Embedded System Software

Ioctl &
Kernel memory allocation & time management

Dept. of Computer Science and Engineering
Sogang University, Seoul, KOREA

ioctl

- **Device driver 내 file_operations에 사용 함수 선언**

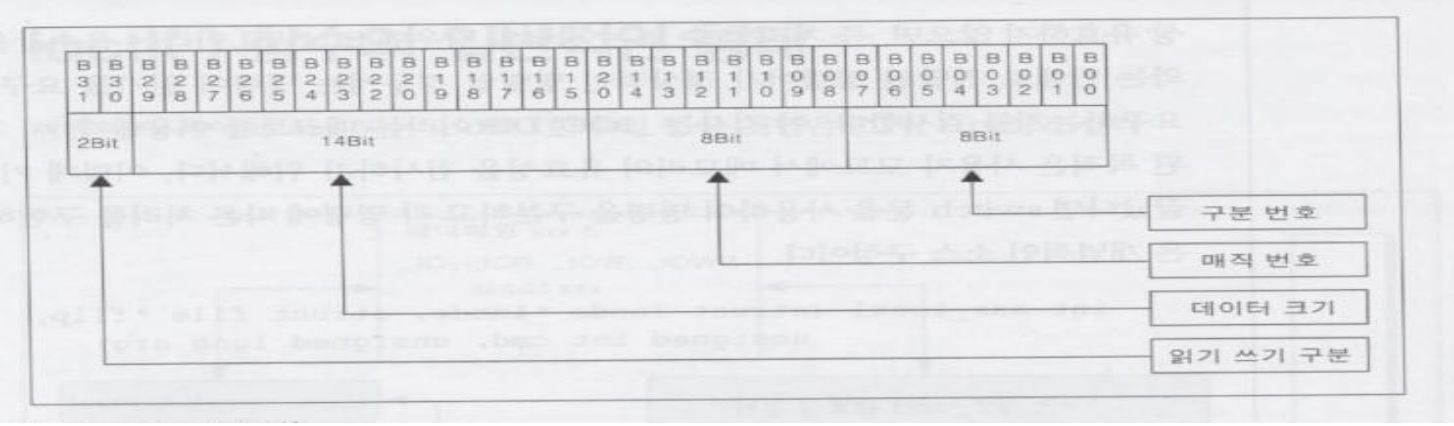
```
//file_operation structure
struct file_operations driver_fops=
{
    owner:   THIS_MODULE,
    open:    new_driver_open,
    unlocked_ioctl: new_driver_ioctl,
    release:     new_driver_release,
};
```

ioctl

- **ioctl()**

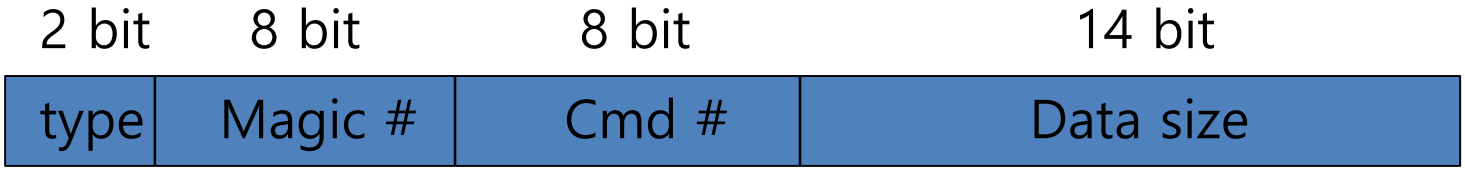
```
ret = ioctl( int fd , int request , char *argp )

long xxx_ioctl(struct file *inode, unsigned int cmd, unsigned long arg)
{
        return ret;
}
```

