

## Lab3 进程切换 实验报告

计科 161220071 李杨

实验目的：

实现简单的任务调度，实现基于时间中断信号进行的进程之间的切换的全过程。并提供系统使用 fork、sleep、exit

实验流程：

1：bootloader 从实模式进入保护模式，加载内核至内存，并跳转执行

2：内核初始化 idt,gdt,tss,串口，8259A

3：启动时钟源

4：加载用户程序至内存

5：初始化内核 IDLE 线程的进程控制块（process control block），初始化用户程序的进程控制块

6：切换至用户程序的内核堆栈，弹出用户程序的现场信息，返回用户态执行用户程序

核心函数：

首先要能够为系统提供 fork、sleep、exit 这三个操作的调用。

Fork 系统调用用于创建子进程，关键点在于父子进程之间的不断切换。内核需要为子进程分配一块独立的内存，将父进程的空间、用户态堆栈完全拷贝至子进程的内存中。为子进程分配独立的进程控制块，完成对子进程的进程控制块的设置。

Sleep 系统调用用于进程主动阻塞自身

内核需要将进程有 RUNNING 状态切换为 BLOCKED 状态, 设置该进程的 sleep 时间片, 切换运行其他的 RUNNING 状态的进程。

Exit 系统调用用于进程主动销毁自身

内核需要将进程有 RUNNING 状态切换为 dead 状态, 回收该进程的所有资源, 切换运行其他 RUNNABLE 状态的进程。

具体操作：

1 : BootLoader 从实模式进入保护模式并加载内核, 跳转执行。与实验二一致

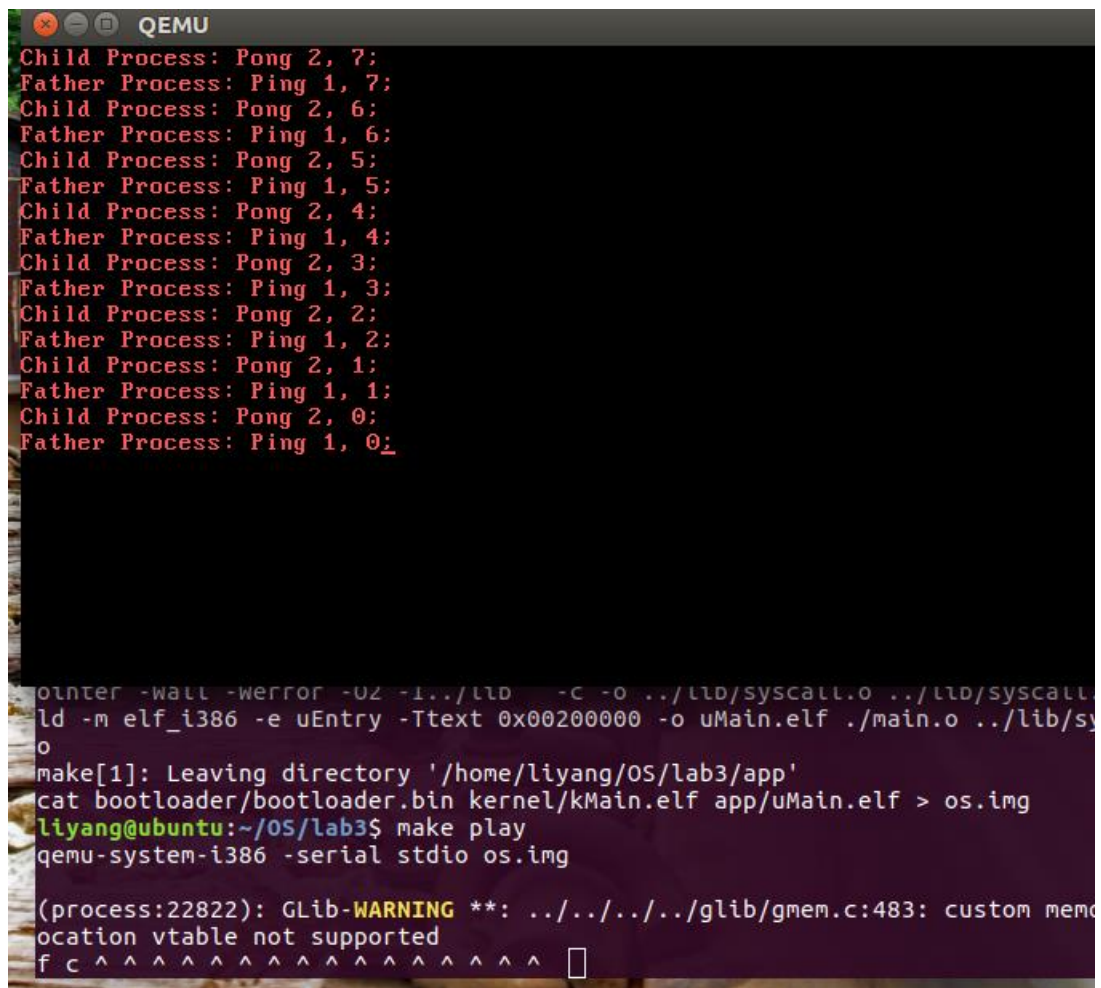
2 : 在实验二的基础上添加信号量为 0x20 的时间中断信号的处理机制, 用于进行父子进程间的切换。

