

디스크 관리

Hadoop

Byeongjoon Noh

powernoh@sch.ac.kr



Contents

1. 디스크와 파티션
2. 디스크 추가
3. 사용자별 공간 할당
4. RAID

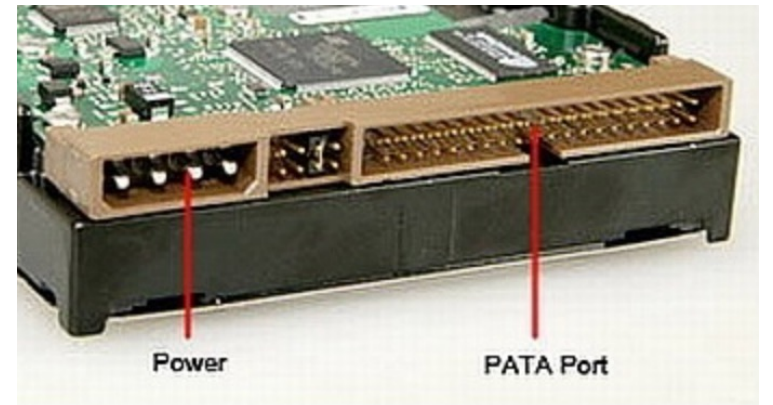
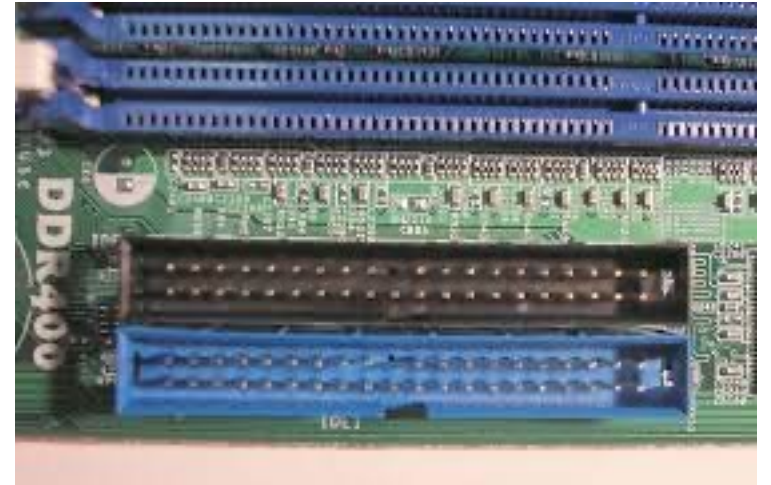
1. 디스크와 파티션

컴퓨터 저장 장치 및 인터페이스

- IDE (Integrated Drive Electronics)
- SATA (Serial ATA)
- SCSI (Small Computer System Interface)
- SAS (Serial Attached SCSI)
- SSD (Solid State Drive)

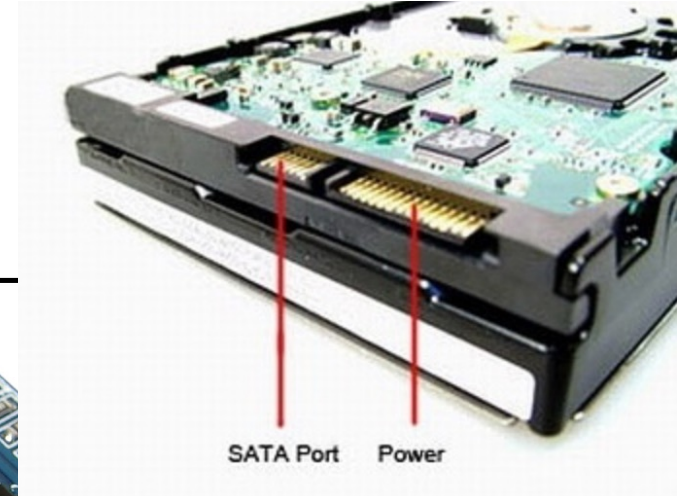
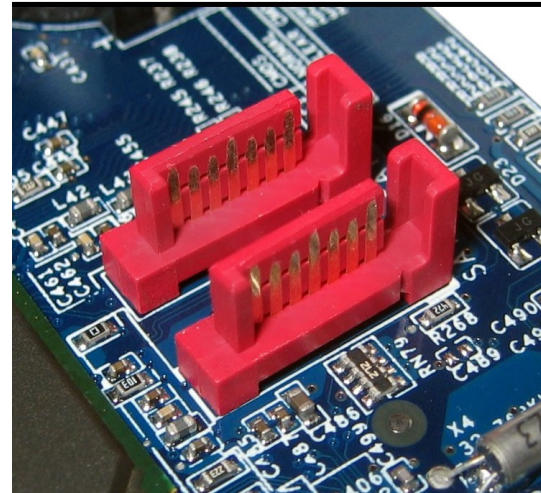
컴퓨터 저장 장치 및 인터페이스