- **Command structure**



ioctl commands

- **Command is an integer (32 bit)**

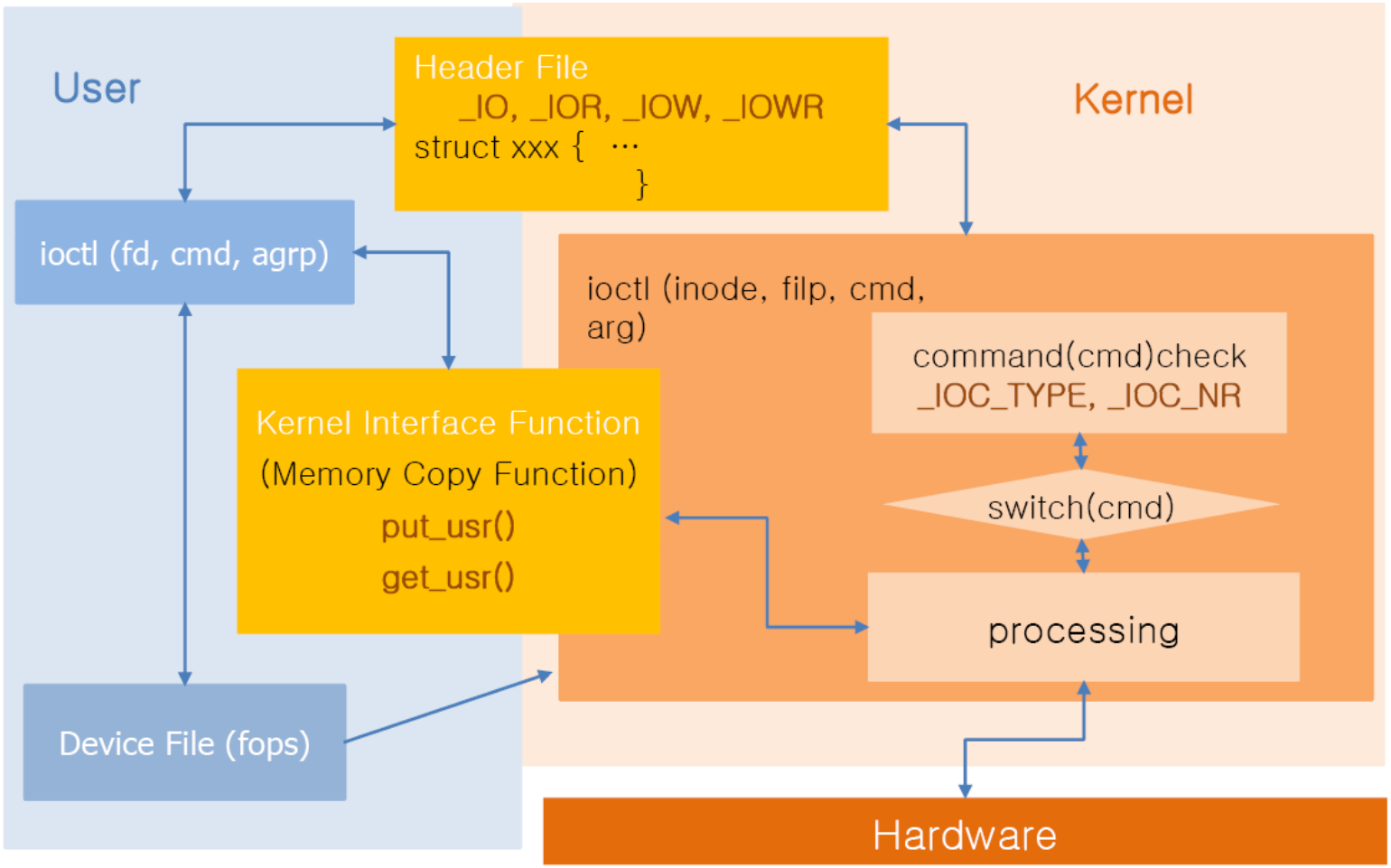


- **Example**
  - **User가 ioctl(fd , 2151694592,....) 를 호출했을 때, 그 command의 의미는**
  - **2151694592 = 10 / 00000001 / 00000001 / 0000010000000 (2진수)**
  - **Type = 2, magic # = 1, cmd # = 1, data size = 256**
- **Command를 만드는 과정이 매우 험난하다 => macro 제공**

ioctl commands

- **Macros to encode / decode a command**
  - **#include <asm/ioctl.h>**
  - **Encoding**
    - _IO(type, number)
    - _IOR(type, number, datatype)
    - _IOW(type, number, datatype)
    - _IOWR(type, number, datatype)
  - **Decoding**
    - _IOC_TYPE(command)
    - _IOC_NR(command)
    - _IOC_DIR(command)
    - _IOC_SIZE(command)

