CS307 Operating Systems

Project 1: Android Process Tree

Fan Wu and Bo Wang

Department of Computer Science and Engineering

Shanghai Jiao Tong University

Spring 2016

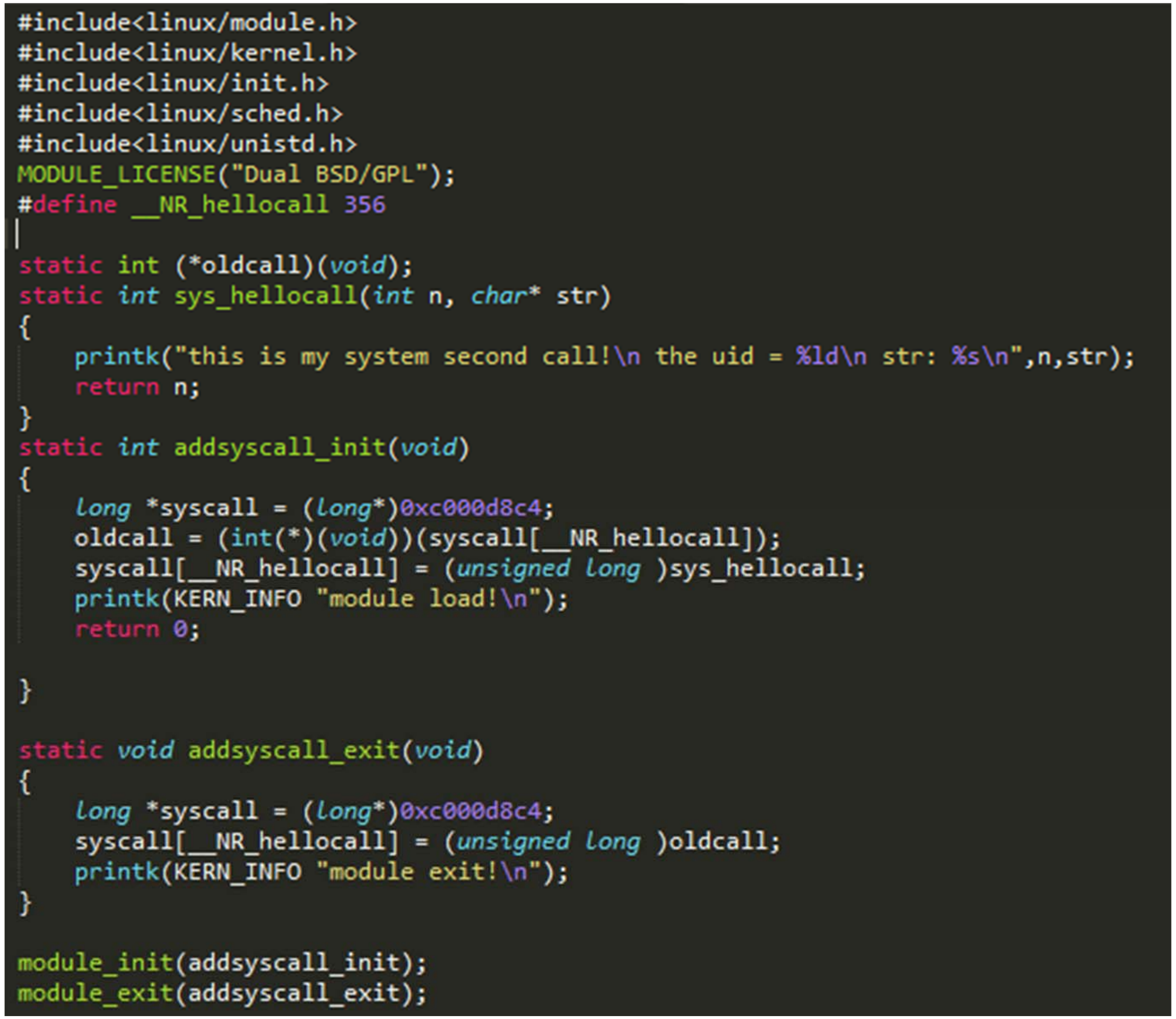
Problem 1

- Add system call dynamically.
- Use module.
- But the original android kernel does not support module.
- Compile a New One.
- Kernel is supported on website.
  - http://www.cs.sjtu.edu.cn/~fwu/teaching/res/android-kernel.tar.gz
  - Extract the kernel folder into the user folder.
- Linux Only

