

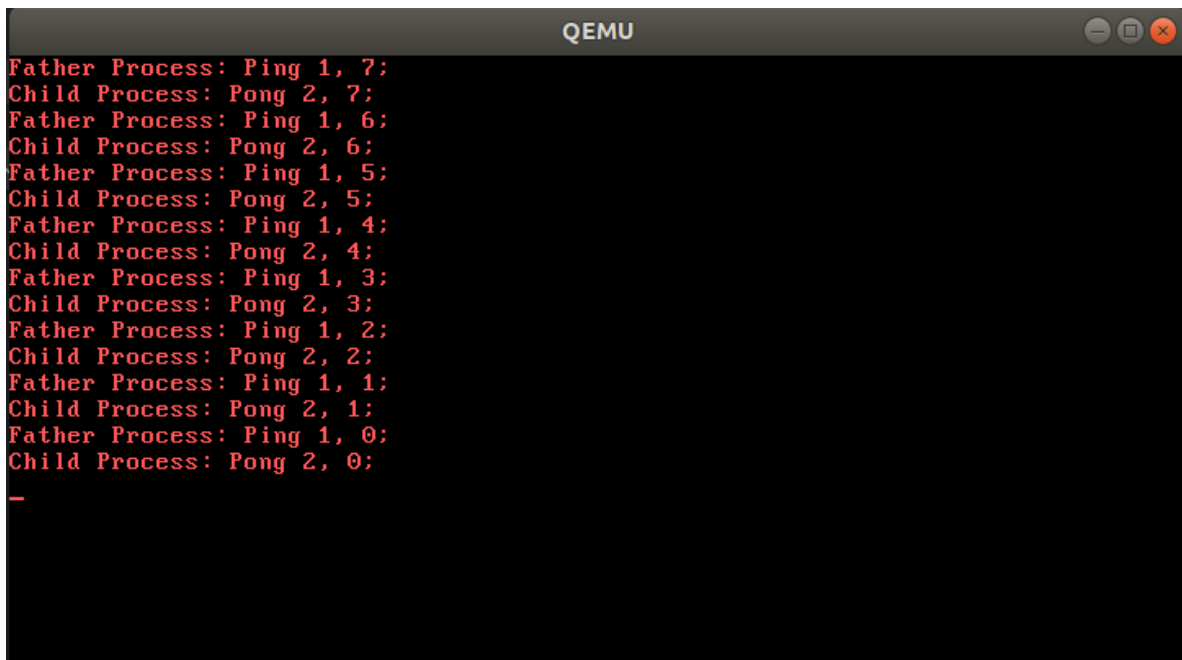
Lab3 进程切换

| 学号 | 姓名 | 邮箱 |
|-----------|----|--|
| 181860154 | 朱倩 | infinite0124@163.com |

一、实验进度

完成了所有必做内容和选做的中断嵌套

二、实验结果



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

```
QEMU
Father Process: Ping 1, 1;
Child Process: Pong 2, 1;
Father Process: Ping 1, 0;
Child Process: Pong 2, 0;
printf test begin...
the answer should be:
#####
Hello, welcome to OSlab! I'm the body of the game. Bootblock loads me to the memory position of 0x100000, and Makefile also tells me that I'm at the location of 0x100000. \x~!@#/(^&*()_+`1234567890-=..... Now I will test your printf: 1 + 1 = 2, 123 * 456 = 56088
0, -1, -2147483648, -1412567295, -32768, 102030
0, ffffffff, 80000000, abcdef01, ffff8000, 18e8e
#####
your answer:
=====
Hello, welcome to OSlab! I'm the body of the game. Bootblock loads me to the memory position of 0x100000, and Makefile also tells me that I'm at the location of 0x100000. \x~!@#/(^&*()_+`1234567890-=..... Now I will test your printf: 1 + 1 = 2, 123 * 456 = 56088
0, -1, -2147483648, -1412567295, -32768, 102030
0, ffffffff, 80000000, abcdef01, ffff8000, 18e8e
=====
Test end!!! Good luck!!!
```

在syscallFork中利用如下代码测试中断嵌套:

```
enableInterrupt();
    //copy data and code
    for(int i=0;i<0x100000;i++)
    {
        *((uint8_t *)((pos+1)*0x100000+i))=*((uint8_t *)
((current+1)*0x100000+i));
        asm volatile("int $0x20"); //xxx Testing irqTimer during syscall
    }
disableInterrupt();
```

三、实验修改的代码位置

1. lab3/lib/syscall.c:

```
pid_t fork()//调用syscallFork
int exec(const char *filename, char * const argv[])//调用syscallExec
int sleep(uint32_t time)//调用syscallsleep
int exit()//调用syscallExit
```

2. lab3/kernel/kernel/irqHandle.c:

```
void timerHandle(struct TrapFrame *tf)//实现时间中断功能（必要时进行进程切换）
void syscallFork(struct TrapFrame *tf)//完成fork功能
void syscallExec(struct TrapFrame *tf) //完成exec功能
void syscallsleep(struct TrapFrame *tf)//完成sleep功能
void syscallExit(struct TrapFrame *tf)//完成exec功能
```

3. lab3/kernel/kernel/kvm.c:

```
int loadElf(const char *filename, uint32_t physAddr, uint32_t *entry)
//实现加载elf文件到指定物理地址并返回文件入口的功能
```

