

03.ioctl(Input/Output control)

Unknown macro: 'html'

Excuse the ads! We need some help to keep our site up.

Unknown macro: 'html'

List

- [ioctl\(Input/Output control\)](#)
 - [DESCRIPTION](#)
 - [SYNOPSIS](#)
 - [Using macros to generate ioctl command numbers](#)
- [Example](#)
 - [Source code of module](#)
 - [chardev_ioctl](#)
 - [Source code of Test program](#)
 - [Make & Run](#)
- [References](#)

ioctl(Input/Output control)

DESCRIPTION

- [ioctl](#) .
 - [read\(\), write\(\)](#) , .
 - .
 - [SPI](#) , [I2C](#) , [read](#), [write](#) .
 - [CD-ROM](#) [ioctl](#) .

SYNOPSIS

- [ioctl](#) .
 - [fd open](#) .
 - [request](#)
 - .

ioctl()

```
#include <sys/ioctl.h>
int ioctl(int d, int request, ...);
```



- <https://en.wikipedia.org/wiki/ioctl>
- <http://man7.org/linux/man-pages/man2/ioctl.2.html>

Using macros to generate ioctl command numbers

- [Linux](#) `"/usr/include/asm/ioctl.h"` [ioctl](#) .
- .

Type

Macro	Description
_IO(int type, int number)	type, number ioctl .
_IOR(int type, int number, data_type)	ioctl
_IOW(int type, int number, data_type)	ioctl
_IORW(int type, int number, data_type)	ioctl .

- .

Argument

Argument	Description
type	• 8 . • " . • ioctl TCP IP .
number	• 8 • ' ioctl .
data_type	• .

- .
 - "SET_DATA" , , .
 - "GET_DATA" , , .

Example

```
struct ioctl_info{
    unsigned long size;
    unsigned int buf[128];
};

#define                    IOCTL_MAGIC                    'G'
#define                    SET_DATA                        _IOW(IOCTL_MAGIC, 2 , ioctl_info )
#define                    GET_DATA                        _IOR(IOCTL_MAGIC, 3 , ioctl_info )
```

- .

Macro to read field values

Function	Description
_IOC_NR()	number
_IOC_TYPE()	type
_IOC_SIZE()	data_type
_IOC_DIR()	



- <http://www.circlemud.org/jelson/software/fusd/docs/node31.html>

Example

- `ioctl` .
 - `ioctl()` .
- `ioctl()` I2C, GPIO, .



- KR
 - http://forum.falinux.com/zbxe/index.php?mid=lecture_tip&search_target=nick_name&search_keyword=%EC%9D%B4%EC%9A%B0%EC%98%81&page=3&document_srl=569969
 - http://forum.falinux.com/zbxe/index.php?mid=lecture_tip&search_target=nick_name&search_keyword=%EC%9D%B4%EC%9A%B0%EC%98%81&page=3
 - <http://forum.falinux.com/pds/data-s2410/No14-1.pdf>
- JP
 - <https://qiita.com/take-iwiw/items/1fdd2e0faaaa868a2db2>

Source code of module

chardev_ioctl

- (User space) `ioctl()` `chardev_ioctl()` .
 - cmd "SET_DATA" `copy_from_user()` info .
 - cmd "GET_DATA" `copy_to_user()` Kernel info .

chardev.c

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/types.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <linux/cdev.h>
#include <linux/sched.h>
#include <linux/device.h>
#include <linux/slab.h>
#include <asm/current.h>
#include <linux/uaccess.h>

#include "chardev.h"
MODULE_LICENSE("Dual BSD/GPL");