- IDE (Integrated Drive Electronics)
 - 하드 드라이브와 컴퓨터 메인보드를 연결하는 인터페이스 표준
 - 데이터 전송을 위한 컨트롤러를 하드 드라이브 자체에 통합
 - 1980년대 후반에 등장
 - PATA (Parallel ATA)라고도 불림



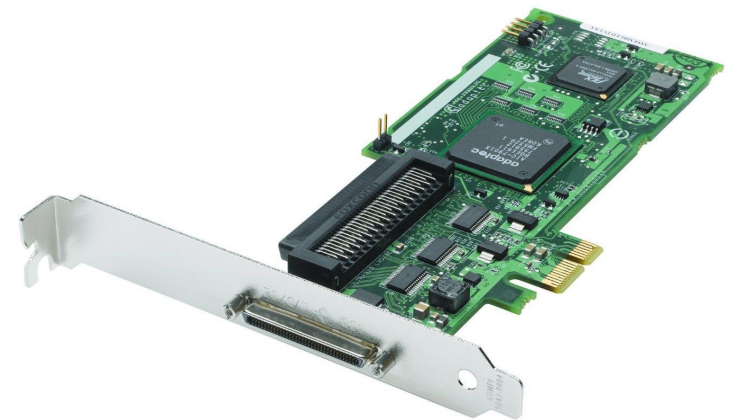
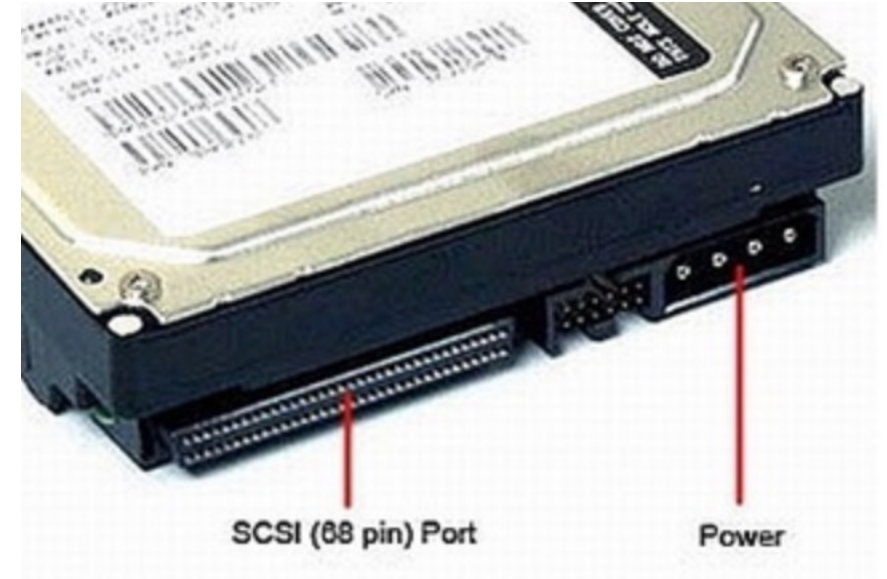
컴퓨터 저장 장치 및 인터페이스

- SATA (Serial ATA)
 - IDE의 후속
 - 데이터 저장 장치를 위한 인터페이스 표준
 - IDE 보다 빠른 데이터 전송 속도 제공
 - 얇고 유연한 케이블 사용 → 설치 용이
 - 2000년대 초반에 도입 (현재도 널리 사용)



