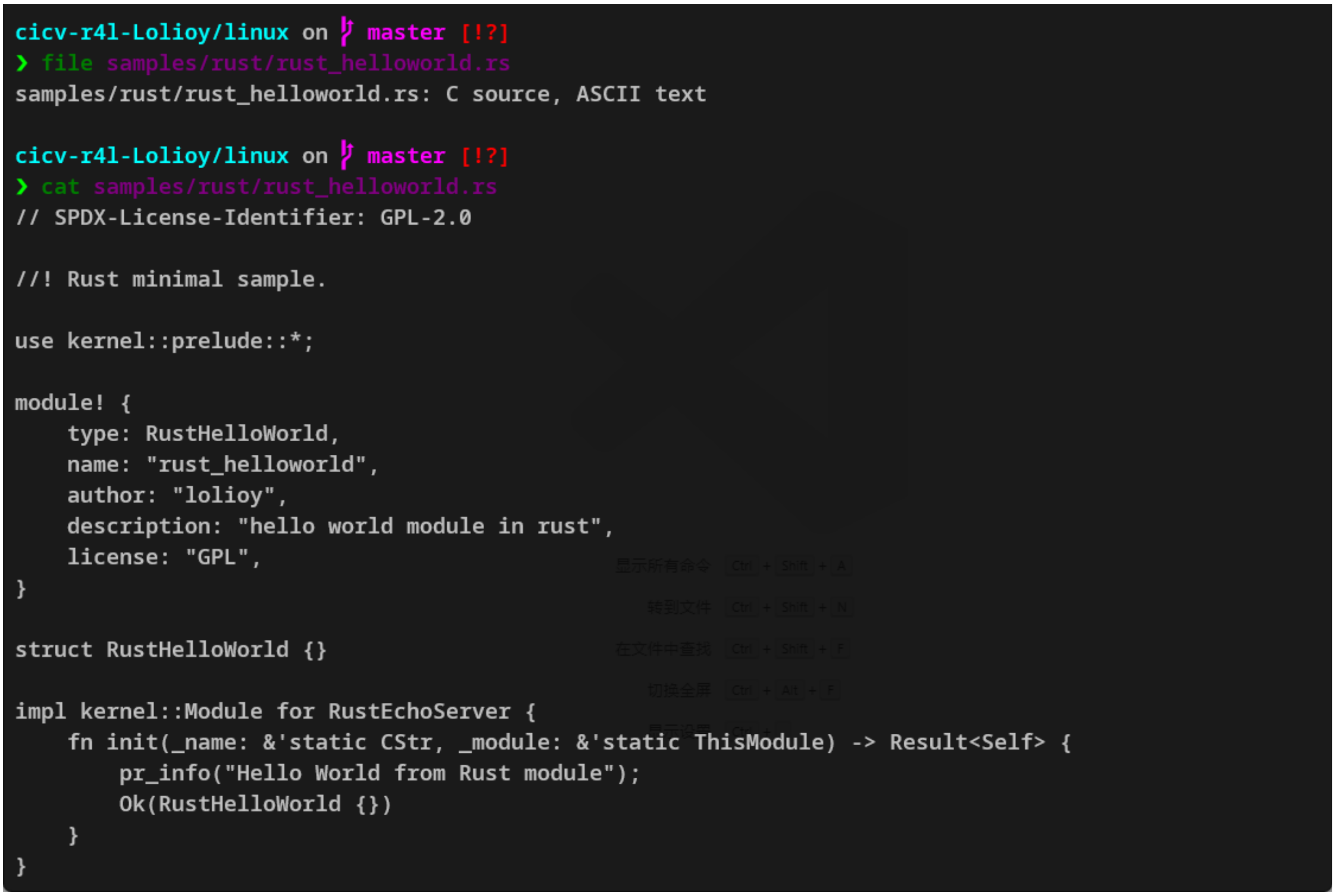
Task3

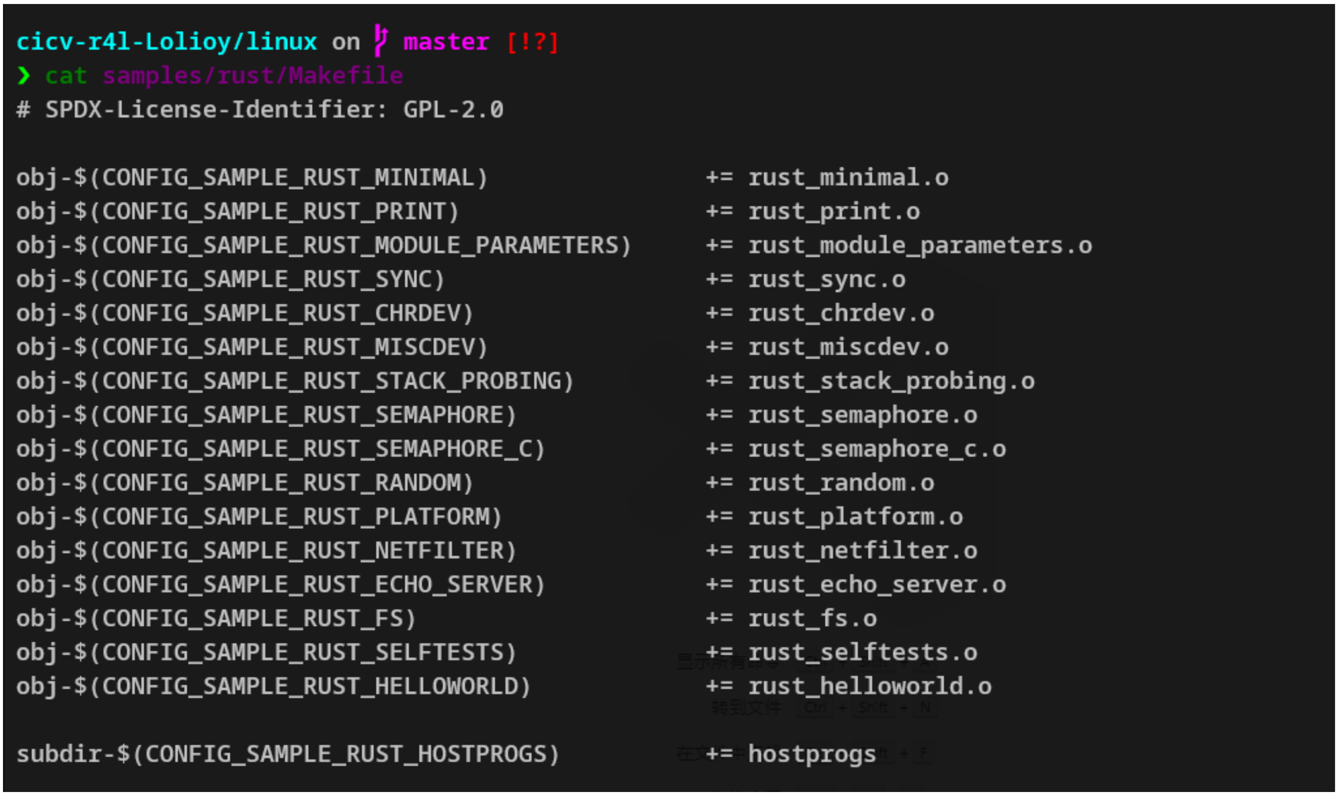
Step1 - 编写 Rust 模块代码



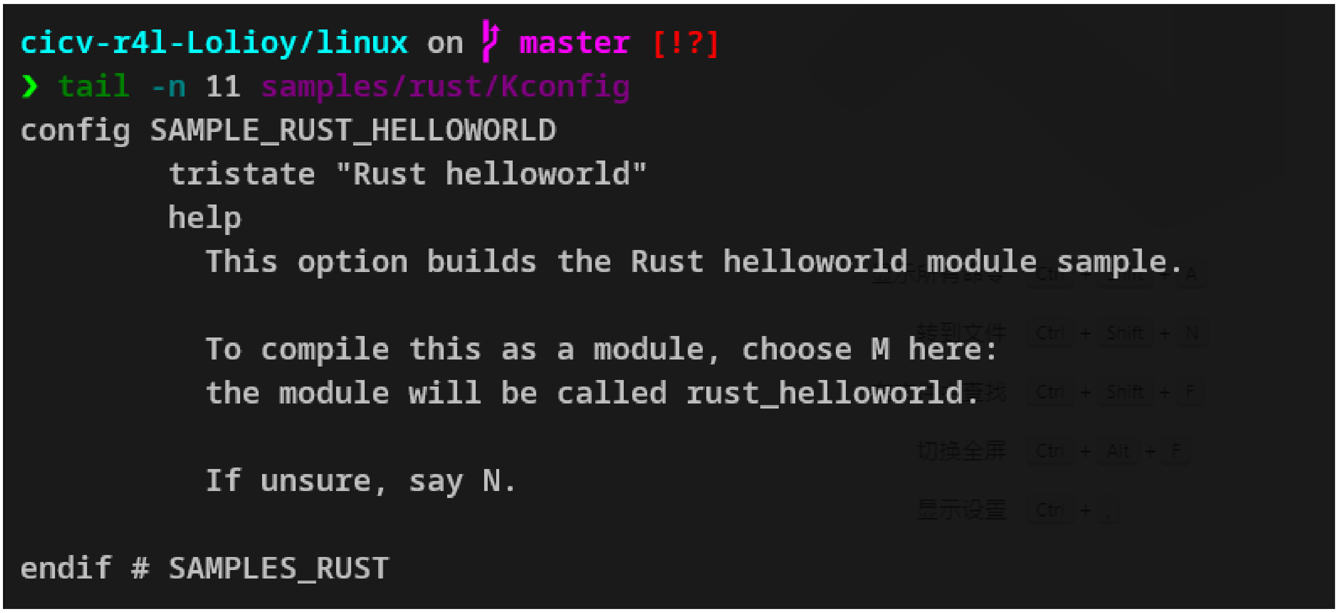
```
cicv-r4l-Lolioy/linux on  master [!?]
