```c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

static uint test_1 = 0;
module_param(test_1, uint, 0644);

static int __init test_module_init(void)
{
    int     d1, d2;

    d1 = 1234;
    d2 = d1 / test_1;

    return d2;
}

static void __exit test_module_exit(void)
{
}

module_init(test_module_init);
module_exit(test_module_exit);
MODULE_LICENSE("GPL");
```

```
obj-m := test.o

ccflags-y += -O0 -march=armv7-a

SRC := $(shell pwd)

#ccflags-y := $(shell echo $(ccflags-y) | sed -e 's/-O[0123s]//g')


all:
    echo "test build"
    echo $(KERNEL_SRC)
    echo $(ccflags-y)
    $(MAKE) -C $(KERNEL_SRC) M=$(SRC)
modules_install:
    echo "install test"
    echo $(KERNEL_SRC)
    $(MAKE) -C $(KERNEL_SRC) M=$(SRC) modules_install

clean:
    rm -f *.o *~ core .depend .*.cmd *.ko *.mod.c
    rm -f Module.markers Module.symvers modules.order
    rm -rf .tmp_versions Modules.symvers
```

```
00000000 <init_module>:

static uint test_1 = 0;
module_param(test_1, uint, 0644);

static int __init test_module_init(void)
{
   0:   e52de004    push    {lr}        ; (str lr, [sp, #-4]!)
   4:   e24dd00c    sub sp, sp, #12
    int     d1, d2;

    d1 = 1234;
   8:   e30034d2    movw    r3, #1234   ; 0x4d2
   c:   e58d3004    str r3, [sp, #4]
    d2 = d1 / test_1;
  10:   e59d2004    ldr r2, [sp, #4]
  14:   e3003000    movw    r3, #0
  18:   e3403000    movt    r3, #0
  1c:   e5933000    ldr r3, [r3]
  20:   e1a00002    mov r0, r2
  24:   e1a01003    mov r1, r3
  28:   ebfffffe    bl  0 <__aeabi_uidiv>
  2c:   e1a03000    mov r3, r0
  30:   e58d3000    str r3, [sp]

    return d2;
  34:   e59d3000    ldr r3, [sp]
}
  38:   e1a00003    mov r0, r3
  3c:   e28dd00c    add sp, sp, #12
  40:   e49df004    pop {pc}        ; (ldr pc, [sp], #4)
```

> ccflags-y += -O1 -march=armv7-a

```
00000000 <init_module>:

static uint test_1 = 0;
module_param(test_1, uint, 0644);

static int __init test_module_init(void)
{
   0:   e92d4008    push    {r3, lr}
    int     d1, d2;

    d1 = 1234;
    d2 = d1 / test_1;
   4:   e3003000    movw    r3, #0
   8:   e3403000    movt    r3, #0
   c:   e30004d2    movw    r0, #1234   ; 0x4d2
  10:   e5931000    ldr r1, [r3]
  14:   ebffffe     bl  0 <__aeabi_uidiv>

    return d2;
}
  18:   e8bd8008    pop {r3, pc}
```

> ccflags-y += -O2 -march=armv7-a

```
00000000 <init_module>:

static uint test_1 = 0;
module_param(test_1, uint, 0644);

static int __init test_module_init(void)
{
   0:   e92d4008    push    {r3, lr}
    int     d1, d2;

    d1 = 1234;
    d2 = d1 / test_1;
   4:   e3003000    movw    r3, #0
   8:   e3403000    movt    r3, #0
   c:   e30004d2    movw    r0, #1234   ; 0x4d2
  10:   e5931000    ldr r1, [r3]
  14:   ebffffe     bl  0 <__aeabi_uidiv>

    return d2;
}
  18:   e8bd8008    pop {r3, pc}
```

即使是整数/,kernel也调用了 `__aeabi_uidiv` 库函数。

kernel code中对/(除法)的禁止真是彻底。