Using ioctl


User
Header File
_IO, _IOR, _IOW, _IOWR
struct xxx { …
}
Kernel
ioctl (fd, cmd, agrp)
ioctl (inode, filp, cmd, arg)
command(cmd)check
_IOC_TYPE, _IOC_NR
Kernel Interface Function
(Memory Copy Function)
put_usr()
get_usr()
switch(cmd)
processing
Device File (fops)
Hardware

ioctl commands

- **Macros to encode / decode a command**
  - **#include <asm/ioctl.h>**
  - **Encoding**
    - _IO(type, number)
    - _IOR(type, number, datatype)
    - _IOW(type, number, datatype)
    - _IOWR(type, number, datatype)
  - **Decoding**
    - _IOC_TYPE(command)
    - _IOC_NR(command)
    - _IOC_DIR(command)
    - _IOC_SIZE(command)

Kernel memory allocation

- kmalloc(), kfree()
  - #include <linux/slab.h>
  - void *kmalloc(size_t size, int flag);
  - void kfree(const void *addr);
- Flags
- GFP_KERNEL, GFP_ATOMIC, GFP_DMA, GFP_USER
- vmalloc(), vfree()
  - #include <linux/vmalloc.h>
  - void *vmalloc(unsigned long size);
  - void vfree(const void *addr);

Kernel memory allocation

- **Memory pool**
  - **#include <linux/mempool.h>**
  - **mempool_t *mempool_create(int min_nr, mempool_alloc_t *alloc_fn, mempool_free_t *free_fn, void *pool_data);**
  - **void mempool_destroy(mempool_t *pool);**
  - **typedef void *(mempool_alloc_t)(int gfp_mask, void *pool_data);**
  - **typedef void (mempool_free_t)(void *element, void *pool_data);**
  - **void *mempool_alloc(mempool_t *pool, int gfp_mask);**
  - **void mempool_free(void *element, mempool_t *pool);**

Time management (1)

- Timer Interrupt
  - HZ : 1초당 발생하는 타이머 인터럽트 수 (#define HZ 1000)
  - <include/asm/param.h>
- jiffies, jiffies_64, get_jiffies_64()
  - jiffies : kernel 2.4에서 초당 HZ값만큼 증가하는 전역 변수
  - jiffies_64 : kernel 2.6
  - jiffies값은 1/HZ초 간격으로 1씩 증가한다.
- Delaying Executing
  - (short delays) mdelay(), udelay(), ndelay()
  - (long delays) jiffies, HZ를 이용한 실행지연
- Setting / Getting System Time
  - void do_gettimeofday(struct timeval *tv)
  - int do_settimeofday(struct timespec *tv)
  - (unsigned long)mktime(year, month, day, hour, minute, second)

Time management (2)

- **Kernel Timer**
- **커널 타이머에 등록된 함수는 한번만 실행됨.**

```
#include <linux/timer.h>
struct timer_list {
                        ……
                        unsigned long expires;  // 만료 시간
                        void (*function) (unsigned long); //만료시 호출 함수
                        unsigned long data; //호출함수에 넣어지는 argument
}

void init_timer(struct timer_list *timer);
void add_timer(struct timer_list * timer);
int del_timer(struct timer_list * timer); //타이머 제거, interrupt context 내 사용가능
int del_timer_sync(struct timer_list * timer); //다른 프로세서에서 타이머 핸들러 실행
중일 경우, 핸들러 종료될 때까지 기다린 후 제거, interrupt context 내 사용불가
int mod_timer(struct timer_list * timer, unsigned long expires);
```

- **실습) kernel_timer**

- **echo "7 6 1 7" > /proc/sys/kernel/printk**
- **insmod kernel_timer.ko**
- **mknod /dev/kernel_timer c 268 0**
- (mknod /dev/[devicename] [type] [major] [minor])
- **./kernel_timer_test**