Start AVD

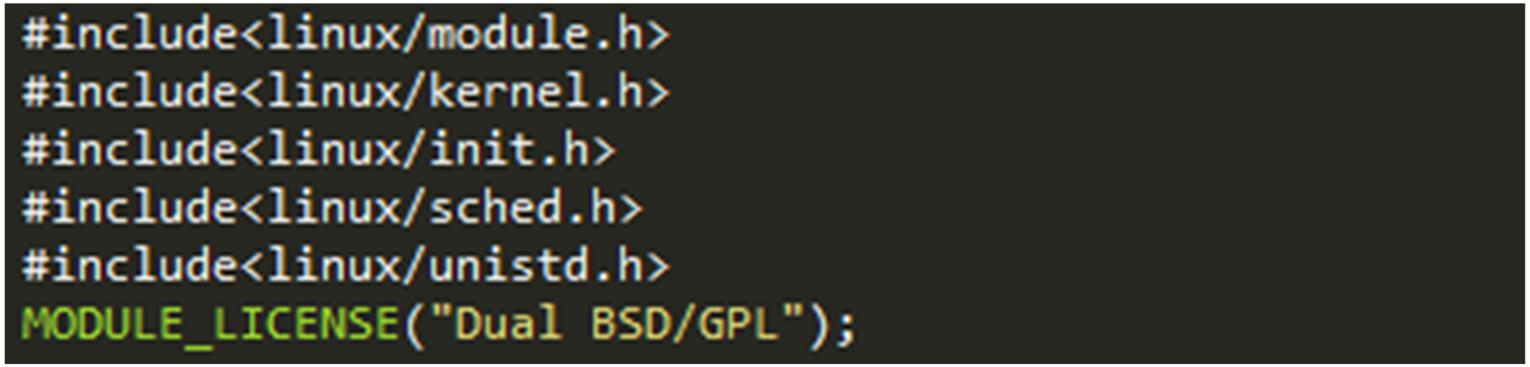
- We will start AVD with a new kernel.
  - emulator –avd YourAvdName –kernel KernelLocation –show-kernel
  - YourAvdName could be OsPrj
  - KernelLocation could be ~/kernel/goldfish/arch/arm/boot/zImage
  - -show-kernel makes kernel information shown in your shell.

Modules Source File



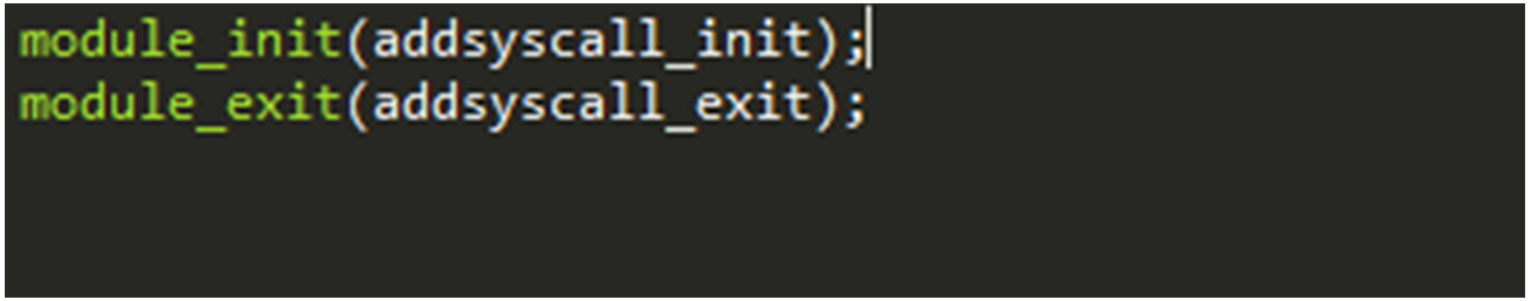
```
#include<linux/module.h>
#include<linux/kernel.h>
#include<linux/init.h>
#include<linux/sched.h>
#include<linux/unistd.h>
MODULE_LICENSE("Dual BSD/GPL");
#define __NR_hellocall 356

static int (*oldcall)(void);
static int sys_hellocall(int n, char* str)
{
    printk("this is my system second call!\n the uid = %ld\n str: %s\n",n,str);
    return n;
}
static int addsyscall_init(void)
{
    long *syscall = (long*)0xc000d8c4;
    oldcall = (int(*)(void))(syscall[__NR_hellocall]);
    syscall[__NR_hellocall] = (unsigned long )sys_hellocall;
    printk(KERN_INFO "module load!\n");
    return 0;

}

static void addsyscall_exit(void)
{
    long *syscall = (long*)0xc000d8c4;
    syscall[__NR_hellocall] = (unsigned long )oldcall;
    printk(KERN_INFO "module exit!\n");
}

module_init(addsyscall_init);
module_exit(addsyscall_exit);
```