#define DRIVER_NAME "chardev"

static const unsigned int MINOR_BASE = 0;
static const unsigned int MINOR_NUM = 1;
static unsigned int chardev_major;
static struct cdev chardev_cdev;
static struct class *chardev_class = NULL;

static int chardev_open(struct inode *, struct file *);
static int chardev_release(struct inode *, struct file *);
static ssize_t chardev_read(struct file *, char *, size_t, loff_t *);
static ssize_t chardev_write(struct file *, const char *, size_t, loff_t *);
static long chardev_ioctl(struct file *, unsigned int, unsigned long);

struct file_operations s_chardev_fops = {
    .open = chardev_open,
    .release = chardev_release,
    .read = chardev_read,
    .write = chardev_write,
}
```

```

        .unlocked_ioctl = chardev_ioctl,
};

static int chardev_init(void)
{
    int alloc_ret = 0;
    int cdev_err = 0;
    int minor = 0;
    dev_t dev;

    printk("The chardev_init() function has been called.");

    alloc_ret = alloc_chrdev_region(&dev, MINOR_BASE, MINOR_NUM, DRIVER_NAME);
    if (alloc_ret != 0) {
        printk(KERN_ERR "alloc_chrdev_region = %d\n", alloc_ret);
        return -1;
    }
    //Get the major number value in dev.
    chardev_major = MAJOR(dev);
    dev = MKDEV(chardev_major, MINOR_BASE);

    //initialize a cdev structure
    cdev_init(&chardev_cdev, &s_chardev_fops);
    chardev_cdev.owner = THIS_MODULE;

    //add a char device to the system
    cdev_err = cdev_add(&chardev_cdev, dev, MINOR_NUM);
    if (cdev_err != 0) {
        printk(KERN_ERR "cdev_add = %d\n", alloc_ret);
        unregister_chrdev_region(dev, MINOR_NUM);
        return -1;
    }

    chardev_class = class_create(THIS_MODULE, "chardev");
    if (IS_ERR(chardev_class)) {
        printk(KERN_ERR "class_create\n");
        cdev_del(&chardev_cdev);
        unregister_chrdev_region(dev, MINOR_NUM);
        return -1;
    }

    device_create(chardev_class, NULL, MKDEV(chardev_major, minor), NULL, "chardev%d", minor);
    return 0;
}

static void chardev_exit(void)
{
    int minor = 0;
    dev_t dev = MKDEV(chardev_major, MINOR_BASE);

    printk("The chardev_exit() function has been called.");

    device_destroy(chardev_class, MKDEV(chardev_major, minor));

    class_destroy(chardev_class);
    cdev_del(&chardev_cdev);
    unregister_chrdev_region(dev, MINOR_NUM);
}

static int chardev_open(struct inode *inode, struct file *file)
{
    printk("The chardev_open() function has been called.");
    return 0;
}

static int chardev_release(struct inode *inode, struct file *file)
{
    printk("The chardev_close() function has been called.");
    return 0;
}

```

```

static ssize_t chardev_write(struct file *filp, const char __user *buf, size_t count, loff_t *f_pos)
{
    printk("The chardev_write() function has been called.");
    return count;
}

static ssize_t chardev_read(struct file *filp, char __user *buf, size_t count, loff_t *f_pos)
{
    printk("The chardev_read() function has been called.");
    return count;
}

static struct iocctl_info info;
static long chardev_iocctl(struct file *filp, unsigned int cmd, unsigned long arg)
{
    printk("The chardev_iocctl() function has been called.");

    switch (cmd) {
        case SET_DATA:
            printk("SET_DATA\n");
            if (copy_from_user(&info, (void __user *)arg, sizeof(info))) {
                return -EFAULT;
            }
            printk("info.size : %ld, info.buf : %s", info.size, info.buf);
            break;
        case GET_DATA:
            printk("GET_DATA\n");
            if (copy_to_user((void __user *)arg, &info, sizeof(info))) {
                return -EFAULT;
            }
            break;
        default:
            printk(KERN_WARNING "unsupported command %d\n", cmd);

            return -EFAULT;
    }
    return 0;
}

module_init(chardev_init);
module_exit(chardev_exit);

```

- .
 - SET_DATA_IOW(,) "struct iocctl_info".
 - SET_DATA_IOR(,) "struct iocctl_info".

chardev.h

