

Stage 2 Report

GitHub ID: github.com/idrey

Repo URL: github.com/cicvedu/cicv-r4l-3-idrey.git

Date: 2024-03.25 .. 04.08

1. 编译Linux内核

a) 分析

按照文档进行操作即可

b) 实践

安装busybox

```
~/Program/cicv-r4l-3-idrey/busybox-1.36.1 git:(master)±97 (19.986s)
make install -j$(nproc)
./_install//usr/sbin/popmaildir -> ../../bin/busybox
./_install//usr/sbin/powertop -> ../../bin/busybox
./_install//usr/sbin/rdate -> ../../bin/busybox
./_install//usr/sbin/rdev -> ../../bin/busybox
./_install//usr/sbin/readahead -> ../../bin/busybox
./_install//usr/sbin/readprofile -> ../../bin/busybox
./_install//usr/sbin/remove-shell -> ../../bin/busybox
./_install//usr/sbin/rtcwake -> ../../bin/busybox
./_install//usr/sbin/seedrng -> ../../bin/busybox
./_install//usr/sbin/sendmail -> ../../bin/busybox
./_install//usr/sbin/setfont -> ../../bin/busybox
./_install//usr/sbin/setlogcons -> ../../bin/busybox
./_install//usr/sbin/svlogd -> ../../bin/busybox
./_install//usr/sbin/telnetd -> ../../bin/busybox
./_install//usr/sbin/tftpd -> ../../bin/busybox
./_install//usr/sbin/ubiaattach -> ../../bin/busybox
./_install//usr/sbin/ubidetach -> ../../bin/busybox
./_install//usr/sbin/ubimkvol -> ../../bin/busybox
./_install//usr/sbin/ubirename -> ../../bin/busybox
./_install//usr/sbin/ubirmvol -> ../../bin/busybox
./_install//usr/sbin/ubirsvol -> ../../bin/busybox
./_install//usr/sbin/ubiupdatevol -> ../../bin/busybox
./_install//usr/sbin/udhcpd -> ../../bin/busybox

-----
You will probably need to make your busybox binary
setuid root to ensure all configured applets will
work properly.
```

检查rust环境

```
~/Program/cicv-r4l-3-idrey/linux git:(master)±1379 (0.383s)
make LLVM=1 rustavailable
Rust is available!
```

进行内核配置

```

~/Program/cicv-r4l-3-idrey/linux git:(master)±1379 (3.156s)
make x86_64_defconfig
HOSTCC scripts/basic/fixedep
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/menu.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#

```

Kernel -> Space (Memory, Support (formerly) /etc, etc)

- [*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
- () Initramfs source file(s)
- [*] Support initial ramdisk/ramfs compressed using gzip
- [*] Support initial ramdisk/ramfs compressed using bzip2
- [*] Support initial ramdisk/ramfs compressed using LZMA
- [*] Support initial ramdisk/ramfs compressed using XZ
- [*] Support initial ramdisk/ramfs compressed using LZ0
- [*] Support initial ramdisk/ramfs compressed using LZ4
- [*] Support initial ramdisk/ramfs compressed using ZSTD
- [] Boot config support
- [*] Preserve cpio archive mtimes in initramfs
- Compiler optimization level (Optimize for performance (-O2)) --->
- [] Configure standard kernel features (expert users) --->
- [] Embedded system
- Kernel Performance Events And Counters --->
- [*] Profiling support
- [*] Rust support**

<Select> < Exit > < Help > < Save > < Load >

```

~/Program/cicv-r4l-3-idrey/linux git:(master)±1379 (55.449s)
make LLVM=1 menuconfig

```

```

HOSTCC scripts/basic/fixedep
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/menu.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
UPD scripts/kconfig/mconf-cfg
HOSTCC scripts/kconfig/mconf.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
HOSTCC scripts/kconfig/lxdialog/textbox.o
HOSTCC scripts/kconfig/lxdialog/util.o
HOSTCC scripts/kconfig/lxdialog/yesno.o
HOSTLD scripts/kconfig/mconf

```

```

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

```

开始编译内核

```
~/Program/cicv-r4l-3-idrey/linux git:(master)±1379 (6m 28.18s)
make LLVM=1 -j$(nproc)
  CC      arch/x86/boot/string.o
  LDS    arch/x86/boot/compressed/vmlinux.lds
  CC      arch/x86/boot/tty.o
  CC      arch/x86/boot/video.o
  CC      arch/x86/boot/video-mode.o
  AS      arch/x86/boot/compressed/kernel_info.o
  CC      arch/x86/boot/version.o
  CC      arch/x86/boot/video-vga.o
  CC      arch/x86/boot/video-vesa.o
  CC      arch/x86/boot/video-bios.o
HOSTCC  arch/x86/boot/tools/build
  AS      arch/x86/boot/compressed/head_64.o
CPUSTR  arch/x86/boot/cpustr.h
  CC      arch/x86/boot/cpu.o
VOFFSET  arch/x86/boot/compressed/../voffset.h
  CC      arch/x86/boot/compressed/string.o
  CC      arch/x86/boot/compressed/cmdline.o
  CC      arch/x86/boot/compressed/error.o
OBJCOPY  arch/x86/boot/compressed/vmlinux.bin
RELOCS   arch/x86/boot/compressed/vmlinux.relocs
HOSTCC  arch/x86/boot/compressed/mkpiggy
  CC      arch/x86/boot/compressed/cpuflags.o
  CC      arch/x86/boot/compressed/early_serial_console.o
  CC      arch/x86/boot/compressed/kaslr.o
  CC      arch/x86/boot/compressed/ident_map_64.o
  CC      arch/x86/boot/compressed/idt_64.o
  AS      arch/x86/boot/compressed/idt_handlers_64.o
  AS      arch/x86/boot/compressed/mem_encrypt.o
  CC      arch/x86/boot/compressed/pgtable_64.o
  CC      arch/x86/boot/compressed/acpi.o
  AS      arch/x86/boot/compressed/efi_thunk_64.o
  CC      arch/x86/boot/compressed/efl.o
  CC      arch/x86/boot/compressed/misc.o
GZIP     arch/x86/boot/compressed/vmlinux.bin.gz
MKPIGGY  arch/x86/boot/compressed/piggy.S
  AS      arch/x86/boot/compressed/piggy.o
  LD      arch/x86/boot/compressed/vmlinux
ZOFFSET  arch/x86/boot/zoffset.h
OBJCOPY  arch/x86/boot/vmlinux.bin
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
OBJCOPY  arch/x86/boot/setup.bin
BUILD    arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#1)
```

编译生成vmlinux文件

```
~/Program/cicv-r4l-3-idrey/linux git:(master)±1380 (0.05s)
ls
arch certs crypto fs ioURING Kconfig LICENSES mm modules.order README samples sound usr vmlinux.a
block COPYING Documentation include ipc kernel MAINTAINERS modules.builtin Module.symvers README.md scripts System.map virt vmlinux.o
built-in.a CREDITS drivers init Kbuild lib Makefile modules.builtin.modinfo net rust security tools vmlinux
```

