

新建 rust_helloworld.rs

```
yega@debian:~/cicv/cicv-r4l-yegawong/linux$ cd samples/
yega@debian:~/cicv/cicv-r4l-yegawong/linux/samples$ ls
acrn      configfs  fprobe    Kconfig   kobject   Makefile   pktgen    seccomp    uhid       vfs
auxdisplay connector ftrace    kdb        kprobes   mei        qmi       timers     user_events watchdog
binderfs  coresight hidraw     kfifo      landlock  nitro_enclaves rpmsg     trace_events v4l       watch_queue
e         bpff      fanotify  hw_breakpoint kmemleak  livepatch  pidfd     rust       trace_printk vfio-mdev
yega@debian:~/cicv/cicv-r4l-yegawong/linux/samples$ cd rust/
yega@debian:~/cicv/cicv-r4l-yegawong/linux/samples/rust$ ls
hostprogs rust_chrdev.rs  rust_minimal.rs  rust_netfilter.rs  rust_random.rs  rust_semaphore.rs
Kconfig    rust_echo_server.rs rust_miscdev.rs    rust_platform.rs   rust_selftests.rs rust_stack_probing.rs
Makefile   rust_fs.rs      rust_module_parameters.rs rust_print.rs      rust_semaphore_c.c rust_sync.rs
yega@debian:~/cicv/cicv-r4l-yegawong/linux/samples/rust$ touch rust_helloworld.rs
yega@debian:~/cicv/cicv-r4l-yegawong/linux/samples/rust$
```

添加代码

```
// SPDX-License-Identifier: GPL-2.0
//! Rust minimal sample.

use kernel::prelude::*;

module! {
    type: RustHelloWorld,
    name: "rust_helloworld",
    author: "whocare",
    description: "hello world module in rust",
    license: "GPL",
}

struct RustHelloWorld {}

impl kernel::Module for RustHelloWorld {
    fn init(_name: &'static CStr, _module: &'static ThisModule) -> Result<Self> {
        pr_info!("Hello World from Rust module\h");
        Ok(RustHelloWorld {})
    }
}
```

修改 Kconfig

config SAMPLE_RUST_HELLOWORLD

tristate "Self helloworld"

help

This option builds the self helloworld cases for Rust.

If unsure, say N.

```
config SAMPLE_RUST_SELFTESTS
    tristate "Self tests"
    help
        This option builds the self test cases for Rust.

        If unsure, say N.

config SAMPLE_RUST_HELLOWORLD
    tristate "Self helloworld"
    help
        This option builds the self helloworld cases for Rust.

        If unsure, say N.
```

修改 Makefile

obj-\$(CONFIG_SAMPLE_RUST_HELLOWORLD)

+= rust_helloworld.o

```
# SPDX-License-Identifier: GPL-2.0

obj-$(CONFIG_SAMPLE_RUST_MINIMAL) += rust_minimal.o
obj-$(CONFIG_SAMPLE_RUST_PRINT) += rust_print.o
obj-$(CONFIG_SAMPLE_RUST_MODULE_PARAMETERS) += rust_module_parameters.o
obj-$(CONFIG_SAMPLE_RUST_SYNC) += rust_sync.o
obj-$(CONFIG_SAMPLE_RUST_CHRDEV) += rust_chrdev.o
obj-$(CONFIG_SAMPLE_RUST_MISCDEV) += rust_miscdev.o
obj-$(CONFIG_SAMPLE_RUST_STACK_PROBING) += rust_stack_probing.o
obj-$(CONFIG_SAMPLE_RUST_SEMAPHORE) += rust_semaphore.o
obj-$(CONFIG_SAMPLE_RUST_SEMAPHORE_C) += rust_semaphore_c.o
obj-$(CONFIG_SAMPLE_RUST_RANDOM) += rust_random.o
obj-$(CONFIG_SAMPLE_RUST_PLATFORM) += rust_platform.o
obj-$(CONFIG_SAMPLE_RUST_NETFILTER) += rust_netfilter.o
obj-$(CONFIG_SAMPLE_RUST_ECHO_SERVER) += rust_echo_server.o
obj-$(CONFIG_SAMPLE_RUST_FS) += rust_fs.o
obj-$(CONFIG_SAMPLE_RUST_SELFTESTS) += rust_selftests.o
obj-$(CONFIG_SAMPLE_RUST_HELLOWORLD) += rust_helloworld.o

subdir-$(CONFIG_SAMPLE_RUST_HOSTPROGS) += hostprogs
~
```

修改配置

```
rust samples
Arrow keys navigate the menu. <Enter> selects submenus --- (or empty submenus ---). Highlighted letters
are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?>
for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

^(-)
< > Printing macros (NEW)
< > Module parameters (NEW)
< > Synchronisation primitives (NEW)
< > Character device (NEW)
< > Miscellaneous device (NEW)
< > Stack probing (NEW)
< > Semaphore (NEW)
< > Semaphore (in C, for comparison) (NEW)
< > Random (NEW)
< > Platform device driver (NEW)
< > File system (NEW)
< > Network filter module (NEW)
< > Echo server module (NEW)
[ ] Host programs (NEW)
< > Self tests (NEW)
<M> Self helloworld

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

编译完成, 运行 src_e1000/build_image.sh 产生 rust_helloworld.ko 文件

```
yega@debian:~/cicv/cicv-r4l-yegawong/linux/samples/rust$ ls
built-in.a      rust_echo_server.rs  rust_helloworld.o    rust_platform.rs    rust_stack_probing.rs
hostprogs      rust_fs.rs          rust_helloworld.rs   rust_print.rs       rust_sync.rs
Kconfig        rust_helloworld.ko  rust_minimal.rs     rust_random.rs
Makefile       rust_helloworld.mod rust_miscdev.rs      rust_selftests.rs
modules.order  rust_helloworld.mod.c rust_module_parameters.rs rust_semaphore_c.c
rust_chrdev.rs rust_helloworld.mod.o rust_netfilter.rs    rust_semaphore.rs
yega@debian:~/cicv/cicv-r4l-yegawong/linux/samples/rust$ |
```

进入 shell,发现 rust_hellowrold.ko 文件

```
[ 13.837000] platform regulatory.0: Direct firmware load for regulat
[ 13.837235] cfg80211: failed to load regulatory.db
[ 13.838034] ALSA device list:
[ 13.838175]   No soundcards found.
[ 13.873850] Freeing unused kernel image (initmem) memory: 1324K
[ 13.875662] Write protecting the kernel read-only data: 24576k
[ 13.877199] Freeing unused kernel image (text/rodata gap) memory: 2
[ 13.877727] Freeing unused kernel image (rodata/data gap) memory: 8
[ 13.939659] x86/mm: Checked W+X mappings: passed, no W+X pages foun
[ 13.939895] Run sbin/init as init process
[ 13.961640] mount (72) used greatest stack depth: 13920 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 | |

~ # |

Insmod rust_hellowrold.ko 文件打印输出

~ # ls

| | | |
|---------|--------------------|------|
| bin | proc | sbin |
| dev | r4l_e1000_demo.ko | sys |
| etc | root | usr |
| linuxrc | rust_helloworld.ko | |

~ # insmod rust_helloworld.ko

[99.715496] rust_helloworld: Hello World from Rust module

~ # |