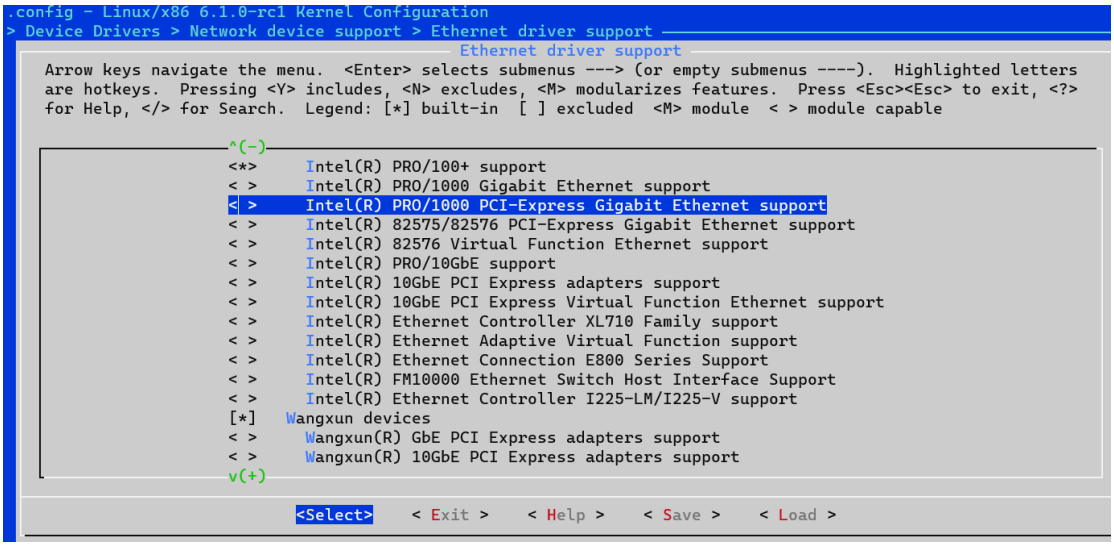
1、编译成内核模块，是在哪个文件中以哪条语句定义的?

答：通常在源代码的目录下，每个文件夹都有一个对应的 Kconfig 和 Makerfile 文件，在 Kbuild 文件中会有一句 obj-y = my_module.o 定义。如果 Kbuild 文件和 Makefile 文件同时存在，那么 Kbuild 文件优先。

2、该模块位于独立的文件夹内，却能编译成 Linux 内核模块，这叫做 out-of-tree module，请分析它是如何与内核代码产生联系的?

答：out-of-tree module 通过独立文件夹内编写的内核模块代码、配置内核编译选项、编写 Makefile 文件以及集成到内核源树等方式与内核产生联系。实际应用中可以用于扩展内核功能、实现个性化定制以及修复内核问题等。

禁用默认网卡驱动



启动编译好的内核

```
[    1.797385] clocksource: Switched to clocksource tsc
[    1.990353] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input3
[    2.000756] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[    2.001602] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[    2.014121] IP-Config: Complete:
[    2.014335]      device=eth0, hwaddr=52:54:00:12:34:56, ipaddr=10.0.2.15, mask=255.255.255.0, gw=10.0.2.1
[    2.014515]      host=10.0.2.15, domain=, nis-domain=(none)
[    2.014596]      bootserver=255.255.255.255, rootserver=255.255.255.255, rootpath=
[    2.017366] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[    2.066322] modprobe (66) used greatest stack depth: 14280 bytes left
[    2.069299] modprobe (67) used greatest stack depth: 14272 bytes left
[    2.073985] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[    2.074962] platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
[    2.075237] cfg80211: failed to load regulatory.db
[    2.076039] ALSA device list:
[    2.076273]   No soundcards found.
[    2.113277] Freeing unused kernel image (initmem) memory: 1328K
[    2.115661] Write protecting the kernel read-only data: 24576k
[    2.117636] Freeing unused kernel image (text/rodata gap) memory: 2032K
[    2.118101] Freeing unused kernel image (rodata/data gap) memory: 824K
[    2.180331] x86/mm: Checked W+X mappings: passed, no W+X pages found.
[    2.180690] Run sbin/init as init process
[    2.207909] mount (71) used greatest stack depth: 13920 bytes left

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

查看网络设备

```
~ # ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
[   18.677423] ip (80) used greatest stack depth: 13032 bytes left
```

挂在自定义驱动模块

```
~ # insmod r4l_e1000_demo.ko
[   22.780120] r4l_e1000_demo: loading out-of-tree module taints kernel.
[   22.783768] r4l_e1000_demo: Rust for linux e1000 driver demo (init)
[   22.784130] r4l_e1000_demo: Rust for linux e1000 driver demo (probe): None
[   22.880434] ACPI: \_SB_.LNKC: Enabled at IRQ 11
[   22.901064] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[   22.902237] insmod (81) used greatest stack depth: 11144 bytes left
```

唤醒 eth0 网卡

```
~ # ip link set eth0 up
[   27.577621] r4l_e1000_demo: Rust for linux e1000 driver demo (net device open)
[   27.579664] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[   27.580459] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[   27.582534] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
~ # [   27.587945] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=0, tdh=0, rdt=7, rdh=0
[   27.588412] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[   27.588530] r4l_e1000_demo: pending_irqs: 3
[   27.588963] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[   27.841674] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=1, tdh=1, rdt=7, rdh=0
[   27.841852] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[   27.841904] r4l_e1000_demo: pending_irqs: 3
[   27.842042] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[   28.441500] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=2, tdh=2, rdt=7, rdh=0
[   28.441883] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[   28.442033] r4l_e1000_demo: pending_irqs: 3
[   28.442214] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[   28.889733] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=3, tdh=3, rdt=7, rdh=0
[   28.889930] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[   28.890023] r4l_e1000_demo: pending_irqs: 3
[   28.890081] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[   28.890439] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=7, rdh=0
[   28.890585] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
```

再次查看网络设备

```
~ # ip addr add 10.0.2.15/255.255.255.0 dev eth0
[  317.613960] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[  317.614159] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
~ # ip a
[  319.951627] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
[  319.951819] r4l_e1000_demo: Rust for linux e1000 driver demo (net device get_stats64)
1: lo: <LOOPBACK> mtu 65536 qdisc noop qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 52:54:00:12:34:56 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe12:3456/64 scope link
       valid_lft forever preferred_lft forever
~ #
```

添加路由，ping 测试

```
~ # ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
[  356.929146] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=4, tdh=4, rdt=7, rdh=0
[  356.929621] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[  356.929756] r4l_e1000_demo: pending_irqs: 131
[  356.929968] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
[  356.931034] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=5, tdh=5, rdt=0, rdh=1
[  356.931175] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[  356.931273] r4l_e1000_demo: pending_irqs: 131
[  356.931763] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=0 ttl=255 time=8.190 ms
[  357.934806] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=6, tdh=6, rdt=1, rdh=2
[  357.934988] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[  357.935085] r4l_e1000_demo: pending_irqs: 131
[  357.935151] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=1 ttl=255 time=0.782 ms
[  358.936467] r4l_e1000_demo: Rust for linux e1000 driver demo (net device start_xmit) tdt=7, tdh=7, rdt=2, rdh=3
[  358.936752] r4l_e1000_demo: Rust for linux e1000 driver demo (handle_irq)
[  358.936820] r4l_e1000_demo: pending_irqs: 131
[  358.936895] r4l_e1000_demo: Rust for linux e1000 driver demo (napi poll)
64 bytes from 10.0.2.2: seq=2 ttl=255 time=0.816 ms
```