2. 对Linux内核进行配置

a) 分析

按照文档进行操作即可

问题回答：

- 问题一：在Kbuild中的`obj-m := r4l_e1000_demo.o`所定义，`obj-m`意为编译成内核模块
- 问题二：通过Makefile中的`$(MAKE) -C $(KDIR) M=$$PWD`与内核代码产生联系

b) 实践

编译内核模块

```
~/Program/cicv-r4l-3-idrey/src_e1000 git:(master)±1379 (1.699s)
make LLVM=1 -j$(nproc)
make -C ..linux M=$PWD
make[1]: Entering directory '/home/zjt/Program/cicv-r4l-3-idrey/linux'
 RUSTC [M] /home/zjt/Program/cicv-r4l-3-idrey/src_e1000/r4l_e1000_demo.o
 MODPOST /home/zjt/Program/cicv-r4l-3-idrey/src_e1000/Module.symvers
  CC [M] /home/zjt/Program/cicv-r4l-3-idrey/src_e1000/r4l_e1000_demo.mod.o
   LD [M] /home/zjt/Program/cicv-r4l-3-idrey/src_e1000/r4l_e1000_demo.ko
make[1]: Leaving directory '/home/zjt/Program/cicv-r4l-3-idrey/linux'
```

启动qemu虚拟机，使用ifconfig查看网卡

```

~/Program/cicv-r4l-3-idrey/src_e1000 git:(master)±137
./build_image.sh
[ 2.053595] netconsole: network logging started
[ 2.252467] tsc: Refined TSC clocksource calibration: 3000.003 MHz
[ 2.254015] clocksource: tsc: mask: 0xfffffffffffffff max_cycles: 0x2b3e47e4533, max_idle_ns: 440795202126 ns
[ 2.255839] clocksource: Switched to clocksource tsc
[ 2.624951] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input3
[ 2.669714] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 2.686189] IP-Config: Complete:
[ 2.686343]   device=eth0, hwaddr=52:54:00:12:34:56, ipaddr=10.0.2.15, mask=255.255.255.0, gw=10.0.2.1
[ 2.686735]   host=10.0.2.15, domain=, nis-domain=(none)
[ 2.686897]   bootserver=255.255.255.255, rootserver=255.255.255.255, rootpath=
[ 2.691238] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[ 2.752832] modprobe (66) used greatest stack depth: 14272 bytes left
[ 2.764273] cfg80211: Loaded X.509 cert 'sforshee: 00b28df47ae9c ea7'
[ 2.766172] platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
[ 2.766901] cfg80211: failed to load regulatory.db
[ 2.768208] ALSA device list:
[ 2.768347]   No soundcards found.
[ 2.813872] Freeing unused kernel image (initmem) memory: 1324K
[ 2.814397] Write protecting the kernel read-only data: 24576K
[ 2.817318] Freeing unused kernel image (text/rodata gap) memory: 2032K
[ 2.818209] Freeing unused kernel image (rodata/data gap) memory: 824K
[ 2.973877] x86/mm: Checked W+X mappings: passed, no W+X pages found.
[ 2.974375] Run /sbin/init as init process
[ 2.988104] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 3.023576] mount (71) used greatest stack depth: 14160 bytes left
[ 3.143585] mdev (73) used greatest stack depth: 13960 bytes left

```

Please press Enter to activate this console.

```

~ # ifconfig
eth0      Link encap:Ethernet HWaddr 52:54:00:12:34:56
          inet addr:10.0.2.15 Brcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fe12:3456/64 Scope:Link
          inet6 addr: fec0::5054:ff:fe12:3456/64 Scope:Site
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:220 (220.0 B) TX bytes:672 (672.0 B)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

```

~ #

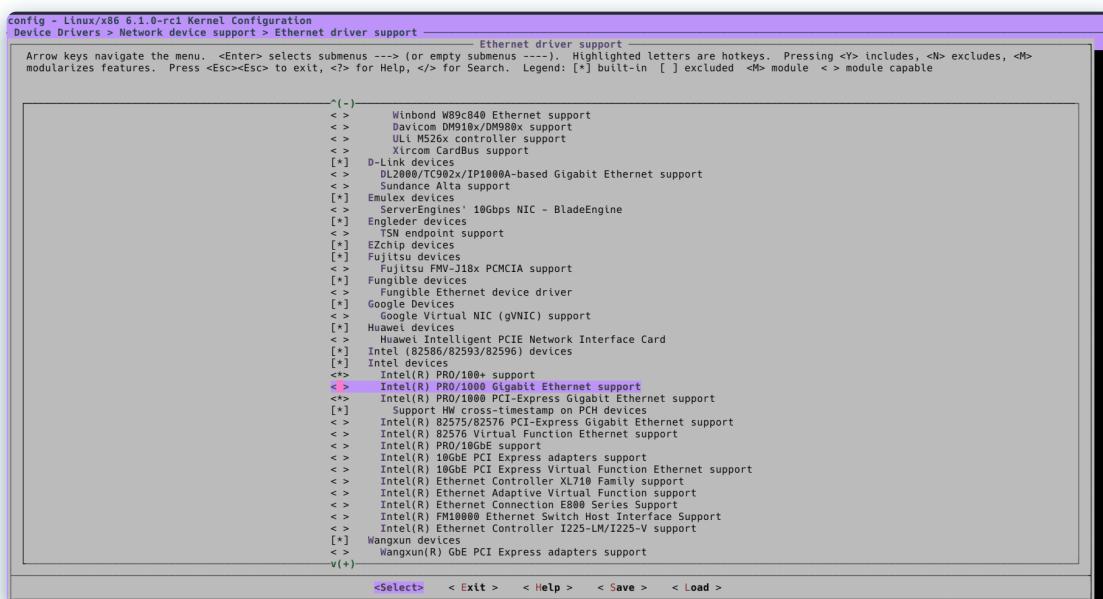
执行ping测试

```

~ # ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15): 56 data bytes
64 bytes from 10.0.2.15: seq=0 ttl=64 time=0.675 ms
64 bytes from 10.0.2.15: seq=1 ttl=64 time=1.095 ms
64 bytes from 10.0.2.15: seq=2 ttl=64 time=1.292 ms
64 bytes from 10.0.2.15: seq=3 ttl=64 time=1.044 ms
64 bytes from 10.0.2.15: seq=4 ttl=64 time=0.750 ms
64 bytes from 10.0.2.15: seq=5 ttl=64 time=1.001 ms
64 bytes from 10.0.2.15: seq=6 ttl=64 time=0.975 ms
64 bytes from 10.0.2.15: seq=7 ttl=64 time=1.244 ms
^C
--- 10.0.2.15 ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0.675/1.009/1.292 ms
~ #