❯ file samples/rust/rust_helloworld.rs
samples/rust/rust_helloworld.rs: C source, ASCII text

cicv-r4l-Lolioy/linux on  master [!?]
❯ cat samples/rust/rust_helloworld.rs
// SPDX-License-Identifier: GPL-2.0

//! Rust minimal sample.

use kernel::prelude::*;

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

struct RustHelloWorld {}

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

显示所有命令 Ctrl + Shift + A
转到文件 Ctrl + Shift + N
在文件中查找 Ctrl + Shift + F
切换全屏 Ctrl + Alt + F

Step2 - 修改 Makefile 和 Kconfig 文件

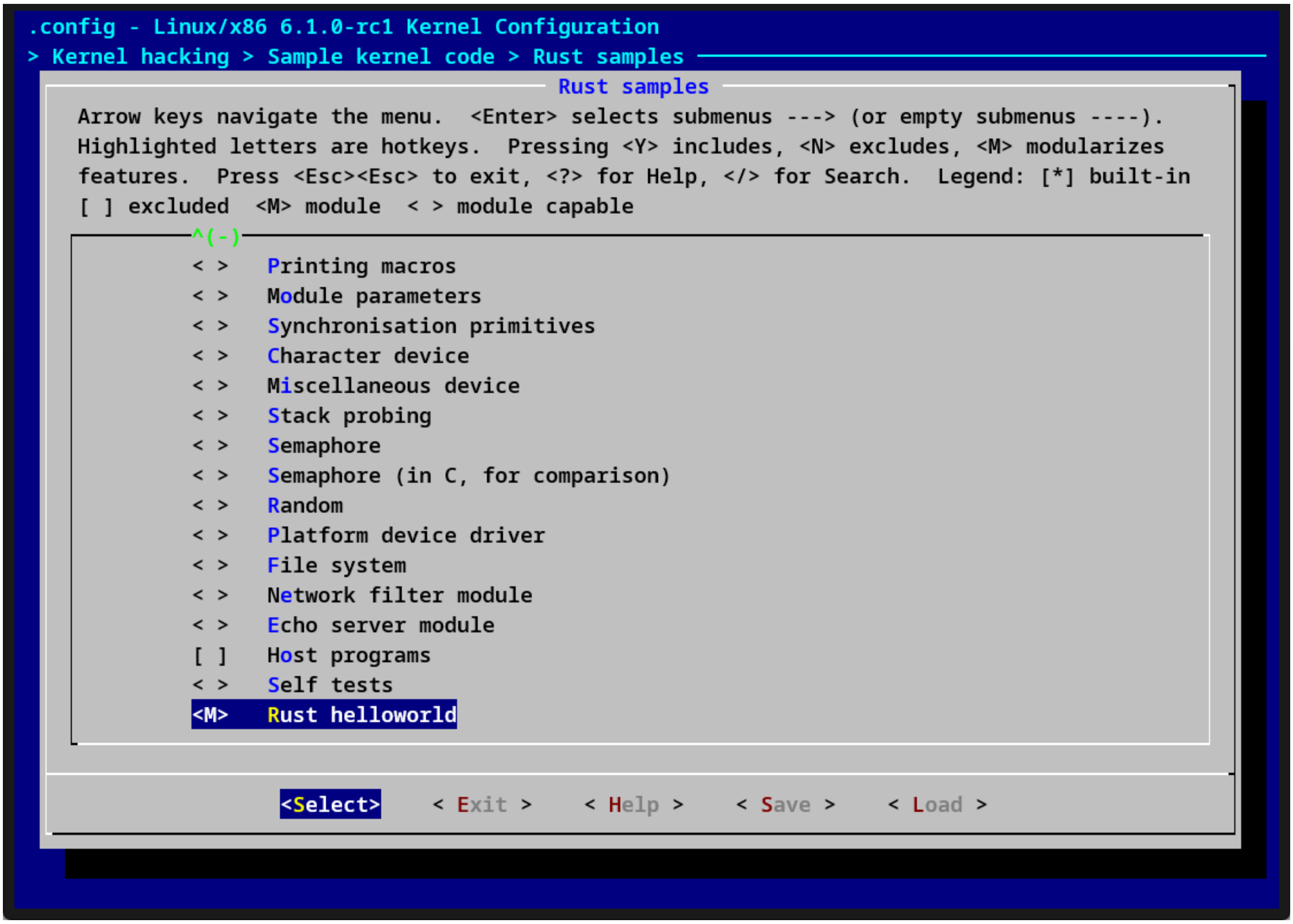