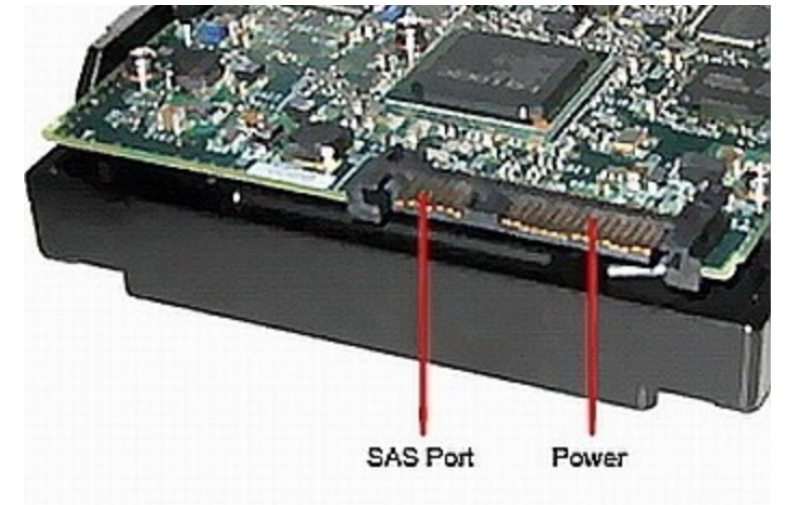
컴퓨터 저장 장치 및 인터페이스

- SCSI (Small Computer System Interface)
 - 주변 장치를 컴퓨터에 연결하기 위한 세트의 표준 인터페이스
 - 하드 드라이브, 스캐너, 프린터 등 다양한 종류의 장치 지원
 - 고성능 워크스테이션과 서버에서 선호
 - 복수의 장치를 동시에 연결할 수 있음