```

禁用C版本网卡



重新编译内核

```
~/Program/cicv-r4l-3-idrey/linux git:(master)±1380 (25.371s)
make LLVM=1 -j$(nproc)
RUSTC L rust/kernel.o
EXPORTS rust/exports_kernel_generated.h
CC     rust/exports.o
AR     rust/built-in.a
AR     drivers/net/ethernet/intel/built-in.a
AR     drivers/net/ethernet/built-in.a
AR     drivers/net/built-in.a
AR     drivers/built-in.a
AR     built-in.a
AR     vmlinux.a
LD     vmlinux.o
OBJCOPY modules.builtin.modinfo
GEN     modules.builtin
MODPOST Module.symvers
UPD     include/generated/utsversion.h
CC     init/version-timestamp.o
LD     .tmp_vmlinux.kallsyms1
NM     .tmp_vmlinux.kallsyms1.syms
KSYMS   .tmp_vmlinux.kallsyms1.S
AS     .tmp_vmlinux.kallsyms1.S
LD     .tmp_vmlinux.kallsyms2
NM     .tmp_vmlinux.kallsyms2.syms
KSYMS   .tmp_vmlinux.kallsyms2.S
AS     .tmp_vmlinux.kallsyms2.S
LD     vmlinux
NM     System.map
SORTTAB vmlinux
CC     arch/x86/boot/version.o
VOFFSET arch/x86/boot/compressed/..voffset.h
OBJCOPY arch/x86/boot/compressed/vmlinux.bin
RELOCS  arch/x86/boot/compressed/vmlinux.relocs
CC     arch/x86/boot/compressed/kaslr.o
GZIP    arch/x86/boot/compressed/vmlinux.gz
CC     arch/x86/boot/compressed/misc.o
MKPIGGY arch/x86/boot/compressed/piggy.S
AS     arch/x86/boot/compressed/piggy.o
LD     arch/x86/boot/compressed/vmlinux
ZOFFSET arch/x86/boot/zoffset.h
OBJCOPY arch/x86/boot/vmlinux.bin
AS     arch/x86/boot/header.o
LD     arch/x86/boot/setup.elf
OBJCOPY arch/x86/boot/setup.bin
BUILD   arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#2)
```

重新进入qemu配置联网，

执行这个命令报错： ip: RTNETLINK answers: Invalid argument。将这个命令用 ifconfig eth0 broadcast 10.0.2.255 替代就好了

```
~/Program/cicv-r4l-3-idrey/src_e1000 git:(master)±1380
./build_image.sh
Please press Enter to activate this console.
~ # ls
bin          linuxrc      root        usr
dev          proc         sbin
etc          r4l_e1000_demo.ko sys
~ # insmod r4l_e1000_demo.ko
[ 43.180554] r4l_e1000 demo: loading out-of-tree module taints kernel.
[ 43.186549] r4l_e1000 demo: Rust for linux e1000 driver demo (init)
[ 43.187028] r4l_e1000 demo: Rust for linux e1000 driver demo (probe): None
[ 43.370562] ACPI: \_SB_.LNKC: Enabled at IRQ 11
[ 43.392654] r4l_e1000 demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 43.396219] insmod (81) used greatest stack depth: 11192 bytes left
~ # ip link set eth0 up
[ 57.435818] r4l_e1000 demo: Rust for linux e1000 driver demo (net device open)
[ 57.438848] r4l_e1000 demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 57.440056] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 57.444541] r4l_e1000 demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 57.451958] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=7, rdh=0
[ 57.452766] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 57.453148] r4l_e1000 demo: pending_irqs: 3
[ 57.453818] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
[ 57.919652] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=7, rdh=0
[ 57.921395] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 57.922238] r4l_e1000 demo: pending_irqs: 3
[ 57.922958] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
[ 58.380849] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=7, rdh=0
[ 58.382437] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 58.383211] r4l_e1000 demo: pending_irqs: 3
[ 58.383918] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
[ 58.960170] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=3, tdh=3, rdt=7, rdh=0
[ 58.961160] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 58.961337] r4l_e1000 demo: pending_irqs: 3
[ 58.961497] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
[ 58.962179] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=7, rdh=0
[ 58.962802] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 58.963040] r4l_e1000 demo: pending_irqs: 3
[ 58.963161] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
[ 59.268824] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=7, rdh=0
[ 59.270257] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 59.271018] r4l_e1000 demo: pending_irqs: 3
[ 59.271599] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
[ 62.795950] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=7, rdh=0
[ 62.796389] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 62.796618] r4l_e1000 demo: pending_irqs: 3
[ 62.796948] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
[ 70.219956] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=7, tdh=7, rdt=7, rdh=0
[ 70.221532] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 70.222338] r4l_e1000 demo: pending_irqs: 3
[ 70.223057] r4l_e1000 demo: Rust for linux e1000 driver demo (napi poll)
```

