目前kernel中不支持除法操作。

```
static int __init test_module_init(void)
{
    uint32_t remiander;
    uint32_t n32;
    uint32_t divisor;

    double d_64;
    double d_divisor;
    double d_quotient;


    n32 = 64;
    divisor = 3;
    remiander = do_div(n32, divisor);
    printk("%u / %u = %u (%u)\n", 64, divisor, n32, remiander);

    printk("%u / %u = %u (%u)\n", 64, divisor, 64 / divisor, 64 % divisor);

    n32 = 12345678;
    divisor = 7313;
    remiander = do_div(n32, divisor);
    printk("%u / %u = %u (%u)\n", 12345678, divisor, n32, remiander);
    printk("%u / %u = %u (%u)\n", 12345678, divisor, 12345678 / divisor, 12345678 % divisor);

    d_64 = 12345678;
    d_divisor = 7313;
    d_quotient = d_64 / d_divisor;

    printk("%f / %f = %f\n", d_64, d_divisor, d_quotient);

    return 0;
}
```

运行结果如下

```
64 / 3 = 21 (1)
64 / 3 = 21 (1)
12345678 / 7313 = 1688 (1334)
12345678 / 7313 = 1688 (1334)
%f / %f = %f
```

对于如下的真正希望的除法操作

```
    d_64 = 12345678;
    d_divisor = 7313;
    d_quotient = d_64 / d_divisor;

    printk("%f / %f = %f\n", d_64, d_divisor, d_quotient);
```

反汇编后的code如下

```
    d_64 = 12345678;
    d_divisor = 7313;
    d_quotient = d_64 / d_divisor;

    printk("%f / %f = %f\n", d_64, d_divisor, d_quotient);

 12c:   e3000000    movw    r0, #0
 130:   e3083c29    movw    r3, #35881  ; 0x8c29
 134:   e3a08000    mov r8, #0
 138:   e34c4af7    movt    r4, #51959  ; 0xcaf7
 13c:   e3a02103    mov r2, #-1073741824    ; 0xc0000000
 140:   e3443167    movt    r3, #16743  ; 0x4167
 144:   e1cd80f0    strd    r8, [sp]
 148:   e1cd40f8    strd    r4, [sp, #8]
 14c:   e3400000    movt    r0, #0
 150:   ebfffffe    bl  0 <printk>
```

完全没有看到除法指令！！！