Modules Source File - Definition



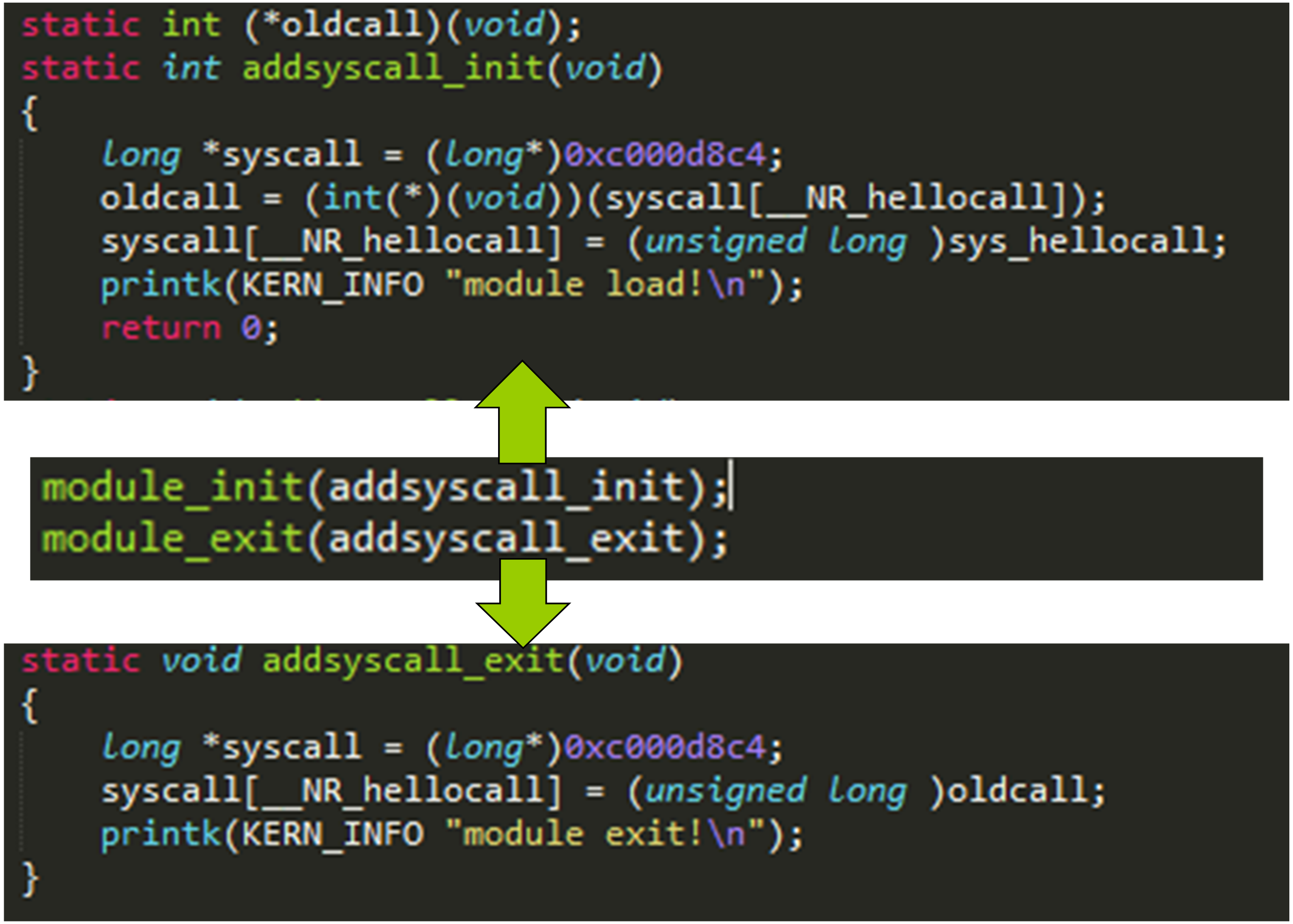
```
#include<linux/module.h>
#include<linux/kernel.h>
#include<linux/init.h>
#include<linux/sched.h>
#include<linux/unistd.h>
MODULE_LICENSE("Dual BSD/GPL");
```

- Properties of module. No need to change them



```
module_init(addsyscall_init);
module_exit(addsyscall_exit);
```

