# Device mapper

http://techgmm.blogspot.kr/2012/07/writing-your-own-device-mapper-target.html

# Introduction

- Generic framework to map block devices onto another
  - e.g. **LVM2**, software RAIDs, dm-crypt, multipath, …


- **/dev/mapper/***
  - List of virtual block devices managed by device mapper

- **dmsetup**
  - Tool to manipulate device mapper
  - Guide: http://linuxgazette.net/114/kapil.html

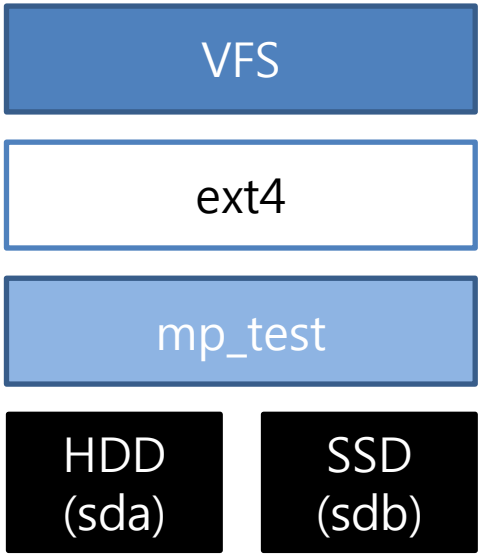
# Using device mapper

- **Installation**
  - `apt-get install dmsetup`
- **Example:** multipath device
  - `dmsetup create mp_test`

```
    0  1024  linear  /dev/sda2  1024
 1024  2048  linear  /dev/sdb2  0
```