执行ping

```

~/Program/clicv-r4l-3-idrey/src_e1000 git:(master)±1380
./build_image.sh
~ # ip addr add broadcast 10.0.2.1 114.252059] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=7, rdh=1
[ 114.25315] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 114.254355] r4l_e1000_demo: pending_irqs: 3
[ 114.255074] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
~ # ip addr add broadcast 10.0.2.255 dev eth0
[ 135.053837] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 135.055442] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
ip: RTNETLINK answers: Invalid argument
~ # ip addr add 10.0.2.255/255.255.255.0 dev eth0
[ 172.499115] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 172.499558] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
~ # [ 173.132121] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=7, rdh=0
[ 173.133634] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 173.134507] r4l_e1000_demo: pending_irqs: 3
[ 173.135926] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)

~ # ip route add default via 10.0.2.1
~ # ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
[ 225.686970] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=3, tdh=3, rdt=7, rdh=0
[ 225.687375] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 225.687589] r4l_e1000_demo: pending_irqs: 131
[ 225.687841] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[ 225.689537] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=0, rdh=1
[ 225.689884] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 225.690047] r4l_e1000_demo: pending_irqs: 131
[ 225.690881] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=0 ttl=255 time=10.655 ms
[ 226.697108] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=1, rdh=2
[ 226.698785] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 226.699669] r4l_e1000_demo: pending_irqs: 131
[ 226.700506] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=1 ttl=255 time=6.194 ms
[ 227.703189] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=2, rdh=3
[ 227.704739] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 227.705524] r4l_e1000_demo: pending_irqs: 131
[ 227.706183] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=2 ttl=255 time=4.715 ms
[ 228.708769] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=7, tdh=7, rdt=3, rdh=4
[ 228.710261] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 228.711031] r4l_e1000_demo: pending_irqs: 131
[ 228.711695] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=3 ttl=255 time=4.748 ms
[ 229.714238] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=4, rdh=5
[ 229.715980] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 229.716852] r4l_e1000_demo: pending_irqs: 131
[ 229.717590] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=4 ttl=255 time=4.865 ms
[ 230.720017] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=5, rdh=6
~ /Program/clicv-r4l-3-idrey/src_e1000 git:(master)±1380
./build_image.sh
[ 229.744228] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=4, rdh=5
[ 229.715980] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 229.716852] r4l_e1000_demo: pending_irqs: 131
[ 229.717590] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=4 ttl=255 time=4.865 ms
[ 230.720017] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=5, rdh=6
[ 230.721703] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 230.722604] r4l_e1000_demo: pending_irqs: 131
[ 230.723470] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=5 ttl=255 time=5.608 ms
[ 231.726282] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=6, rdh=7
[ 231.727867] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 231.728708] r4l_e1000_demo: pending_irqs: 131
[ 231.729513] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=6 ttl=255 time=4.641 ms
[ 232.731925] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=3, tdh=3, rdt=7, rdh=0
[ 232.733551] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 232.734506] r4l_e1000_demo: pending_irqs: 131
[ 232.735226] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=7 ttl=255 time=4.966 ms
[ 233.737978] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=0, rdh=1
[ 233.739721] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 233.740600] r4l_e1000_demo: pending_irqs: 131
[ 233.741441] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=8 ttl=255 time=5.417 ms
[ 234.743588] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=1, rdh=2
[ 234.744032] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 234.744318] r4l_e1000_demo: pending_irqs: 131
[ 234.744554] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=9 ttl=255 time=1.527 ms
[ 235.746144] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=2, rdh=3
[ 235.747923] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 235.748542] r4l_e1000_demo: pending_irqs: 131
[ 235.748742] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=10 ttl=255 time=3.279 ms
[ 236.750723] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=7, tdh=7, rdt=3, rdh=4
[ 236.752366] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 236.753162] r4l_e1000_demo: pending_irqs: 131
[ 236.753966] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=11 ttl=255 time=4.839 ms
`-c
--- 10.0.2.2 ping statistics ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 1.527/5.121/10.655 ms
~ # [ 291.915810] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=4, rdh=5
[ 291.917305] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 291.918074] r4l_e1000_demo: pending_irqs: 3
[ 291.919149] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)

```

使用ifconfig查看网络，可以看到r4l网卡模块

```

--- 10.0.2.2 ping statistics ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 1.527/5.121/10.655 ms
~ # [ 291.915810] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=4, rdh=5
[ 291.917305] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 291.918074] r4l_e1000_demo: pending_irqs: 3
[ 291.919149] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)

~ # ifconfig
[ 323.123537] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
eth0      Link encap:Ethernet HWaddr 52:54:00:12:34:56
          inet addr:10.0.2.255 Bcast:0.0.0.0 Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fe12:3456/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
~ #

```

3. 编写HelloWorld模块

a) 分析

按照教程操作即可

相关配置：

- Kconfig

```
config SAMPLE_RUST_HELLOWORLD
    tristate "Print HelloWorld in Rust"
    help
        This option builds the Rust printing HelloWorld sample.

        To compile this as a module, choose M here:
        the module will be called rust_print.

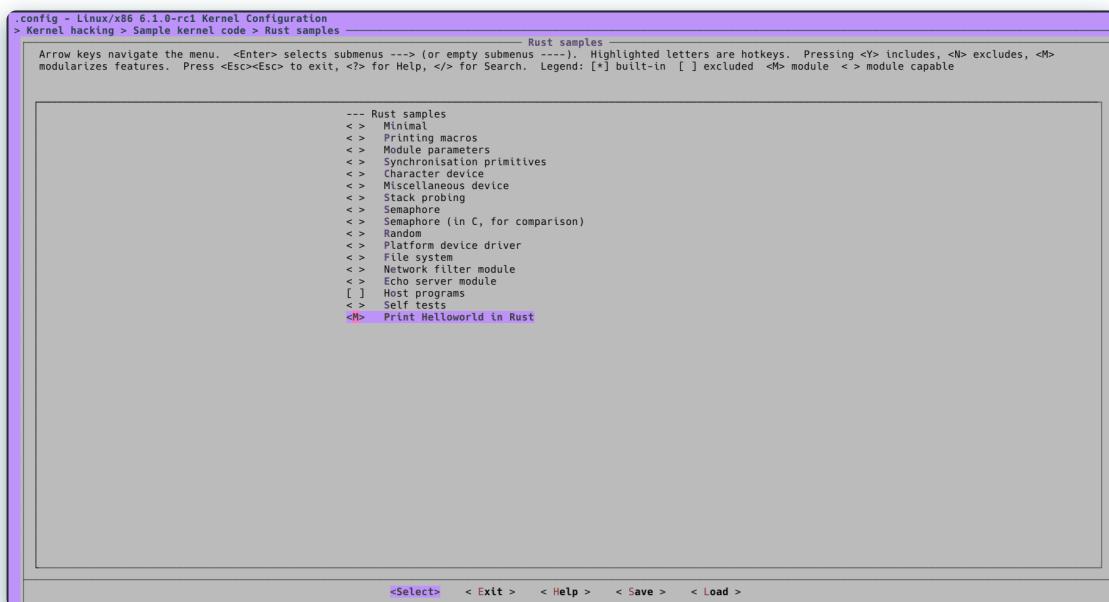
        If unsure, say N.
```

- Makefile

```
obj-$(CONFIG_SAMPLE_RUST_HELLOWORLD)      += rust_helloworld.o
```

b)

配置内核



编译内核

```

~/Program/cicv-r4l-3-ldney/linux git:(master)±1384 (16.426s)
make LLVM=1 -j$(nproc)
CC [M] samples/rust/rust_helloworld.mod.o
UPD include/generated/utsversion.h
CC init/version-timestamp.o
LD .tmp_vmlinux.kallsyms1
LD [M] samples/rust/rust_helloworld.ko
NM .tmp_vmlinux.kallsyms1.syms
KSYMS .tmp_vmlinux.kallsyms1.S
AS .tmp_vmlinux.kallsyms1.S
LD .tmp_vmlinux.kallsyms2
NM .tmp_vmlinux.kallsyms2.syms
KSYMS .tmp_vmlinux.kallsyms2.S
AS .tmp_vmlinux.kallsyms2.S
LD vmlinux
NM System.map
SORTTAB vmlinux
CC arch/x86/boot/version.o
VOFFSET arch/x86/boot/compressed/../voffset.h
OBJCOPY arch/x86/boot/compressed/vmlinux.bin
RELOCS arch/x86/boot/compressed/vmlinux.relocs
CC arch/x86/boot/compressed/kaslr.o
GZIP arch/x86/boot/compressed/vmlinux.bin.gz
CC arch/x86/boot/compressed/msc.o
MKPIGGY arch/x86/boot/compressed/piggy.S
AS arch/x86/boot/compressed/piggy.o
LD arch/x86/boot/compressed/vmlinux
ZOFFSET arch/x86/boot/zoffset.h
OBJCOPY arch/x86/boot/vmlinux.bin
AS arch/x86/boot/header.o
LD arch/x86/boot/setup.elf
OBJCOPY arch/x86/boot/setup.bin
BUILD arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#5)

~/Program/cicv-r4l-3-ldney/linux/samples/rust git:(master)±1383 (0.063s)
ls
built-in.a Makefile      rust_echo_server.rs  rust_helloworld.mod  rust_helloworld.o  rust_misctdev.rs   rust_platform.rs
hostprogs modules.order  rust_fs.rs        rust_helloworld.mod.c rust_helloworld.rs  rust_module_parameters.rs  rust_print.rs
Kconfig    rust_chrdev.rs  rust_helloworld.ko  rust_helloworld.mod.o  rust_minimal.rs   rust_netfilter.rs  rust_random.rs