```

#ifndef CHAR_DEV_H_
#define CHAR_DEV_H_
#include <linux/ioctl.h>

struct iocctl_info{
    unsigned long size;
    char buf[128];
};

#define          IOCTL_MAGIC          'G'
#define          SET_DATA              _IOW(IOCTL_MAGIC, 2 ,struct iocctl_info)
#define          GET_DATA              _IOR(IOCTL_MAGIC, 3 ,struct iocctl_info)

#endif

```

Makefile

```
obj-m := chardev.o

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

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

Source code of Test program

- .
 - open() "/dev/chardev0" fd.
 - ioctl() fd, "SET_DATA", (&set_info).
 - ioctl() fd, "GET_DATA", (&get_info).
 - printf() .
 - close() fd.

test.c

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/ioctl.h>
#include "chardev.h"

int main()
{
    int fd;
    struct ioctl_info set_info;
    struct ioctl_info get_info;

    set_info.size = 100;
    strncpy(set_info.buf, "lazenca.0x0", 11);

    if ((fd = open("/dev/chardev0", O_RDWR)) < 0){
        printf("Cannot open /dev/chardev0. Try again later.\n");
    }

    if (ioctl(fd, SET_DATA, &set_info) < 0){
        printf("Error : SET_DATA.\n");
    }

    if (ioctl(fd, GET_DATA, &get_info) < 0){
        printf("Error : SET_DATA.\n");
    }

    printf("get_info.size : %ld, get_info.buf : %s\n", get_info.size, get_info.buf);

    if (close(fd) != 0){
        printf("Cannot close.\n");
    }
    return 0;
}
```

Make & Run

- .

Build & Run

```
lazenca0x0@ubuntu:~/Kernel/Module/ioctl$ make
make -C /lib/modules/4.18.0-11-generic/build M=/home/lazenca0x0/Kernel/Module/ioctl modules
make[1]: Entering directory '/usr/src/linux-headers-4.18.0-11-generic'
Makefile:982: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-devel or elfutils-libelf-devel"
CC [M] /home/lazenca0x0/Kernel/Module/ioctl/chardev.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/lazenca0x0/Kernel/Module/ioctl/chardev.mod.o
LD [M] /home/lazenca0x0/Kernel/Module/ioctl/chardev.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.18.0-11-generic'
lazenca0x0@ubuntu:~/Kernel/Module/ioctl$ sudo insmod chardev.ko
[sudo] password for lazenca0x0:
lazenca0x0@ubuntu:~/Kernel/Module/ioctl$ ./test
get_info.size : 100, get_info.buf : lazenca.0x0
lazenca0x0@ubuntu:~/Kernel/Module/ioctl$ dmesg |tail
[ 53.358158] rfkill: input handler disabled
[ 97.190519] chardev: loading out-of-tree module taints kernel.
[ 97.190571] chardev: module verification failed: signature and/or required key missing - tainting kernel
[ 97.191019] The chardev_init() function has been called.
[ 117.777326] The chardev_open() function has been called.
[ 117.777328] The chardev_ioctl() function has been called.
[ 117.777329] SET_DATA
[ 117.777330] info.size : 100, info.buf : lazenca.0x0
[ 117.777331] The chardev_ioctl() function has been called.
[ 117.777331] GET_DATA
lazenca0x0@ubuntu:~/Kernel/Module/ioctl$
```

References

- <https://www.wdic.org/w/TECH/ioctl>
- <http://www.circlemud.org/jelson/software/fusd/docs/node31.html>
- <https://hunn1979.blogspot.com/2016/04/ioctl.html>
- <http://guswnsla1223.tistory.com/126>
- <https://qiita.com/take-iwiw/items/ade0a73d4c05fc7961d3>



Unknown macro: 'html'