```
cicv-r4l-Lolioy/linux on  master [!?]
❯ cat samples/rust/Makefile
# SPDX-License-Identifier: GPL-2.0

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
```



```
cicv-r4l-Lolioy/linux on  master [!?]
❯ tail -n 11 samples/rust/Kconfig
config SAMPLE_RUST_HELLOWORLD
        tristate "Rust helloworld"
        help
          This option builds the Rust helloworld module sample.

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

          If unsure, say N.

endif # SAMPLES_RUST
```

Step3 - menuconfig 配置

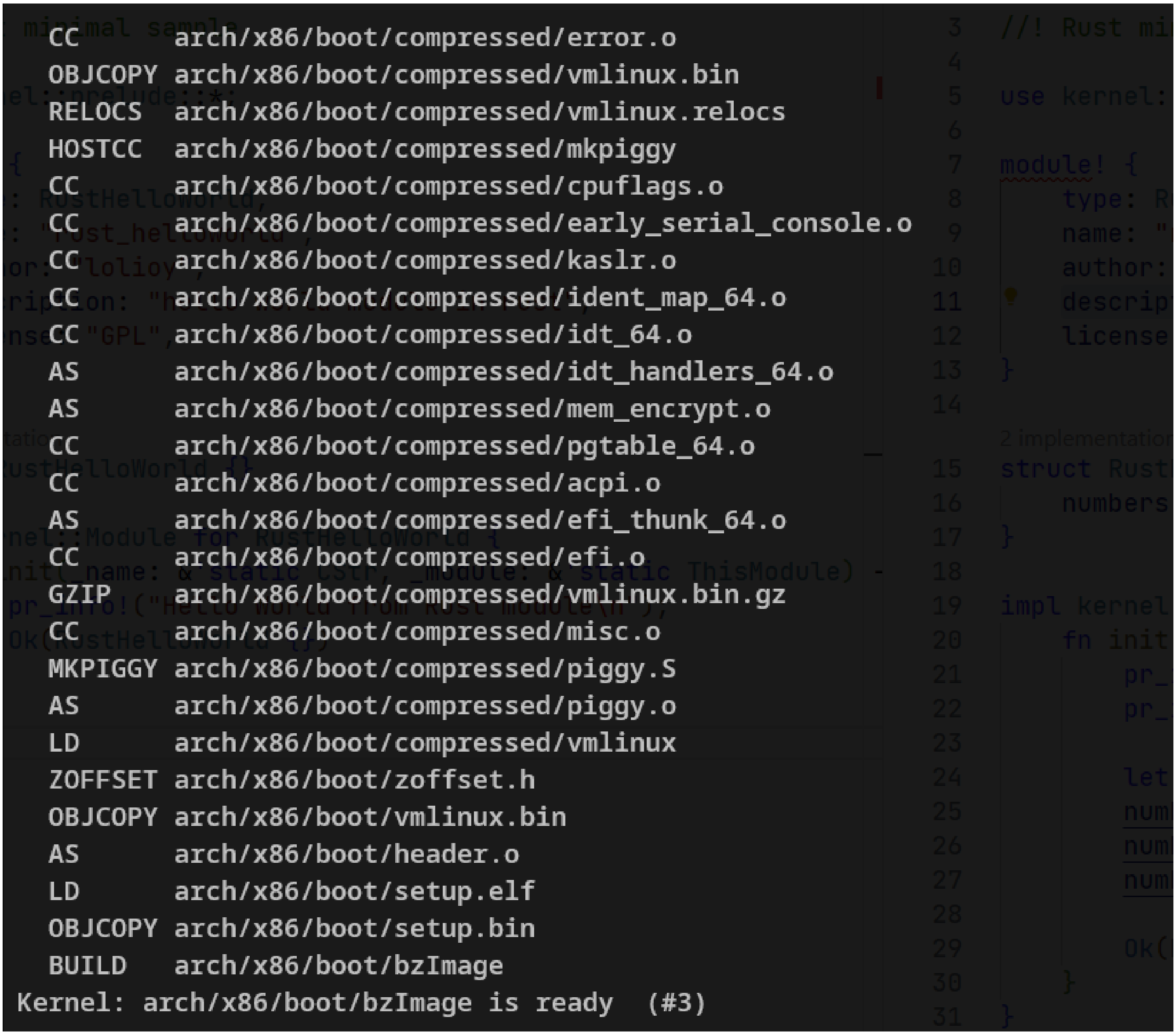


