操作系统课程设计实验报告

学院：泰山学堂

专业：计算机取向

学号：201600301022

姓名：周玉昆

指导教师：杨兴强

同组者：苏柏瑞

助教：程鲁豫

实验目的：

阅读并分析 linux 0.11 源代码，对其中感兴趣的部分进行实验。
提取相关数据并可视化系统运行状态。

实验流程：

1-6 周：阅读相关源代码，并找出感兴趣的部分，分组进行实验。
7-12 周：提取数据，进行相关分析。
13-16 周：可视化数据，进行总结。

实验过程：

在与同组者苏柏瑞同学交流之后，我们确定了感兴趣的方向：进程调度。

经过协调，我们确定了分工：本人主要负责阅读、分析代码，提取相关数据，苏柏瑞同学主要负责数据的可视化。

下面是关于我的工作的简短介绍：

Linux 系统涉及到进程的代码很多，我们在这些代码中挑选了最具代表性的几个函数，进行了重点剖析。

1. 进程的创建：fork()

fork()函数位于汇编程序 system_call.s 中

```
.align 2
sys_fork:
 call find_empty_process
 testl %eax,%eax
 js 1f
 push %gs
 pushl %esi
 pushl %edi
 pushl %ebp
 pushl %eax
 call copy_process
 addl $20,%esp
1:  ret
```

可以看到，该函数会调用 fork.c 中的两个函数：find_empty_process()和 copy_process()，前者返回一个进程号以供后者使用，后者以当前进程为模板创建新的进程。

因此，我在 copy_process()函数中加入了断点，以显示当前状态下所有进程的状态。

2. 进程的调度：schedule()

该函数位于 sched.c 中，作用是进行进程调度。

在实验过程中，我们发现，该函数被调用的及其频繁，并且大多数的调用为进程 0 的空转。如果将所有信息都输出出来，信息量会很庞大且无用。

通过分析代码，我们发现在 schedule()的末尾，会调用 switch_to(next)函数，其中 next 为将要切换的进程号。由此，我们在输出信息之前，会进行一次判断，如果是切换到进程 0 执行（即系统空转），不对信息进行输出。

通过这种方法，大幅减少了获取的数据，提高了数据的质量。

3. 进程的结束：sys_exit()

该函数位于 exit.c 中，作用是结束进程。

该函数会调用 release()函数，删除进程数组中的相应指针并释放内存空间。

将断点设置在 release()函数释放相应进程之前，获取当前进程数组中的数据。

有关获取数据：

本次实验中使用了王天昊学长提供的 linux 实验平台，该平台可以使用在原程序中加入 log 函数实现断点并格式化输出想要的数据。

鉴于进程数据结构 task_struct 中的数据庞多，我们挑选了若干便于可视化的重要数据进行提取。

分别是：

Operation: 操作数
process No ： 进程数组序号
current: 执行该函数的当前进程 pid
state: 进程状态
pid: 进程 pid
father: 进程父亲 pid
counter: 时间片
priority: 优先级