3 : 在初始化完 idt,gdt,tss,串口,8259A 等一系列的存储器后, 调用 initTimer 函数引入时钟源, 并初始化 PCB 表项。Pcb 表是一个 struct ProcessTble 结构体类型的数组, 依次将 pcb[i].next 执行下一节点形成链, pcb 链表有三个指针节点组成, 分别为指向头结点的 pcb\_head, 指向当前节点的 pcb\_cur 和指向可用空间的 pcb\_free。

4 : 在加载用户程序至内存后, 与 lab2 不同的是, 程序会进入 IDLE 线程。先从 pcb 表项中获取一个空闲 pcb, 设置对应的段选择子, 将该进程的现场信息保存至 pcb 的 trapframe 中, 等待时钟中断信号的到来后返回用户程序执行代码, 代码将会调用 fork 函数创建子进程, 并不断通过时间中断信号在父子进程间切换, 并打印相对应的信息。这其中的 schedule 函数功能为找到 pcb 链表中第一个 RUNNABLE 进程, 依据 TSS 中记录的当前进程的 SS0 : ESP0, 从当前进程的用户堆

栈切换至内核堆栈，将现场信息压入栈中，切换至下一个进程，加载该进程的`内核栈`，弹出现场信息，切换至用户堆栈，返回执行用户程序。

实现结果：



```
QEMU
Child Process: Pong 2, 7;
Father Process: Ping 1, 7;
Child Process: Pong 2, 6;
Father Process: Ping 1, 6;
Child Process: Pong 2, 5;
Father Process: Ping 1, 5;
Child Process: Pong 2, 4;
Father Process: Ping 1, 4;
Child Process: Pong 2, 3;
Father Process: Ping 1, 3;
Child Process: Pong 2, 2;
Father Process: Ping 1, 2;
Child Process: Pong 2, 1;
Father Process: Ping 1, 1;
Child Process: Pong 2, 0;
Father Process: Ping 1, 0;

olinter -Wall -Werror -O2 -I../lib -c -o ../lib/syscall.o ../lib/syscall.c
ld -m elf_i386 -e uEntry -Ttext 0x00200000 -o uMain.elf ./main.o ../lib/syscall.o
make[1]: Leaving directory '/home/liyang/OS/lab3/app'
cat bootloader/bootloader.bin kernel/kMain.elf app/uMain.elf > os.img
liyang@ubuntu:~/OS/lab3$ make play
qemu-system-i386 -serial stdio os.img

(process:22822): GLib-WARNING **: ../../../../glib/gmem.c:483: custom memory allocation vtable not supported
f c ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
```