```

复制模块

```

~/Program/cicv-r4l-3-ldney/linux/samples/rust git:(master)±1383 (0.016s)
cp rust_helloworld.ko ../../src_e1000/rootfs/

```

安装模块

```

~/Program/cicv-r4l-3-ldney/src_e1000 git:(master)±1383
./build_image.sh
[ 1.453390] In-situ OAM (TOAM) with IPv6
[ 1.454821] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[ 1.458032] NET: Registered PF_PACKET protocol family
[ 1.460065] 9pnet: Installing 9P2000 support
[ 1.460770] Key type dns_resolver registered
[ 1.462682] IPI shorthand broadcast: enabled
[ 1.463146] sched_clock: Marking stable (1432181211, 30289624)-->(1478379219, -15908384)
[ 1.465123] registered taskstats version 1
[ 1.465282] Loading compiled-in X.509 certificates
[ 1.473660] cryptomgr_test (44) used greatest stack depth: 15584 bytes left
[ 1.480691] PM: Magic number: 0:62:886
[ 1.481671] printk: console [netcon0] enabled
[ 1.481925] netconsole: network logging started
[ 1.555036] ata2: found unknown device (class 0)
[ 1.575997] ata2.00: ATAPI: QEMU DVD-ROM, 2.5+, max UDMA/100
[ 1.597178] scsi 1:0:0:0: [sr0] QEMU DVD-ROM 2.5+ PQ: 0 ANSI: 5
[ 1.633933] sr 1:0:0:0: [sr0] scsi3-mmc drive: 4x/4x cd/rw xa/form2 tray
[ 1.634566] cdrom: Uniform CD-ROM driver Revision: 3.20
[ 1.653120] sr 1:0:0:0: Attached scsi generic sg0 type 5
[ 2.040845] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input3
[ 2.192618] tsc: Refined TSC clocksource calibration: 2999.939 MHz
[ 2.193999] clocksource: tsc: mask: 0xffffffffffff max_cycles: 0x2b3e0ba14de, max_idle_ns: 440795216487 ns
[ 2.195797] clocksource: Switched to clocksource tsc
[ 14.352955] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[ 14.408664] modprobe (67) used greatest stack depth: 14272 bytes left
[ 14.425655] cfg80211: Loaded X.509 cert 'sforshee: 00b28df47ae9ceaf'
[ 14.428203] platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
[ 14.429279] cfg80211: failed to load regulatory.db
[ 14.431100] ALSA device list:
[ 14.431280] No soundcards found.
[ 14.482792] Freeing unused kernel image (initmem) memory: 1324K
[ 14.483645] Write protecting the kernel read-only data: 24576K
[ 14.487664] Freeing unused kernel image (text/rodata gap) memory: 2032K
[ 14.488983] Freeing unused kernel image (rodata/data gap) memory: 840K
[ 14.650194] x86/mm: Checked W-X mappings: passed, no W+X pages found.
[ 14.656677] Run sbin/init as init process
[ 14.686275] mount (72) used greatest stack depth: 14160 bytes left
[ 14.822343] mdev (74) used greatest stack depth: 13960 bytes left

Please press Enter to activate this console.
~ # ls
bin          proc          sbin
dev          r4l_e1000_demo.ko  sys
etc          root          usr
linuxrc      rust_helloworld.ko
[ 18.076815] ls (80) used greatest stack depth: 13920 bytes left
~ # insmod rust_helloworld.ko
[ 27.708899] rust_helloworld: Hello world from Rust module!
~ #

```

4. 添加Remove代码

a) 分析

要添加Remove逻辑，那就需要填充相关的方法。

首先尝试一下最初版本下卸载模块后再添加会发生什么，显示BAR 0错误，说明之前初始化pci设备的bar区域未能释放。

```
[ 251.018975] r4l_e1000_demo: pending_irqs: 3
[ 251.019437] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
~ # rmmod r4l_e1000_demo.ko
[ 255.902522] r4l_e1000_demo: Rust for linux e1000 driver demo (exit)
[ 255.903195] r4l_e1000_demo: Rust for linux e1000 driver demo (remove)
[ 255.903367] r4l_e1000_demo: Rust for linux e1000 driver demo (device_remove)
[ 255.904382] r4l_e1000_demo: Rust for linux e1000 driver demo (net device stop)
[ 255.904733] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 255.911859] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
~ # insmod r4l_e1000_demo.ko
[ 267.457486] r4l_e1000_demo: Rust for linux e1000 driver demo (init)
[ 267.457976] r4l_e1000_demo: Rust for linux e1000 driver demo (probe): None
[ 267.458221] r4l_e1000_demo 0000:00:03.0: BAR 0: can't reserve [mem 0xeb80000-0xeb9ffff]
[ 267.458483] r4l_e1000_demo: probe of 0000:00:03.0 failed with error -16
~ #
```

可以看到probe函数中有 `dev.request_selected_regions` 调用，需要找到对应的释放方法。该方法在 `pci::Device` 中定义，但却没有对应的释放方法。最后在bindings找到了pci的`release_region`方法，效仿 `request_selected_regions` 添加 `release_selected_regions` 方法。

遵循 应放尽放 的逻辑，在 `net::DeviceOperations` 中的 `stop` 函数中做了如下操作
(drop, enable, stop ...) :

```
fn stop(_dev: &net::Device, _data: &NetDevicePrvData) → Result {
    pr_info!("Rust for linux e1000 driver demo (net device stop)\n");
    _dev.netif_stop_queue();
    _dev.netif_carrier_off();
    _data.napi.disable();
    drop(_data);
    Ok(())
}
```

同样的在， `pci::Driver` 中的 `remove` 函数中做了如下操作：

```
fn remove(dev: &mut pci::Device, data: &Self::Data) {
    pr_info!("Rust for linux e1000 driver demo (remove)\n");
    dev.release_selected_regions(data._bars);
    dev.disable_device();
    drop(data);
}
```