컴퓨터 저장 장치 및 인터페이스

- SAS (Serial Attached SCSI)
 - SCSI의 후속
 - 고성능 서버와 스토리지 네트워크에 사용되는 인터페이스



SATA 장치와 SCSI 장치 구성

- 컴퓨터 시스템 장치 구성도

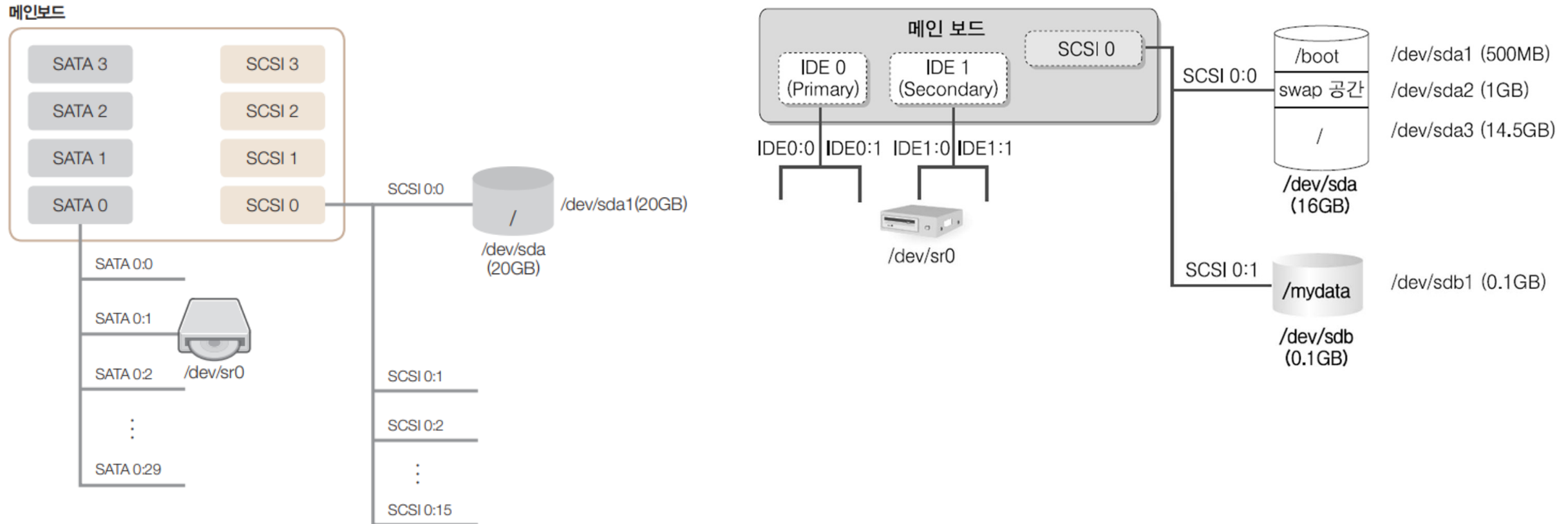


그림 9-1 Server 가상머신의 디스크 구성

SATA 장치와 SCSI 장치 구성

- 컴퓨터 시스템 장치 구성도 (VMWare)
 - 메인보드의 SATA 0번 슬롯에는 각각 30개의 SATA 장치를 장착할 수 있음
 - VMWare는 SATA슬롯 4개를 지원 → SATA 장치 120개 까지 장착 가능
 - SATA는 주로 SATA 0:0, SATA0:1, ... 로 표기
 - VMWare에서는 SATA 0:1에 CD/DVD가 장착됨
 - VMWare [Virtual Machine Settings]에서 확인

