

lab5_debugfs 实验说明

本实验的目的：写一个内核模块，利用 debugfs 的 API 接口函数来创建新的节点，然后在节点里和用户空间进行数据交互。

基本实验步骤

1. 进入 `rlk_lab/rlk_basic/chapter_10/lab1` 目录。

```
# export ARCH=arm
# export CROSS_COMPILE=arm-linux-gnueabi-
# make BASEINCLUDE=/home/figo/work/runninglinuxkernel/runninglinuxkernel_4.0
```

这里 `BASEINCLUDE` 指定你当前 `runninglinuxkernel_4.0` 的目录路径。

然后把 `*.ko` 拷贝到 `runninglinuxkernel_4.0/kmodules` 目录下。

运行如下脚本启动 Qemu。

```
#cd runninglinuxkernel_4.0
# sh run.sh arm32 #启动虚拟机
```

在 Qemu 虚拟机:

```
#cd /mnt
# insmod debugfs-test.ko
```

```
/mnt # insmod debugfs-test.ko
I created benshushu on debugfs
```

创建的目录会在哪里呢？`debugfs` 一般会 mount 到 `/sys/kernel/debug/` 这个目录。

如果大家发现这个目录不存在，或者里面没有东西，说明 `debugfs` 没有挂载，或者输入“`df`”命令来查看。那么可以手工挂载。

```
mount -t debugfs none /sys/kernel/debug/
```

进入到 `/sys/kernel/debug/` 目录后，你会发现有一个 `benshushu` 的目录。

```

/sys/kernel/debug # ls
bdi                kprobes            suspend_stats
benshushu          memblock           tracing
clk                mmc0               ubi
extfrag            pm_qos             ubifs
fault_around_bytes regmap             usb
gpio               regulator          virtio-ports
hid                sleep_time         wakeup_sources

```

进入 benshushu 目录之后，你发现有一个 my_debug 的节点。

```

/sys/kernel/debug/benshushu # ls
my_debug
/sys/kernel/debug/benshushu # ls -l
total 0
-r--r--r--  1 0      0          0 Feb  3 13:56 my_debug
/sys/kernel/debug/benshushu # █

```

接下来就可以对这个节点进行读写操作了。

```

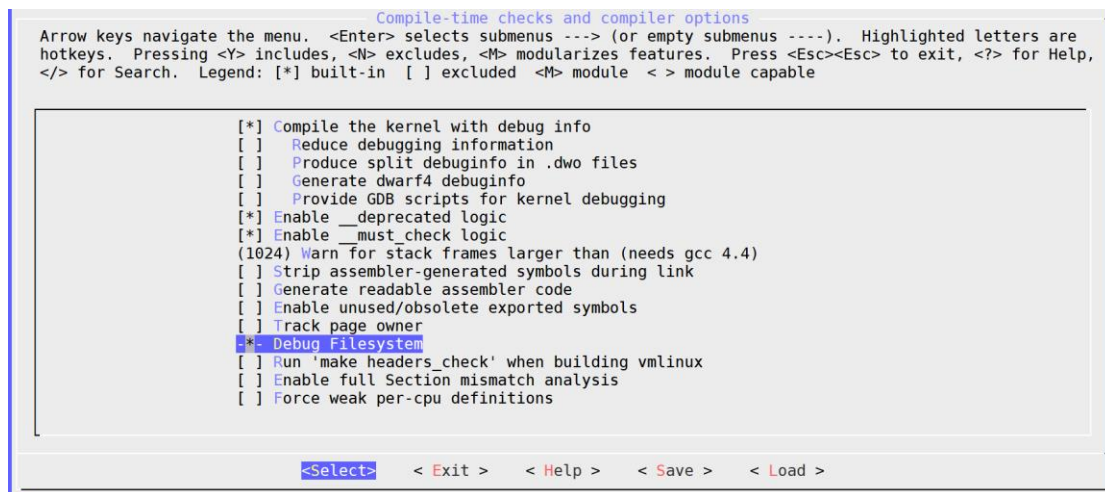
/sys/kernel/debug/benshushu # cat my_debug
100
/sys/kernel/debug/benshushu # echo 200 > my_debug
param has been set to 200
/sys/kernel/debug/benshushu # cat my_debug
200
/sys/kernel/debug/benshushu #

