. 介绍

测量 *task（指代进程或者线程）*context switch 的开销。

1.1. 需要了解的部分

- context switch

2. 初始代码

2.1. 文件介绍

- Makefile: 编译文件。
- bootblock.s: 内核启动程序，使用任务一中实现的代码。
- Createimage.c: 生成内核镜像的Linux工具，使用任务一中实现的代码。
- entry_mips.S: 代码中*task* context switch时需要进行的操作，本文件也提供了获取时间函数。
- kernel.c: 内核最先执行的文件，放在内核的起始处。
- lock.c: 实现一个互斥锁，本次任务中不用修改。
- scheduler.c: 调度器，实现*task*的调度，本次任务中需要完成多个tasks轮流运行，以及yield和exit操作。
- syslib.S: 系统调用函数，进程通过系统调用进入内核。
- queue.c: 队列操作函数，在本次任务中，如果非抢占调度器用队列实现，可以直接使用该文件内函数。

- task.c: 将本次任务中所有*task*通过一个task_info的结构体引用。初始化*task*时就可以通过该文件提供的task数组，做本次任务时task数组的值不需要改变。完成不同任务时，请给task数组赋不同的task值，用于不同的测试目的。文件中的注释部分表明每一个任务需要测试的进程或者线程（将注释部分去掉就可以直接使用）。
- process1/2/3.c：定义了测试用例进程，可以分别用于不同的任务。
- th1/2/3/4.c：定义了测试用例内核线程，可以分别用于不同的任务。
- util.c: 提供了一些print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。
- *.h: 相应.c/.S文件的头文件。

2.2. 获取：

课程网站。

2.3. 运行

createimage 为提供的可执行文件，当 createimage.c 实现完成后，将 Makefile 中的 createimage 项去掉注释。

make 命令编译文件

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

分别介绍进程上下文和线程上下文经历的过程（函数，每个函数做了什么）。这个过程怎么测量？

3.2. 开发

本次任务涉及的有 thread4/5.c(th3.c)和 process3.c，需要在这些 *task* 中添加 yield,do_yield,exit 和 do_exit,以及使用 util.c 的 get_timer()函数测量 *task* 的时间，通过 print_str 和 print_int 函数输出 *task* context switch 时间。

get_timer()函数返回的为 CPU 执行的 cycle 数，CPU 的 cycle 数除以 MHZ 变量后返回的为执行的毫秒数。由于系统使用的为 32 位寄存器，所以在运行过程中 CPU cycle 可能会发生溢出情况，这里先不做考虑。

4. 测试

分别输出进程和线程 context switch 的开销。此处怎样输出不做要求。