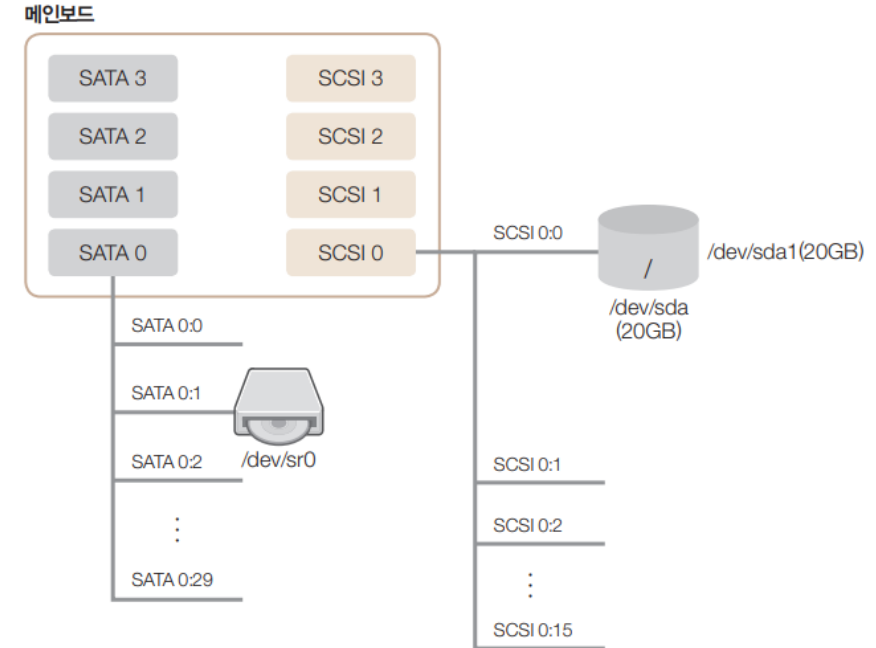
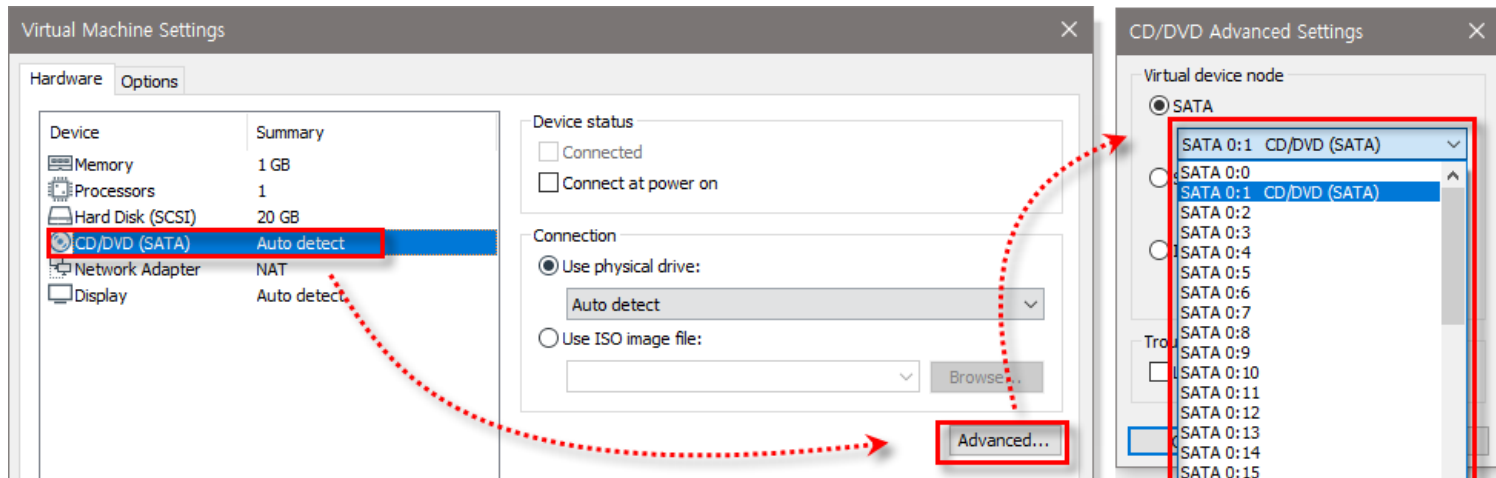