```

进阶思考

debugfs 是内核专门为调试为设置的一个虚拟文件系统，在现有的内核代码中，有很多调试功能都集成到 debugfs 中，比如前面讲到的 动态打印功能。还有很多设备驱动子系统的调试功能等等。

有很多功能需要在 config 文件中打开 CONFIG_DEBUG_FS=y 选项。



以 regmap 这个子系统为例，打开 drivers/base/regmap/Makefile 文件：

```
1 obj-$(CONFIG_REGMAP) += regmap.o regcache.o
2 obj-$(CONFIG_REGMAP) += regcache-rbtree.o regcache-lzo.o regcache-flat.o
3 obj-$(CONFIG_DEBUG_FS) += regmap-debugfs.o
4 obj-$(CONFIG_REGMAP_AC97) += regmap-ac97.o
5 obj-$(CONFIG_REGMAP_I2C) += regmap-i2c.o
6 obj-$(CONFIG_REGMAP_SPI) += regmap-spi.o
7 obj-$(CONFIG_REGMAP_SPMI) += regmap-spmi.o
8 obj-$(CONFIG_REGMAP_MMIO) += regmap-mmio.o
9 obj-$(CONFIG_REGMAP_IRQ) += regmap-irq.o
```

我们发现只要系统中定义了 CONFIG_DEBUG_FS，就会编译 regmap-debugfs.c 文件。系统运行之后，你会发现有一个 regmap 的目录。

```
/sys/kernel/debug # ls
bdi                kprobes            suspend_stats
benshushu          memblock            tracing
clk                mmc0                ubi
extfrag            pm_qos              ubifs
fault_around_bytes regmap              usb
gpio               regulator            virtio-ports
hid                sleep_time           wakeup_sources
/sys/kernel/debug #
```

进入 regmap 目录之后，你会发现里面有很多不同设备对应的调试 regmap 目录或者节点。

```

/sys/kernel/debug/regmap # ls
dcc:amp@0
dcc:amp@1
dcc:osc@0
dcc:osc@1
dcc:osc@2
dcc:power@0
dcc:power@1
dcc:volt@0
dcc:volt@1
dcc:volt@2
dcc:volt@3
dcc:volt@4
dcc:volt@5
smb:motherboard:mcc:osc@0
smb:motherboard:mcc:osc@1
smb:motherboard:mcc:osc@2
smb:motherboard:mcc:reboot@0
smb:motherboard:mcc:reset@0
smb:motherboard:mcc:shutdown@0
smb:motherboard:mcc:temp@0
smb:motherboard:mcc:volt@0
syscon.0.auto-sys_id
syscon.4.auto-sys_misc
syscon.5.auto-sys_procid
/sys/kernel/debug/regmap # █

```

如果大家对这些问题感兴趣,可以关注笨叔的旗舰篇视频,笨叔会在视频中和大家详细解答。

shop115683645.taobao.com



奔跑吧
LINUX社区

配套视频 **旗舰篇**

第1季

内存管理



旗舰篇一次订阅，持续更新

规划中	第二季	进程管理和调度 / 中断 / 锁 (已出)
第三季	虚拟化	
第四季	Linux 内核和应用开发调试必杀技	
第五季	红帽系列	

shop115683645.taobao.com

Linux视频课程



微信公众号：奔跑吧 linux 社区

1. > 一键订阅，持续更新
2. > 最有深度和广度的 Linux 视频
3. > 手把手解读 Linux 内核代码
4. > 紧跟 Linux 开源社区技术热点
5. > 笨叔叔的 VIP 私密群答疑
6. > 图书 + 视频，全新学习模式

微店：

微店



奔跑吧Linux内核



扫码识别

淘宝店：<https://shop115683645.taobao.com/>

微信公众号：

