

Linux Kernel Programming

General

- [The Linux Kernel documentation \(raw\)](#)
- [The DeviceTree Specification](#)

Style & Documentation

- [How to write kernel documentation](#)
- [Linux kernel coding style](#)
- [Writing kernel-doc comments](#)
- [kernel-doc-nano-HOWTO.txt](#)
- [How to get printk format specifiers right](#)

RT Linux

- [Real-Time Linux Wiki](#)
- Linux Foundation Wiki: [BROKEN-LINK:Real-Time LINUXLINK-BROKEN](#)
- Linux Foundation Wiki: [BROKEN-LINK:HOWTO setup Linux with PREEMPT_RT properlyLINK-BROKEN](#)
- Linux Foundation Wiki: [BROKEN-LINK:HOWTO build a simple RT applicationLINK-BROKEN](#)

Drivers

- J. Corbet, A. Rubini, G. Kroah-Hartman: [Linux Device Drivers \(LDD3\)](#)
- P. J. Salzman, M. Burian, O. Pomerantz: [The Linux Kernel Module Programming Guide](#)
- Tariq Shureih (Intel): [HOWTO: Linux Device Driver Dos and Don'ts](#)
- Linux Kernel Documentation: [Driver Model](#)
- Linux Kernel Documentation: [Building External Modules](#)

Network

- [PHY Abstraction Layer](#)
- [phylink \(SFP\)](#)

Code/Examples

- [ioctl called from within kernel code](#)

Debug

- Embedded Linux Wiki: [Debugging by printing](#)
- Stack Overflow: [Linux-kernel debug printouts?](#)
- Linux Kernel Documentation: [pr_debug\(\)](#)

Hints

- Use `WARN_ON()` and `BUG_ON()` macros for run-time assertions.
- Force a backtrace (programmatically) using `dump_stack()`.
- Use this code in a `Makefile` to enable `pr_debug()`, `dev_dbg()` and friends based output in kernel ring buffer:
 - for a particular source file

```
CFLAGS_<basename>.o := -DDEBUG
```

- or; in whole kernel module

```
ccflags-y += -DDEBUG
```

See [this](#) Wiki page how to show such messages.

Locking

- Linux Kernel Documentation: [Proper Locking Under a Preemptible Kernel](#)
- Linux Kernel Documentation: [Spin locks](#)
- Linux Kernel Documentation: [Generic Mutex Subsystem](#)
- Linux Kernel Documentation: [RT-mutex implementation design](#)

Hints

- Suppress scheduling by use of function `preempt_disable()`.
- Functions `irqs_disabled()` and `in_interrupt()` are very helpful regarding protection of non-preemptive code (see also the comments at [Stack Overflow](#)).
- For use of `[num]delay()`, `usleep()` and friends see [timers-howto.txt](#).

Memory

- Mel Gorman: [Understanding the Linux Virtual Memory Manager](#)
- Ulrich Drepper: [What every programmer should know about memory](#)
- Server Fault: [How is kernel oom score calculated?](#)

Random

- Jake Edge: [A system call for random numbers: getrandom\(\)](#)
- ArchWiki: [Rng-tools](#)

From:

<https://wiki.rho62.de/> - **rho62 Wiki**

Permanent link:

<https://wiki.rho62.de/doku.php?id=programming:linux:start>

Last update: **2023/05/22 08:54**