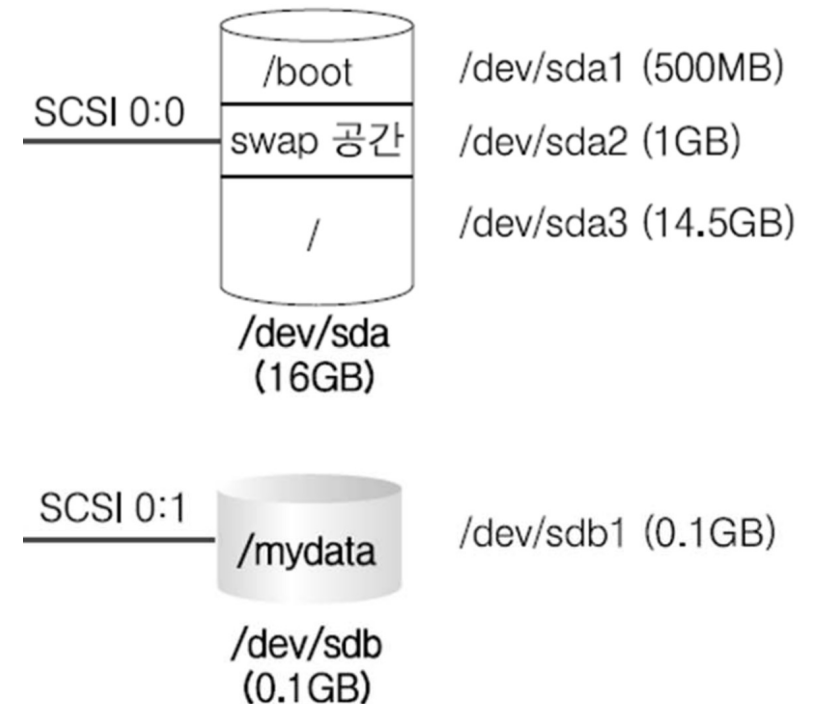
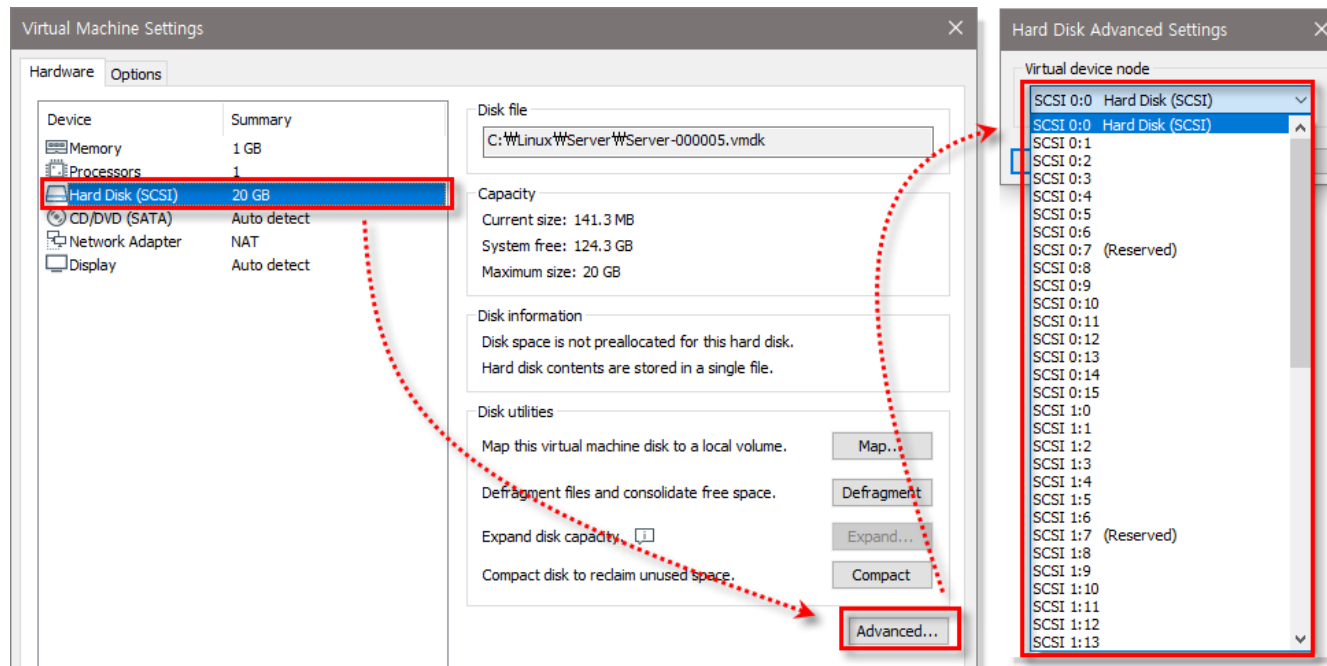