Modules Source File - Functions



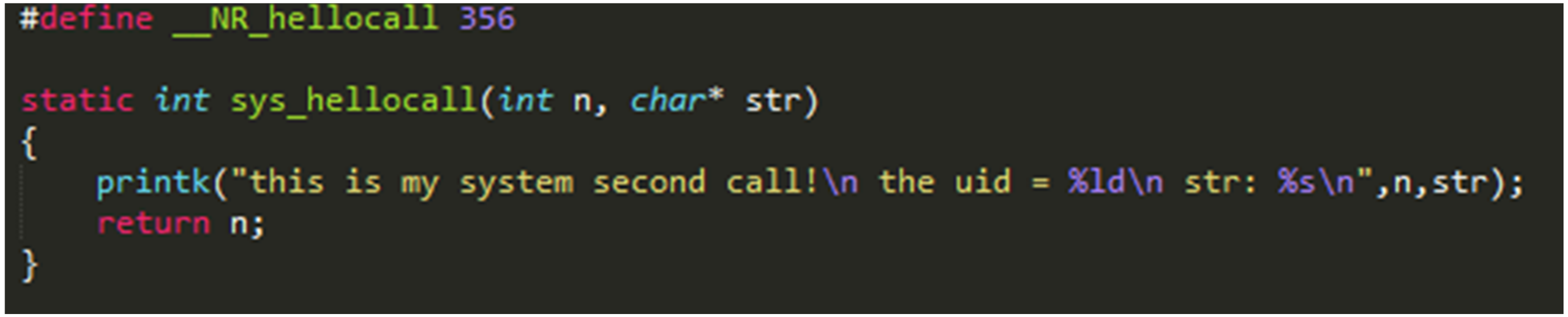
```
static int (*oldcall)(void);
static int addsyscall_init(void)
{
    long *syscall = (long*)0xc000d8c4;
    oldcall = (int(*)(void))(syscall[__NR_hellocall]);
    syscall[__NR_hellocall] = (unsigned long )sys_hellocall;
    printk(KERN_INFO "module load!\n");
    return 0;
}
```

```
module_init(addsyscall_init);
module_exit(addsyscall_exit);
```

```
static void addsyscall_exit(void)
{
    long *syscall = (long*)0xc000d8c4;
    syscall[__NR_hellocall] = (unsigned long )oldcall;
    printk(KERN_INFO "module exit!\n");
}
```

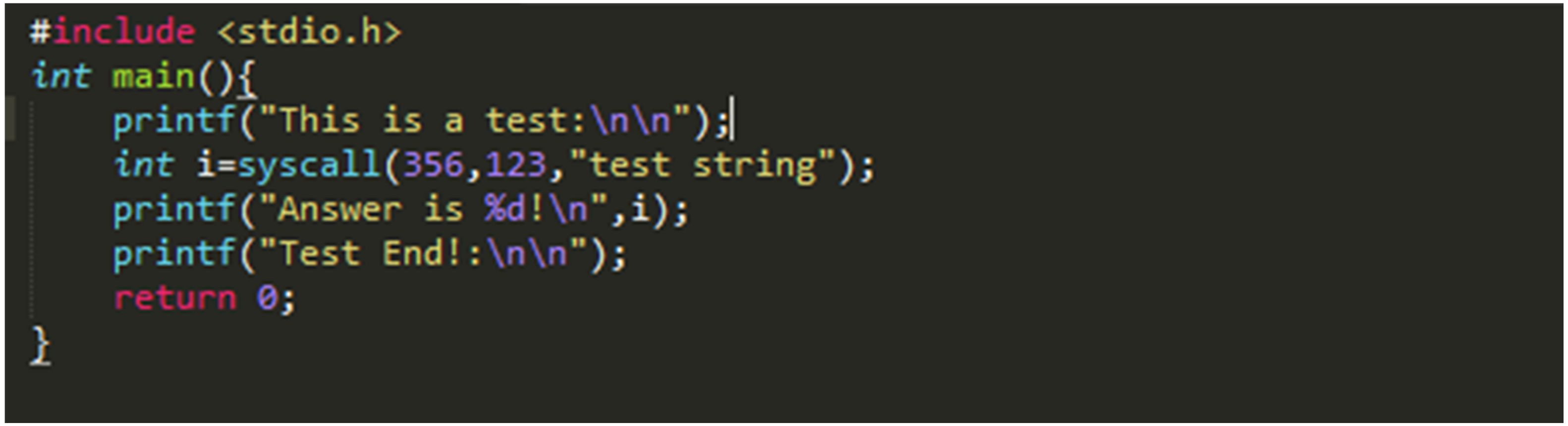
Modules Source File – System Call

- You should change this part to accomplish project.



