

Writing a Kernel Module in C for Linux

Writing a Linux kernel module allows you to extend or modify the behavior of the Linux kernel without needing to reboot or recompile the kernel itself. Kernel modules can be drivers, filesystems, or other system utilities.

2. Development Environment Setup

Install the required tools and headers:

```
sudo apt update
sudo apt install build-essential linux-headers-$(uname -r)
```

4. Writing the Makefile

Filename: Makefile

```
obj-m += hello_module.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

6. Debugging and Logging

Use `dmesg` to view kernel logs. Use `printk()` for logging—use log levels like `KERN_INFO`, `KERN_WARNING`, `KERN_ERR`.

8. Advanced Topics

- Working with device drivers
- Using `procfs` or `sysfs` for user-space communication

- Handling interrupts
- Writing character device drivers

10. Conclusion

Kernel modules are powerful tools for extending kernel capabilities on demand. Learning to write them improves your understanding of how the Linux kernel operates under the hood. Always follow best practices and test thoroughly.