start sector | nr sectors | target type | backing device | arguments

  - **Result:** `/dev/mapper/mp_test`




- Mapped device can be nested
  - **LVM** over **RAID**

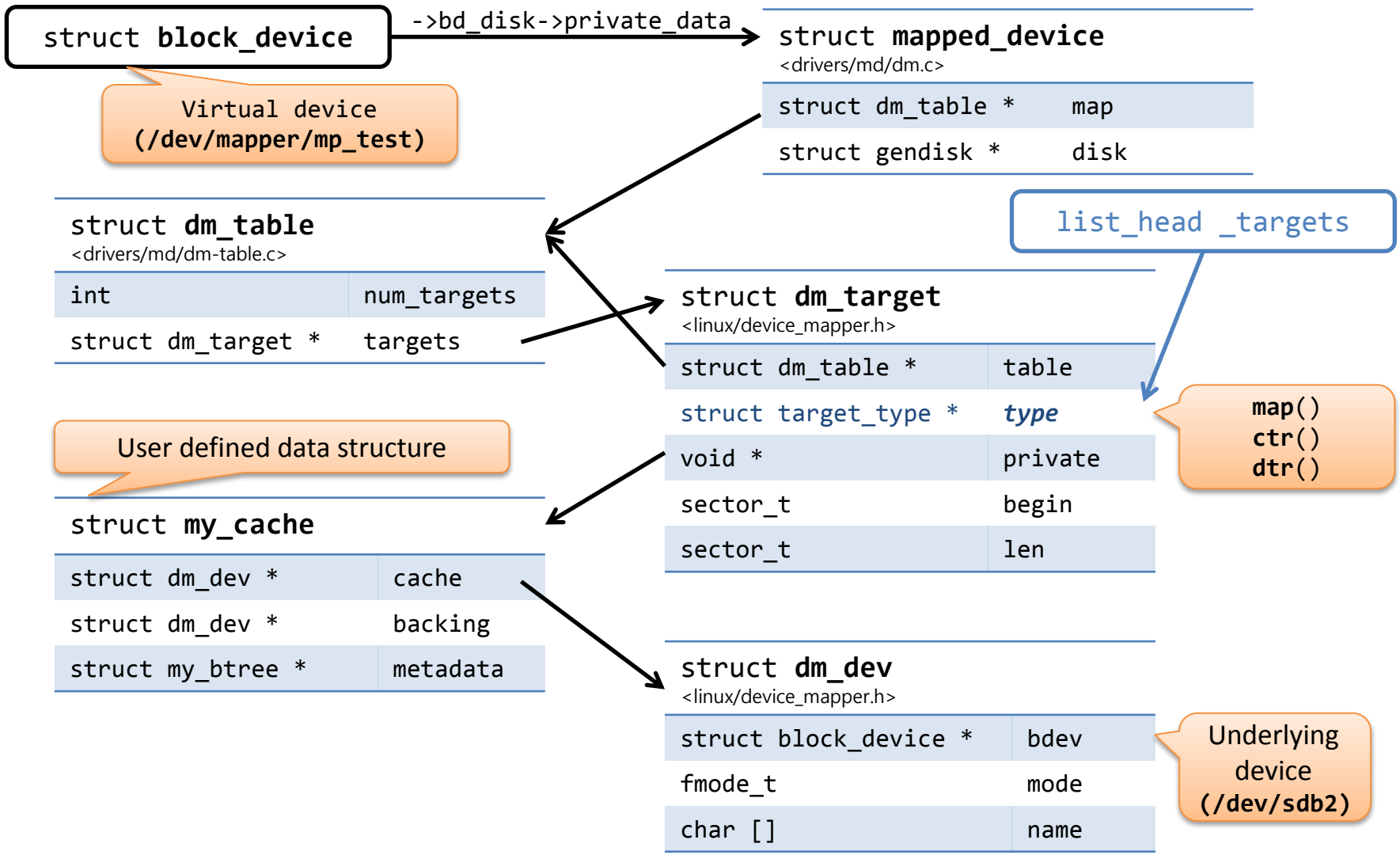

# Writing new DM target

- DM target is a **kernel module**
  - Describes mapping behavior

- **Include files**
  - **Module**: <linux/module.h>, <linux/kernel.h>, <linux/init.h>
  - **Block I/O**: <linux/bio.h>
  - **Device mapper**: <linux/device-mapper.h>

- **Method functions** to implement
  - `map()`: remapping ***bio*** to *real* target device — `submit_bio()`
  - `ctr()`: create a new virtual block device — `dmsetup create`
  - `dtr()`: stop using virtual block device — `dmsetup remove`

# Device mapper target



# Map method

- static int **map**(struct **dm_target** *ti,
  struct **bio** *bio, union **map_info** *map_context)

- **Return value**
  - **DM_MAPIO_SUBMITTED**
    - submitted bio to underlying device: used `submit_bio()`
  - **DM_MAPIO_REMAPPED**
    - bio request is remapped: changed `bi_dev`, `bi_sector`
    - Device mapper should submit bio
  - **DM_MAPIO_REQUEUE**
    - Problem with the bio, *but can finish the request with deferred retry*
    - Map method will be called again

# Map() method example

- For write request (`bio->bi_rw & 1`)
  - Write through caching
    - Allocate and initiate **`bio`** for cache device → submit_bio()
    - Allocate and initiate **`bio`** for backing device → submit_bio()
    - Return **DM_MAPIO_SUBMITTED**
- For read request
  - Cache hit
    - Update **`bi_bdev = bdev_cache, bi_sector = cache_offset`**
  - Cache miss
    - Update **`bi_bdev = bdev_backing`**
  - Return **DM_MAPIO_REMAPPED**

# Ctr / Dtr method

- static int **ctr**(struct **dm_target** *ti,
  unsigned **int** argc, **char** **argv)
  - Call **`dm_get_device`**( ) to lock backing devices
  - Initialize device data structures and worker threads
  - **Return value**
    - 0 on success
    - Negative on error
- static void **dtr**(struct **dm_target** *ti)
  - Call **`dm_put_device`**( ) function
  - Free device data structures

# Registering target

- Enables the mapping method available to the kernel

- int ***dm_register_target***(struct ***target_type*** *)
  - struct ***target_type***:
    - name = “target name”
    - version = { version }
    - module = THIS_MODULE
    - **ctr(), dtr(), map(),** …
  - Returns 0 on success

- void ***dm_unregister_target***(struct ***target_type*** *)