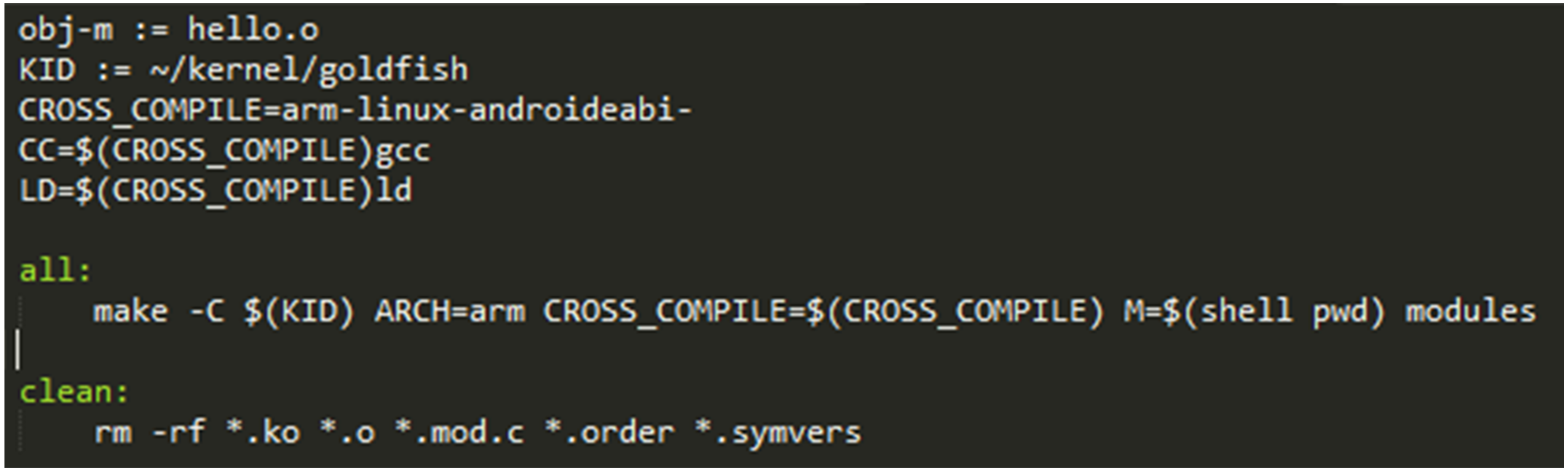
```
#define __NR_hellocall 356

static int sys_hellocall(int n, char* str)
{
    printk("this is my system second call!\n the uid = %ld\n str: %s\n",n,str);
    return n;
}
```

- Sample of using system call



```
#include <stdio.h>
int main(){
    printf("This is a test:\n\n");
    int i=syscall(356,123,"test string");
    printf("Answer is %d!\n",i);
    printf("Test End!:\n\n");
    return 0;
}
```

Modules Make File



```
obj-m := hello.o
KID := ~/kernel/goldfish
CROSS_COMPILE=arm-linux-androideabi-
CC=$(CROSS_COMPILE)gcc
LD=$(CROSS_COMPILE)ld

all:
    make -C $(KID) ARCH=arm CROSS_COMPILE=$(CROSS_COMPILE) M=$(shell pwd) modules

clean:
    rm -rf *.ko *.o *.mod.c *.order *.symvers
```

- Save source file and make file in one folder.
- KID is the location of your kernel.
- Add Environment Variable
  - #your ndk location#/toolchains/arm-linux-androideabi-4.9/prebuilt/linux-x86_64/bin
- Type make in shell in the folder.
- Then you will get a file *.ko, this is your module.

Use Module

- Upload your .ko file to avd
- Install mod
  - insmod *.ko
- Remove mod
  - rmmod *.ko
- List mod
  - lsmod
- Delete you .ko file before you want to update it.
- Remove the mod installed before you delete .ko file.

Some problem

- Apt-get 404 not found.
  - pls try again, the network is not stable.
- AVD is tooooooooo slow.
  - pls be patient.
- android avd can not work.
  - Use “ctrl+alt+t” instead of “ctrl+alt+F1”
- Adb usage

For Help?

- Teaching Assistant
  - Bo Wang
    - Email: wangbo0727@outlook.com, wangbo0727@126.com
    - WeChat：hadesghost727
  - Jiapeng Xie
    - Email: jsxiejp@163.com
    - WeChat: xjp18248794518

- Some useful website
  - http://www.csdn.net/
  - http://stackoverflow.com/
  - http://developer.android.com/

For Help?

Q&A