

블록 디바이스 드라이버

블록 디바이스

- Block Device
 - random access 가능
 - block 단위의 입출력이 가능한 장치
 - 버퍼 캐쉬에 의한 내부 장치 표현
 - 파일 시스템에 의해 mount 되어 관리되는 장치
- Storage Technology
 - IDE/ATA (advanced technology attachment), SATA (serial ATA)
 - ATAPI (ATA with packet interface) – CD-ROM, SCSI
 - USB mass-storage
 - RAID
 - SD/miniSD/microSD, MMC, PCMCIA/CF
 - MTD(memory technology device) - block device emulation over flash memory
 - RAM disk
 - loopback device – regular file을 block device로 사용

블록 디바이스(계속)

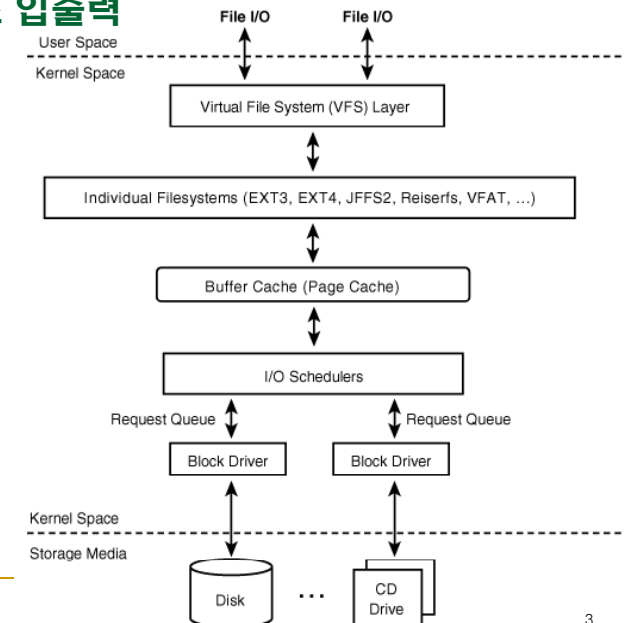
- 리눅스에서의 Block device

fd* : Floppy disk
hd* : Hard disk
sda : SCSI disk

brw-----	1	root	floppy	2,	0	May	6	1998	fd0
brw-rw----	1	root	disk	3,	0	May	6	1998	hda
brw-rw----	1	root	disk	3,	1	May	6	1998	hda1
brw-rw----	1	root	disk	8,	0	May	6	1998	sda
brw-rw----	1	root	disk	8,	1	May	6	1998	sda1

파일 관련 정보 중 첫 문자인 **b**는 block device를 의미

블록 디바이스 입출력



입출력 스케줄러

- 파일 시스템 드라이버는 블록 디바이스에 대한 입출력이 필요한 경우에 해당 디바이스에 대한 request queue에 입출력 요청을 설정함.
- 블록 디바이스 드라이버는 request queue에서 요청을 받아서 요청한 입출력 연산을 수행함.
- 입출력 스케줄러는 디스크 탐색 시간을 최소화하도록 request queue를 유지함
 - (ex) 엘리베이터 알고리즘

블록 디바이스 연산

- block device의 file 연산 정의 구조체 (include/linux/fs.h)

```
struct block_device_operations {
    int (*open) (struct inode *, struct file *);          /* Open */
    int (*release) (struct inode *, struct file *);      /* Close */
    int (*ioctl) (struct inode *, struct file *, unsigned, unsigned long); /*IOCTL*/
    int (*unlocked_ioctl) (struct inode *, struct file *, unsigned, unsigned long);
    ...
    int (*media_changed) (struct gendisk *);
        /* Check if media is available or ejected */
    int (*revalidate_disk) (struct gendisk *);
        /* Gear up for newly inserted media */
    ...
};
```

 - read, write 함수가 없음, 대신에 request 함수를 제공해야 함

블록 디바이스 드라이버의 자료구조

- gendisk 구조체 (generic disk) (include/linux/genhd.h)

```
struct gendisk {
    int major;          /* Device major number */
    int first_minor;   /* Starting minor number */
    int minors;        /* Maximum number of minors.
                        one minor number per disk partition */
    char disk_name[32]; /* Disk name */
    ...
    struct block_device_operations *fops; /* Block device operations. */
    struct request_queue *queue; /* request queue associated with
                                this disk. */
    ...
};
```

- request_queue 구조체 (include/linux/blkdev.h)
- request 구조체 (include/linux/blkdev.h)

```
struct request {
    struct request_queue *q; /* The container request queue */
    sector_t sector; /* Sector from which data access is requested */
    unsigned long nr_sectors; /* Number of sectors left to submit */
    struct bio *bio; /* The associated bio */
    char *buffer; /* The buffer for data transfer */
    struct request *next_rq; /* Next request in the queue */
};
```

블록 디바이스 드라이버 예 – RAM Disk

- 소스코드: rxd.c 참조

- 블록 디바이스 연산 정의

```
static struct block_device_operations rxdrv_fops = {  
    .owner = THIS_MODULE,  
    .unlocked_ioctl = rx_unlocked_ioctl  
};
```

- 구현된 함수

```
static long rx_unlocked_ioctl (struct file *filp, unsigned int cmd, unsigned  
    long arg) { ... }  
static void rx_request (struct request_queue *q) { ... }  
static int __init init_rxd (void) { ... }  
static void __exit exit_rxd (void) { ... }
```