

## 实验五：在 arm linux 中增加系统调用

### 实验内容：

- 1、在 ARM-Linux 内核中加入一个系统调用，系统调用的功能随便你选。例如打印 姓名\_学号。然后写一个应用程序，调用该系统执行。该应用程序可以当内核启动之后执行，也可以采用开机自启动的方式执行。
- 2、提示：系统调用 中打印语句采 `printk`（内核打印）而非 `printf`。查看内核打印信息的方式为 `dmesg`，查看新增的系统调用能否被成功执行、
- 3、验收方式：每组的打印结果，至少包含（1）姓名学号 （2）系统调用号。每组验收时一人验收即可

实验报告：每人的实验报告中至少包含 **自己的姓名学号** 以及系统调用号。（每人不同）

### 实验要求：

- 1、进一步熟悉 Linux 系统操作
- 2、回顾上一次实验的自启动流程
- 3、掌握内核中增加系统调用的流程和步骤

### 主要步骤提示：

- 1、确保前三次实验能够完成，qemu 能够正常系统，文件系统能够创建、自启动流程掌握
- 2、Arm linux 中增加系统调用
  - （1）主要增加文件：arch/arm/kernel/mysyscall.c （自行创建）
  - （2）主要修改文件：arch/arm/kernel/Makefile，增加 mysyscall.o  
Arch/arm/kernel/calls.S，添加 CALL(sys\_helloworld)
  - （3）在 arch/arm/include/asm/unistd.h 修改系统调用的数量
- 3、重新编译内核、重新挂载根文件系统、启动……

结果示例：（可不同 需要有学号）

```
EXT3-fs (mmcblk0): write access will be enabled during recovery
kjournald starting. Commit interval 5 seconds
EXT3-fs (mmcblk0): recovery complete
EXT3-fs (mmcblk0): mounted filesystem with writeback data mode
VFS: Mounted root (ext3 filesystem) readonly on device 179:0.
Freeing unused kernel memory: 236K (805be000 - 805f9000)
random: nonblocking pool is initialized
start hello
hello world from syscall
end hello, the syscall number is 24

Please press Enter to activate this console. █
```