然而，这样一通操作下来，虽然没有报BAR的问题，重新安装网卡反而Ping不通，而且napi poll也失效了

```
~ # ip link set eth0 up
[ 120.974814] r4l_e1000_demo: Rust for linux e1000 driver demo (net device open)
[ 120.976978] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 120.977335] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 120.977719] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
~ # [ 120.981098] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=7, rdh=0
[ 120.981761] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 120.982057] r4l_e1000_demo: pending_irqs: 3
[ 120.982237] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 120.982437] r4l_e1000_demo: pending_irqs: 0
```

经过问群里的同学，发现是irq未释放的问题，最后在 `stop` 函数加上了相关的逻辑

```
let irq_ptr = _data._irq_handler.load(core::sync::atomic::Ordering::Relaxed);
unsafe{ drop(Box::from_raw(irq_ptr));}
```

b) 实践

进入QEMU，安装模块并Ping通

```
~/Program/cctv-r4l-3-torey/src_e1000 git:(master):i1397
./build_image.sh

[ 14.652374] mount (72) used greatest stack depth: 14160 bytes left
[ 14.771429] mdev (74) used greatest stack depth: 13960 bytes left

Please press Enter to activate this console.
~ # ls
bin          proc      sbin
dev          r4l_e1000_demo.ko  sys
etc          root      usr
linuxrc      r4l_e1000.helloworld.ko

~ # insmod r4l_e1000_demo.ko
[ 52.376887] r4l_e1000_demo: loading out-of-tree module taints kernel.
[ 52.388687] r4l_e1000_demo: Rust for linux e1000 driver demo (init)
[ 52.389702] r4l_e1000_demo: Rust for linux e1000 driver demo (probe): None
[ 52.390308] r4l_e1000_demo: IRQ0000
[ 52.566683] APCI->LSB_LNKCEnabled at IRQ 11
[ 52.566683] APCI->LSB_LNKCEnabled at IRQ 11
[ 52.589385] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 52.591358] insmod (81) used greatest stack depth: 11176 bytes left
~ # ip link set eth0 up
78.678012] r4l_e1000_demo: Rust for linux e1000 driver demo (net device open)
78.673182] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
79.615053] r4l_e1000_demo: pending_irqs: 3 link becomes ready
~ # [ 78.679713] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 78.686276] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=7, rdh=0
78.687130] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 78.687480] r4l_e1000_demo: pending_irqs: 3
78.688338] r4l_e1000_demo: pending_irqs: 3 link becomes ready
79.417492] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
79.417380] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=7, rdh=0
79.417517] r4l_e1000_demo: pending_irqs: 3
[ 79.417556] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[ 79.418052] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=7, rdh=0
[ 79.418169] r4l_e1000_demo: pending_irqs: 3
79.419345] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[ 88.441378] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=3, tdh=3, rdt=7, rdh=0
88.441750] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 88.441904] r4l_e1000_demo: pending_irqs: 3
88.442065] r4l_e1000_demo: pending_irqs: 3 link becomes ready
88.442065] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[ 88.443373] r4l_e1000_demo: pending_irqs: 3
[ 88.443473] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=7, rdh=0
88.443561] r4l_e1000_demo: pending_irqs: 3
[ 88.443683] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
88.512554] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=7, rdh=0
[ 88.512937] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 88.513169] r4l_e1000_demo: pending_irqs: 3
[ 88.513357] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)

~ # [ 84.601649] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=7, rdh=0
~/Program/cctv-r4l-3-torey/src_e1000 git:(master):i1397
./build_image.sh

~ # [ 84.601649] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=7, rdh=0
[ 84.663269] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 84.663269] r4l_e1000_demo: pending_irqs: 3
[ 84.689344] r4l_e1000_demo: pending_irqs: 3 link becomes ready
ip addr add broadcast 10.0.2.255 dev eth0
[ 85.871438] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 85.873109] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
ip: RTNETLINK answers: Invalid argument
~ # ip link add dev 10.0.2.2 type veth peer dev eth0
[ 91.645183] r4l_e1000_demo: pending_irqs: 3 link becomes ready
[ 91.645691] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
~ # [ 93.304888] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=7, tdh=7, rdt=7, rdh=0
[ 93.306642] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 93.307965] r4l_e1000_demo: pending_irqs: 3
[ 93.308821] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
ip link set dev 10.0.2.1 down
~ # ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56 data bytes
[ 109.419359] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=7, rdh=0
[ 109.420133] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 109.420133] r4l_e1000_demo: pending_irqs: 131
[ 109.421918] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[ 109.424190] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=0, rdh=1
[ 109.424622] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[ 109.424911] r4l_e1000_demo: pending_irqs: 131
[ 109.426301] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2 seq=0 ttl=255 time=21.974 ms
[ 110.433664] r4l_e1000_demo: pending_irqs: 131
[ 110.434556] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=1, rdh=2
[ 110.435276] r4l_e1000_demo: pending_irqs: 131
[ 110.435998] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2 seq=1 ttl=255 time=4.356 ms
[ 110.436209] r4l_e1000_demo: pending_irqs: 131
[ 110.716453] r4l_e1000_demo: pending_irqs: 131 link becomes ready
[ 110.715419] r4l_e1000_demo: pending_irqs: 3
[ 110.716154] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[ 111.437893] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=2, rdh=3
[ 111.439554] r4l_e1000_demo: pending_irqs: 131
[ 111.440254] r4l_e1000_demo: pending_irqs: 131
[ 111.441565] r4l_e1000_demo: pending_irqs: 131 link becomes ready
[ 112.444507] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=3, rdh=4
[ 112.446343] r4l_e1000_demo: pending_irqs: 131
[ 112.447363] r4l_e1000_demo: pending_irqs: 131 link becomes ready
[ 112.448343] r4l_e1000_demo: pending_irqs: 131
64 bytes from 10.0.2.2 seq=2 ttl=255 time=5.799 ms
[ 113.450788] r4l_e1000_demo: pending_irqs: 131
[ 113.452589] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=4, rdh=5
[ 113.453524] r4l_e1000_demo: pending_irqs: 131
```