```
.config - Linux/x86 6.1.0-rc1 Kernel Configuration
> Kernel hacking > Sample kernel code > Rust samples ─────────────────────────
                              Rust samples
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  Highlighted letters are hotkeys.  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
          < >   Module parameters
          < >   Synchronisation primitives
          < >   Character device
          < >   Miscellaneous device
          < >   Stack probing
          < >   Semaphore
          < >   Semaphore (in C, for comparison)
          < >   Random
          < >   Platform device driver
          < >   File system
          < >   Network filter module
          < >   Echo server module
          [ ]   Host programs
          < >   Self tests
          <M>   Rust helloworld

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

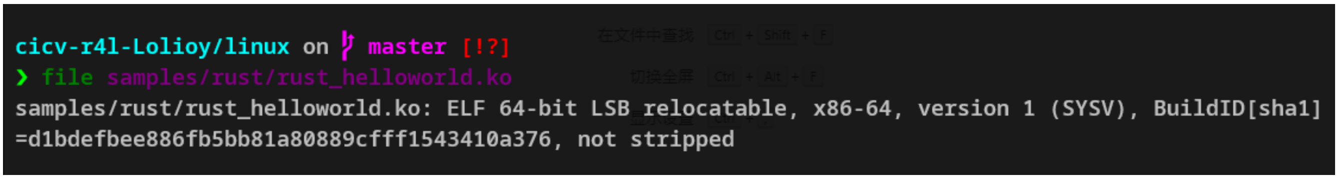
Step4 - 编译 Linux 内核

Bash

```bash
make LLVM=1 -j$(nproc)
```

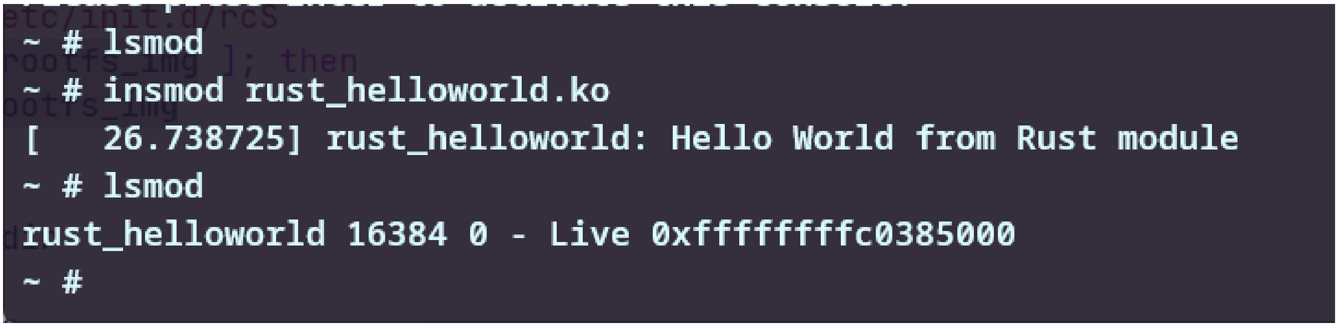


```
  CC      arch/x86/boot/compressed/error.o
  OBJCOPY arch/x86/boot/compressed/vmlinux.bin
  RELOCS  arch/x86/boot/compressed/vmlinux.relocs
  HOSTCC  arch/x86/boot/compressed/mkpiggy
  CC      arch/x86/boot/compressed/cpuflags.o
  CC      arch/x86/boot/compressed/early_serial_console.o
  CC      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/efi.o
  GZIP    arch/x86/boot/compressed/vmlinux.bin.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  (#3)
```



```
cicv-r4l-Lolioy/linux on  master [!?]
❯ file samples/rust/rust_helloworld.ko
samples/rust/rust_helloworld.ko: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), BuildID[sha1]
=d1bdefbee886fb5bb81a80889cfff1543410a376, not stripped
```

Step5 - 安装 rust_helloworld 模块



```
~ # lsmod
~ # insmod rust_helloworld.ko
[   26.738725] rust_helloworld: Hello World from Rust module
~ # lsmod
rust_helloworld 16384 0 - Live 0xffffffffc0385000
~ #
```