其中，state 在 sched.h 中定义，分别为：

TASK_RUNNING	0
TASK_INTERRUPTIBLE	1
TASK_UNINTERRUPTIBLE	2
TASK_ZOMBIE	3
TASK_STOPPED	4

输出函数举例：

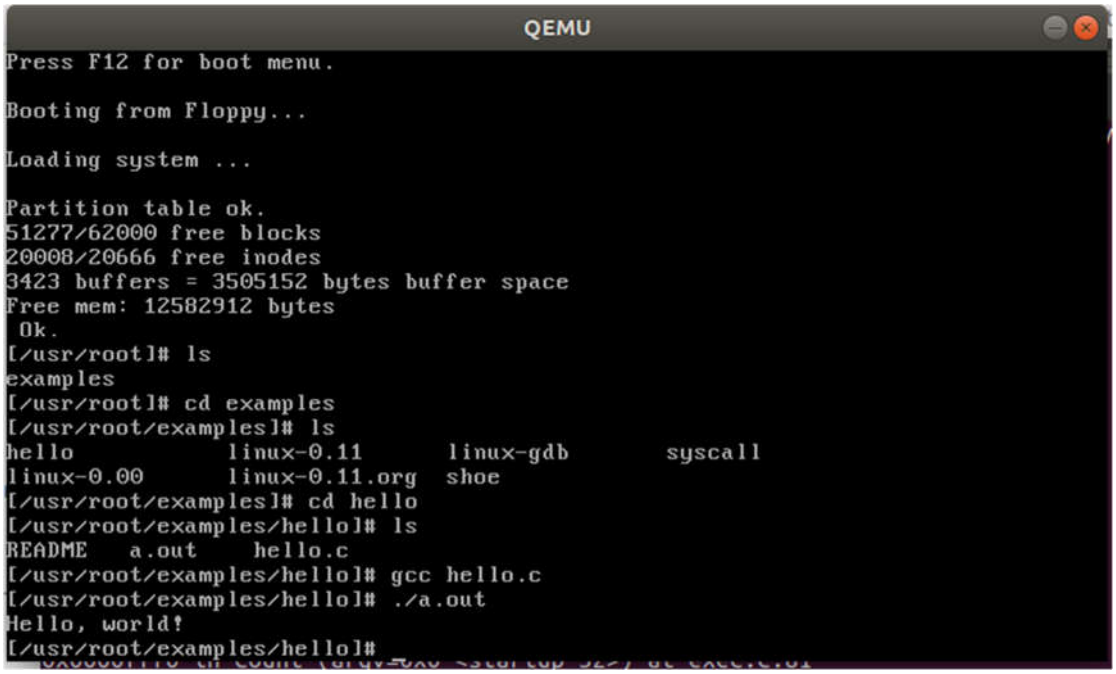
```
void printTasks5(int operation)
{
        int i;
        for(i=0 ; i<NR_TASKS ; i++){
                if(task[i]){
                        struct task_struct * process = task[i];
                        log("Operation: %d\n",operation);
                        log("process No. %d\n",i);
                        log("current: %d\n",current->pid);
                        log("state: %d\n",process->state);
                        log("pid: %d\n",process->pid);
                        log("father: %d\n",process->father);
                        log("counter: %d\n",process->counter);
                        log("priority: %d\n",process->priority);
                        log("#\n");
                }
        }

        log("$\n\n");
}
```