移除模块，并重新安装模块

执行配置并Ping通

```

~/Program/cicv-r4l-3-idrey/src_e1000 git:(master)±1397
./build_image.sh
[ 136.578809] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 136.571361] r4l_e1000 demo: pending_irqs: 3
[ 136.572475] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
[ 140.923496] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=7, rdh=0
[ 140.923571] r4l_e1000 demo: pending_irqs: 3
[ 140.924379] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
[ 141.001628] r4l_e1000 demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 141.001628] r4l_e1000 demo: pending_irqs: 3
ip: RTNETLINK answers: Invalid argument
# ip addr add broadcast 10.0.2.255 dev eth0
[ 145.390922] r4l_e1000 demo: Rust for linux e1000 driver demo (net device get_stats64)
[ 145.391393] r4l_e1000 demo: Rust for linux e1000 driver demo (net device get_stats64)
# ip route add default via 10.0.2.1 dev eth0
[ 149.625832] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 149.625856] r4l_e1000 demo: pending_irqs: 3
# ip route add default via 10.0.2.1
~ # ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56 data bytes
[ 156.408809] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=7, rdh=0
[ 156.409441] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 156.409661] r4l_e1000 demo: pending_irqs: 131
[ 156.409797] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
[ 156.410310] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=0, rdh=1
[ 156.410310] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 156.410921] r4l_e1000 demo: pending_irqs: 131
[ 156.411085] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=1 ttl=255 time=2.983 ms
[ 157.412979] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=1, rdh=2
[ 157.414181] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 157.415331] r4l_e1000 demo: pending_irqs: 131
[ 157.416622] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=1 ttl=255 time=5.094 ms
[ 158.410976] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=3, tdh=3, rdt=2, rdh=3
[ 158.420669] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 158.421586] r4l_e1000 demo: pending_irqs: 131
[ 158.421586] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=2 ttl=255 time=5.094 ms
[ 159.425868] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=3, rdh=4
[ 159.427667] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 159.428612] r4l_e1000 demo: pending_irqs: 131
[ 159.429612] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=3 ttl=255 time=5.111 ms
[ 160.431901] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=4, rdh=5
[ 160.433754] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 160.434755] r4l_e1000 demo: pending_irqs: 131
[ 160.435542] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)

~/Program/cicv-r4l-3-idrey/src_e1000 git:(master)±1397
./build_image.sh
[ 156.409797] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
[ 156.410310] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=0, rdh=1
[ 156.410719] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 156.410921] r4l_e1000 demo: pending_irqs: 131
[ 156.411086] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=1 ttl=255 time=5.094 ms
[ 157.411491] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=1, rdh=2
[ 157.414017] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 157.415231] r4l_e1000 demo: pending_irqs: 131
[ 157.416622] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=2 ttl=255 time=5.094 ms
[ 158.420669] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=3, tdh=3, rdt=2, rdh=3
[ 158.421589] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 158.422581] r4l_e1000 demo: pending_irqs: 131
[ 158.423581] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=3 ttl=255 time=5.094 ms
[ 159.425868] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=3, rdh=4
[ 159.427667] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 159.428612] r4l_e1000 demo: pending_irqs: 131
[ 159.429609] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=4 ttl=255 time=5.111 ms
[ 160.431901] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=4, rdh=5
[ 160.433754] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 160.434755] r4l_e1000 demo: pending_irqs: 131
[ 160.435542] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=5 ttl=255 time=5.111 ms
[ 161.437549] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=5, rdh=6
[ 161.437996] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 161.438703] r4l_e1000 demo: pending_irqs: 131
[ 161.438703] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=6 ttl=255 time=5.179 ms
[ 162.441104] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=7, tdh=7, rdt=6, rdh=7
[ 162.442926] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 162.443858] r4l_e1000 demo: pending_irqs: 131
[ 162.444541] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
64 bytes from 10.0.2.2: seq=7 ttl=255 time=4.764 ms
`C
--- 10.0.2.2 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 10.000/10.000/10.000/0.000 ms
# ip link set dev eth0 up
[ 166.010361] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=7, rdh=0
[ 166.011140] r4l_e1000 demo: pending_irqs: 3
[ 166.012361] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)
[ 201.848437] r4l_e1000 demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=7, rdh=0
[ 201.849595] r4l_e1000 demo: Rust for linux e1000 driver demo (handle_irq)
[ 201.849212] r4l_e1000 demo: pending_irqs: 3
[ 201.849452] r4l_e1000 demo: Rust for linux e1000 driver demo (napi_poll)

```

5. 注册字符设备

a) 分析

问题回答：

- Q：作业5中的字符设备/dev/cicv是怎么创建的？它的设备号是多少？它是如何与我们写的字符设备驱动关联上的？
- A：在 `build_image.sh` 中有一句 `mknod /dev/cicv c 248 0`，设备号248，驱动 init时会注册字符设备，这部分具体逻辑由Rust For Linux框架封装好。

首先需要弄懂字符设备驱动程序的逻辑，`write` 函数是把用户空间的数据读到内核空间，`read` 函数是把内核空间的数据写到用户空间。函数的返回值为操作数据的长度。实现如下：

```

fn write(_this: &Self, _file: &file::File, _reader: &mut impl
kernel::io_buffer::IoBufferReader, _offset: u64,) -> Result<usize> {
    pr_info!("Chrdev write, reader len is {}, offset is {}{}\n", _reader.len(),
    _offset);
    let mut buf = _this.inner.lock();
    if _reader.len() == 0 {
        return Ok(0);
    }
}

```

```

        }
        let len = core::cmp::min(_reader.len(), GLOBALMEM_SIZE as usize);
        _reader.read_slice(&mut buf[..len])?;
        Ok(len)
    }

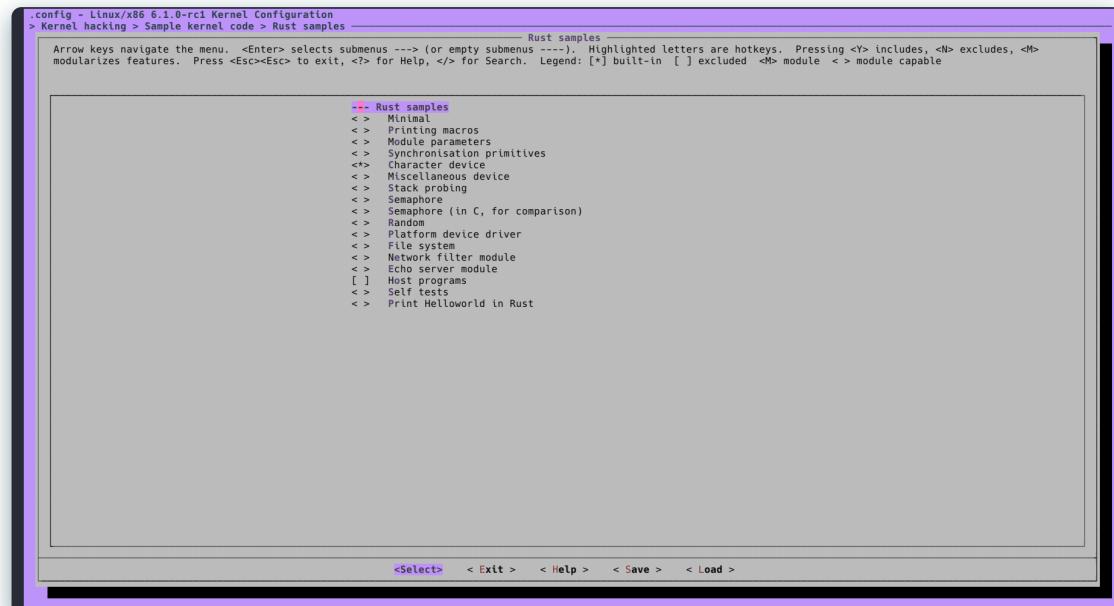
    fn read(_this: &Self, _file: &file::File, _writer: &mut impl
kernel::io_buffer::IoBufferWriter, _offset:u64,) → Result<usize> {
        pr_info!("Chrdev read, writer len is {}, offset is {}{}\n", _writer.len(),
_offset);
        let mut buf = _this.inner.lock();
        if _writer.len() == 0 || _offset ≠ 0 {
            return Ok(0);
        }
        let len = core::cmp::min(_writer.len(), GLOBALMEM_SIZE as usize);
        _writer.write_slice(&mut buf[..len])?;
        Ok(len)
    }
}

