实验指南（第六次实验-机房环境）

首先打开 VMware，选择 Stu 虚拟机

Stu 用户密码：123。

打开终端：Ctrl+Alt+T。

注：机房环境下实验六需要搭建 MenuOS 环境

实验要求：（1）阅读理解 task_struct 数据结构；（2）分析 fork 函数对应的内核处理过程 sys_clone，理解创建一个新进程如何创建和修改 task_struct 数据结构；（3）使用 gdb 跟踪分析一个 fork 系统调用内核处理函数 sys_clone ，验证您对 Linux 系统创建一个新进程的理解。 特别关注新进程是从哪里开始执行的？为什么从那里能顺利执行下去？即执行起点与内核堆栈如何保证一致。

1. 搭建 MenuOS

所需文件：linux-3.18.6.tar.xz 、 menu.zip 放在 Tools 文件夹下。新建一个文件夹 LinuxKernel 并将文件拷贝到新文件夹中（使用 mkdir、cp 命令）。

解压并编译内核：

```
cd ~/LinuxKernel/
xz -d linux-3.18.6.tar.xz
tar -xvf linux-3.18.6.tar
cd linux-3.18.6
make i386_defconfig
make # 时间较长
```

重新配置编译 Linux 内核，使之携带调试信息

```
cd linux-3.18.6
make menuconfig  # 如果报错，执行 sudo apt-get install libncurses5-dev 安装缺失的包。需要联网
```

依次选择

```
kernel hacking
-> Compile-time checks and compiler options
[*]compile the kernel with debug info
```

（注：按空格选择，特别注意最后一条带有 * ）

然后保存（save）并退出。

```
make  # 时间较长
```

2. 给 MenuOS 增加 fork 命令

进入 menu 目录下，增加 fork 命令，并展示功能。参考书上 P119.

```
cd ~/LinuxKernel/
mkdir rootfs
unzip menu.zip(注:解压后名称如果为 menu-master，则重命名为 menu， 使用命令: mv menu-master menu)
cd menu
mv test_fork.c test.c （将 fork 增加到 MenuOS 中，参考 P119）
make rootfs
```

3. 使用 gdb 跟踪分析进程创建的过程

```
cd ~/LinuxKernel/
qemu -kernel linux-3.18.6/arch/x86/boot/bzImage -initrd rootfs.img -S -s
```

进入 gdb

```
gdb: （注：另开一个终端启动 gdb，同样需要进入 LinuxKernel 目录下）
  file linux-3.18.6/vmlinux
  target remote:1234
```

设置断点：

(要求设置多个断点：sys_clone->do_fork->copy_process->dup_task_struct->copy_thread->ret_from_fork，掌握这些函数主体功能和内部主要语句作用，并在实验报告中有所体现。参考 P119-120)

```
  break sys_clone
  break do_fork
  break dup_task_struct
  break copy_process
  break copy_thread
  break ret_from_fork
```

然后跟踪调试

设置断点：break(b)
单步跟踪：next(n)
显示语句上下文：list(l)
显示变量和寄存器：i r
跳过:jump(j)
推出:quit(q)

调试命令 (缩写)	作用
(gdb) break (b)	在源代码指定的某一行设置断点，其中xxx用于指定具体打断点位置
(gdb) run (r)	执行被调试的程序，其会自动在第一个断点处暂停执行。
(gdb) continue (c)	当程序在某一断点处停止后，用该指令可以继续执行，直至遇到断点或者程序结束。
(gdb) next (n)	令程序一行代码一行代码的执行。
(gdb) step（s）	如果有调用函数，进入调用的函数内部；否则，和 next 命令的功能一样。
(gdb) until (u) (gdb) until (u) location	当你厌倦了在一个循环体内单步跟踪时，单纯使用 until 命令，可以运行程序直到退出循环体。 until n 命令中，n 为某一行代码的行号，该命令会使程序运行至第 n 行代码处停止。
(gdb) print (p)	打印指定变量的值，其中 xxx 指的就是某一变量名。
(gdb) list (l)	显示源程序代码的内容，包括各行代码所在的行号。
(gdb) finish（fi）	结束当前正在执行的函数，并在跳出函数后暂停程序的执行。
(gdb) return（return）	结束当前调用函数并返回指定值，到上一层函数调用处停止程序执行。
(gdb) jump（j）	使程序从当前要执行的代码处，直接跳转到指定位置处继续执行后续的代码。
(gdb) quit (q)	终止调试。

实验楼环境下无需安装 MenuOS，直接给其增加命令即可。

首先删除实验楼下的 menu 目录

```
rm -rf menu
```

将群内的 menu.zip 文件上传并解压，之后步骤参考以上第 2、3 条说明。