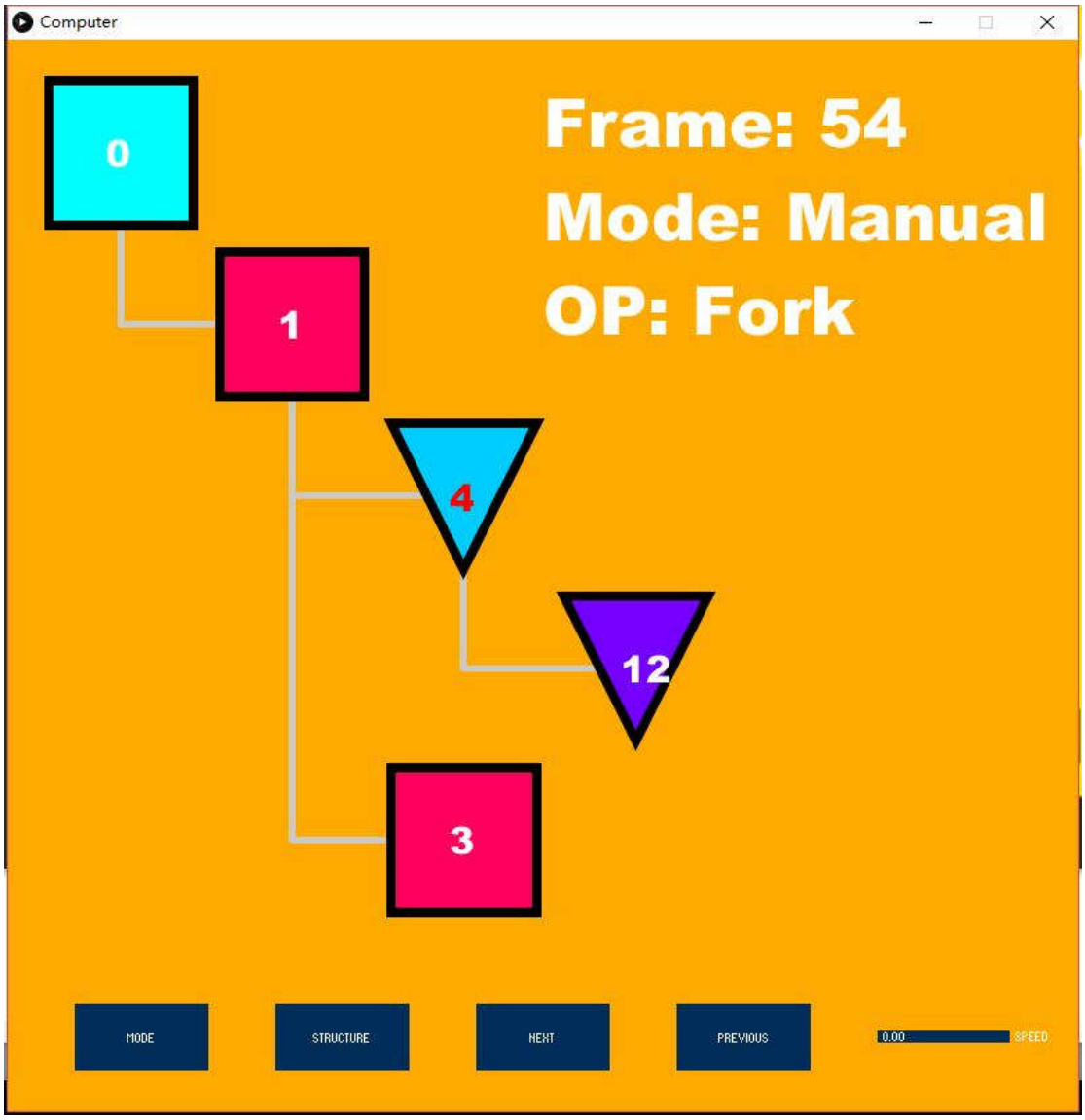
获取数据举例:

```
Operation: 3
process No. 0
current: 0
state: 0
pid: 0
father: -1
counter: 14
priority: 15
#
Operation: 3
process No. 1
current: 0
state: 0
pid: 1
father: 0
counter: 15
priority: 15
#
$
```

操作：切换目录到./examples/hello，使用 gcc 编译 hello.c 并执行。


QEMU
Press F12 for boot menu.
Booting from Floppy...
Loading system ...
Partition table ok.
51277/62000 free blocks
20008/20666 free inodes
3423 buffers = 3505152 bytes buffer space
Free mem: 12582912 bytes
Ok.
[/usr/root]# ls
examples
[/usr/root]# cd examples
[/usr/root/examples]# ls
hello linux-0.11 linux-gdb syscall
linux-0.00 linux-0.11.org shoe
[/usr/root/examples]# cd hello
[/usr/root/examples/hello]# ls
README a.out hello.c
[/usr/root/examples/hello]# gcc hello.c
[/usr/root/examples/hello]# ./a.out
Hello, world!
[/usr/root/examples/hello]#

可视化部分主要由苏柏瑞同学进行。
可视化样例:



实验收获：

通过阅读并分析 linux 源代码，增强了我对操作系统课程所学习的知识的理解。

在提取数据的过程中，遇到了一些小麻烦，也通过实验一一解决了，增强了我的动手能力。

和同伴一起完成实验，我们会给彼此提出意见和建议，遇到问题时也能一起商讨并解决，增强了我们的团队协作能力。