```

在 `read` 函数中，一开始并没有检查是否为0，导致一直循环打印。这一点需要注意。

b) 实践

配置内核



查看输入和输出

```

[ 14.588189] Run sbn/init as init process
[ 14.623846] mount (72) used greatest stack depth: 13920 bytes left

Please press Enter to activate this console.
~ #
~ # echo "Hello" > /dev/civc
[ 25.664591] rust_chrdev: Chrdev write, reader len is 6, offset is 0
~ # cat /dev/civ
cat: can't open '/dev/civ': No such file or directory
~ # cat /dev/civc
[ 35.179389] rust_chrdev: Chrdev read, writer len is 4096, offset is 0
Hello
[ 35.188963] rust_chrdev: Chrdev read, writer len is 4096, offset is 4096
~ # [ 332.880238] kworker/dying (28) used greatest stack depth: 13728 bytes left
~ #

```

6. 实现Completion驱动

a) 分析

阅读C代码，可知驱动使用和completion相关的函数来实现类似于同步的功能。使用Rust实现的思路为：

1. 同为字符设备，可以复用实验5的框架
2. 将C语言中的相关调用利用FFI机制提供Rust接口

通过阅读C语言代码可以发现，驱动程序中有一个 `completion_dev` 结构体，该结构体成员为 `completion` 和字符设备结构体 `cdev`。在 `read` 和 `write` 函数中有 `wait_for_completion` 和 `complete` 两个调用，且其作用对象均为 `dev->completion`，可见 `completion` 类似于一个条件变量。而且，`completion_dev` 结构体是从 `inode` 中取出，由于同一类型的设备只对应一个 `inode`，因此这个 `completion` 对于同一字符设备应该是全局的。当打开多个设备时，其操作的 `completion` 为同一个。因此在Rust驱动程序中，应该将其设为全局变量，也即类似于作业5，使用static声明。

首先得加入 `completion.h` 的bindings。然后声明一个static的 `bindings :: completion`。一开始直接使用 `bindings :: completion`，但因为线程安全性报错，然后实现了相应的Trait后，又提示不在当前的crate，不能在本文件实现。因此声明了一个结构体来hold住Completion，然后在初始化时使用 `bindings :: completion :: default()`，又提示不行。无奈只能再包一层 `Option`，然后在static时初始化为None。接着在 `init` 的时候调用 `bindings :: completion :: default()` 和 `init_completion` 进行初始化。对于 `wait_for_completion` 和 `complete` 两个调用使用相对应的bindings即可。

在实现中还需要得到当前task的pid，查阅代码后发现调用 `task :: Task :: current().pid()` 即可。

对于获取当前函数名称，直接hard-code写入open和write。

需要注意的是，在 `module!` 宏中需要将name设为带有 `completion` 的名称，才能被加载模块的脚本自动装载。

b) 实践

终端—

```
~/Program/clicv-r4l-3-ldney/r4l_experiment git:(master)±1405
./boot.sh
[ 1.919646] registered taskstats version 1
[ 1.925377] Loading compiled-in X.509 certificates
[ 1.925377] cryptomgr.test (44) used greatest stack depth: 15584 bytes left
[ 1.925377] PM: Modifying memory: 16384 bytes
[ 1.932463] printing console [4con] enabled
[ 1.932611] netconsole: network logging started
[ 1.935091] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[ 1.988739] modprobe (66) used greatest stack depth: 14272 bytes left
[ 2.012508] cfg80211: Loaded X.509 cert 'forshee: 00b28ddf47ae9ceaf'
[ 2.012508] cfg80211: failed to load regulatory.db
[ 2.014565] cfg80211: failed to load regulatory.db
[ 2.015802] ALSA device list:
[ 2.016128]   No soundcards found.
[ 2.057772] Freeing unused kernel image (initmem) memory: 1324K
[ 2.059181] Write protecting kernel image (initmem) memory: 1454K
[ 2.059181] Freeing unused kernel image (text/rodata/gap) memory: 2832K
[ 2.062811] Freeing unused kernel image (rodata/data/gap) memory: 824K
[ 2.214893] x86/mm: Checked W-X mappings: passed, no W-X pages found.
[ 2.215895] tsc: Refined TSC clocksource calibration: 2999.976 MHz
[ 2.216314] clocksource: tsc: mask: 0xffffffffffff max_cycles: 440795281911 ns
[ 2.216314] clocksource: tsc: period: 13241111111111111111111111111111 ns
[ 2.217252] init: init as init process
[ 2.239604] mount (70) used greatest stack depth: 14144 bytes left
[ 2.283420] ip (73) used greatest stack depth: 13832 bytes left
[ 2.631874] input: IMExPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input3
[ 4.301512] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 4.301512] e1000: (PCI:0000:00:00:02) HW(IF:NETDEV_CHANGE): eth0: [link becomes ready]
[ 5.346792] mount (75) used greatest stack depth: 12096 bytes left
~ # cd /mnt/002_completion/
/mnt/002_completion # ls
Makefile      completion.ko    completion.mod.o  load_module.sh
Module.symvers completion.mod completion.o    modules.order
README        completion.mod.c completion.rs
/mnt/002_completion # ./load_module.sh
[ 19.446839] completion: loading out-of-tree module taints kernel.
[ 19.454845] completion: Rust character device sample (init)
248
/mnt/002_completion # ls /dev
completion null      pts      tty50
console  ptmx      tty
/mnt/002_completion # cat /dev/completion
[ 29.173696] completion: open is invoked
[ 29.174501] completion: read is invoked
[ 29.174501] completion: process is going to sleep
[ 53.567126] completion: open is invoked
[ 53.567433] completion: write is invoked
[ 53.567647] completion: process 98 awakening the readers...
[ 53.568030] completion: awoken 89
/mnt/002_completion #
```

终端二

```
~/Program/clicv-r4l-3-ldney/r4l_experiment git:(master)±1405
telnet localhost 7023
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

~ # ls /dev
completion console null      ptmx      pts      tty      tty50
~ # echo "Something" > /dev/completion
~ #
```