그림 9-1 Server 가상머신의 디스크 구성

SATA 장치와 SCSI 장치 구성

- 처음 장착된 SCSI 디스크를 /dev/sda, 추가로 장착된 SCSI 디스크를 /dev/sdb, /dev/sdc, /dev/sdd ...로 명명
 - 물리적인 디스크를 나눌 때 → sda, sdb, sdc, ...
- /dev/sda를 파티션(논리 공간)
 - /dev/sda1, /dev/sda2, /dev/sda3, ...



2. 디스크 추가

Linux 시스템에 디스크 추가

- 디스크가 한 개 추가된 경우 (예시)
 - 추가한 디스크의 이름: /dev/sdb
 - 파티션의 논리이름: /dev/sdb1
 - '/mydata' 디렉토리 만들고 이 디렉토리에 마운트

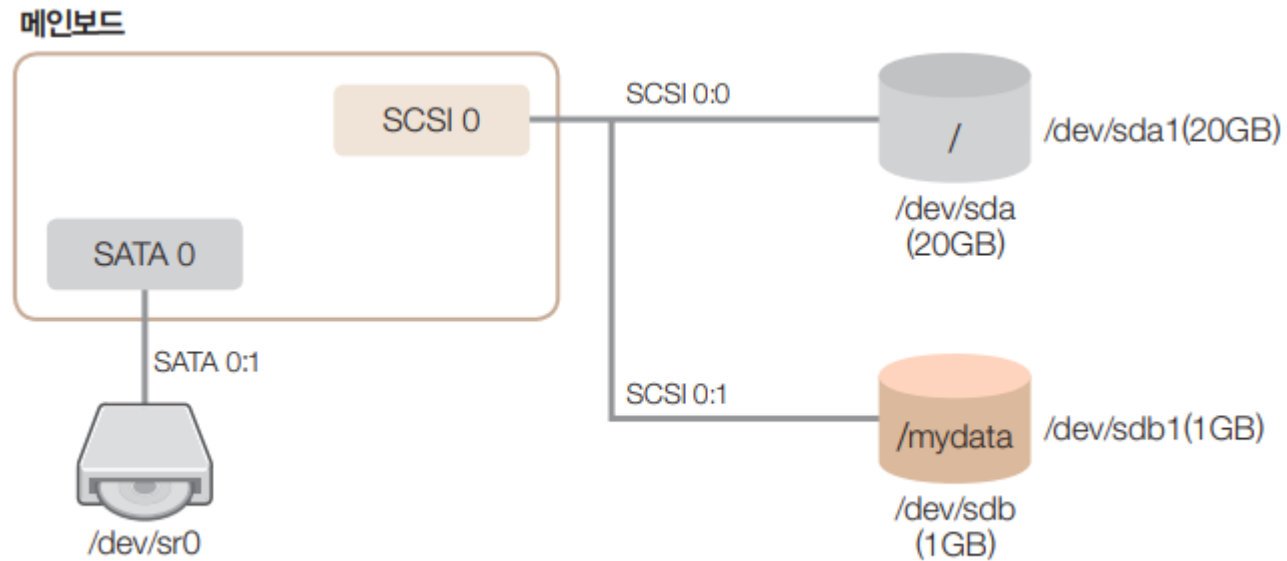


그림 9-4 디스크 하나 추가

Linux 시스템에 디스크 추가

- 디스크 추가 흐름도
 - Linux에서는 하드 디스크 (또는 SSD) 장착 후 추가적인 설정이 필요

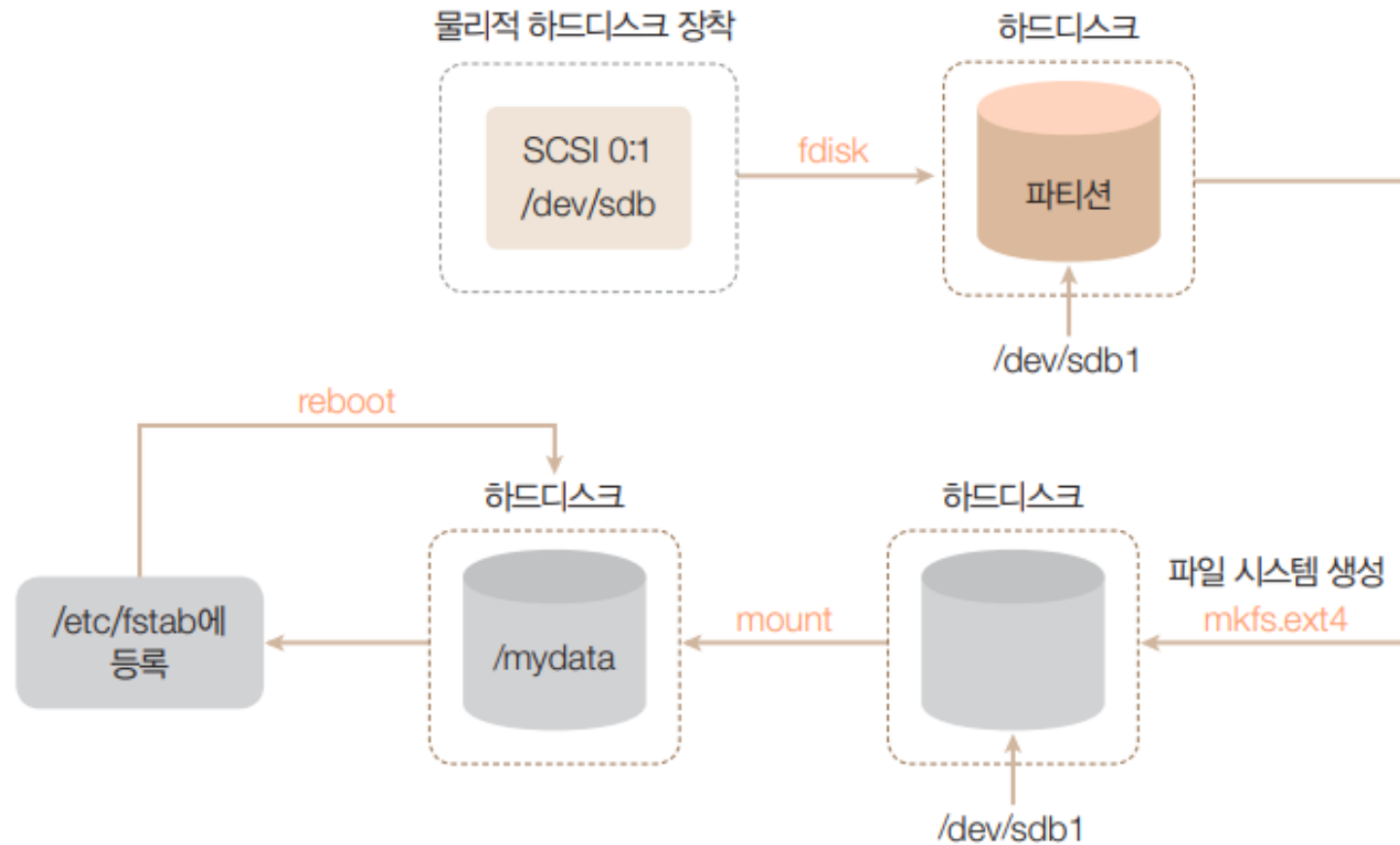
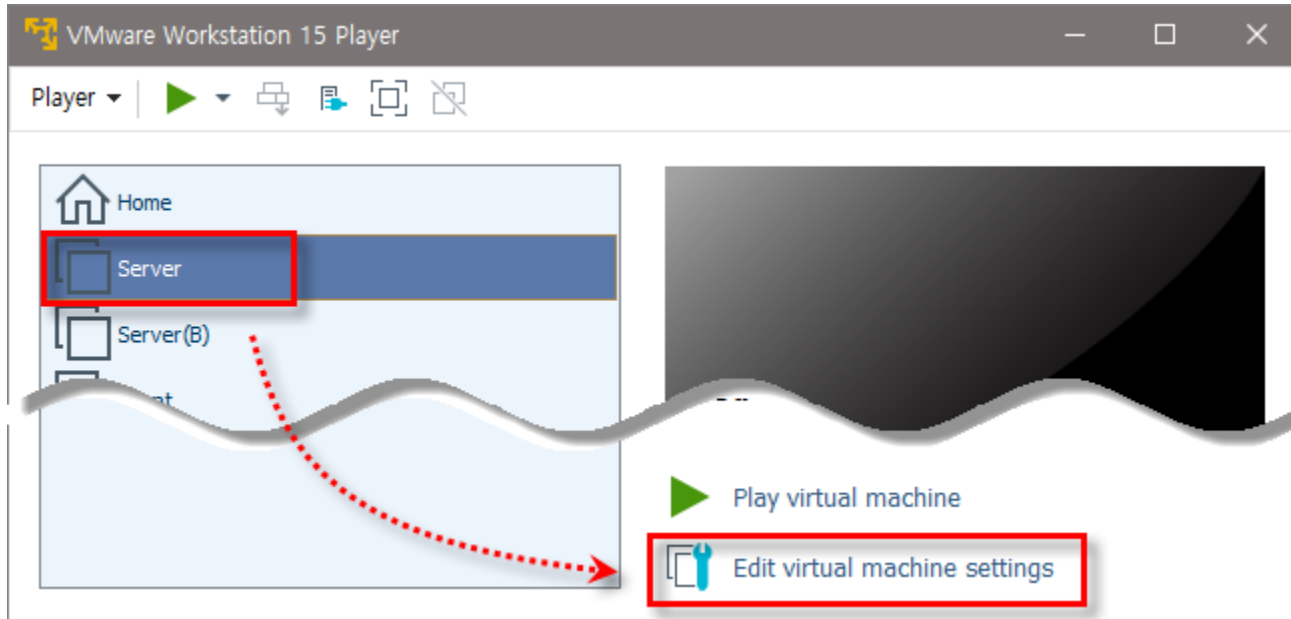


그림 9-5 디스크 하나를 추가한 경우의 전체 흐름

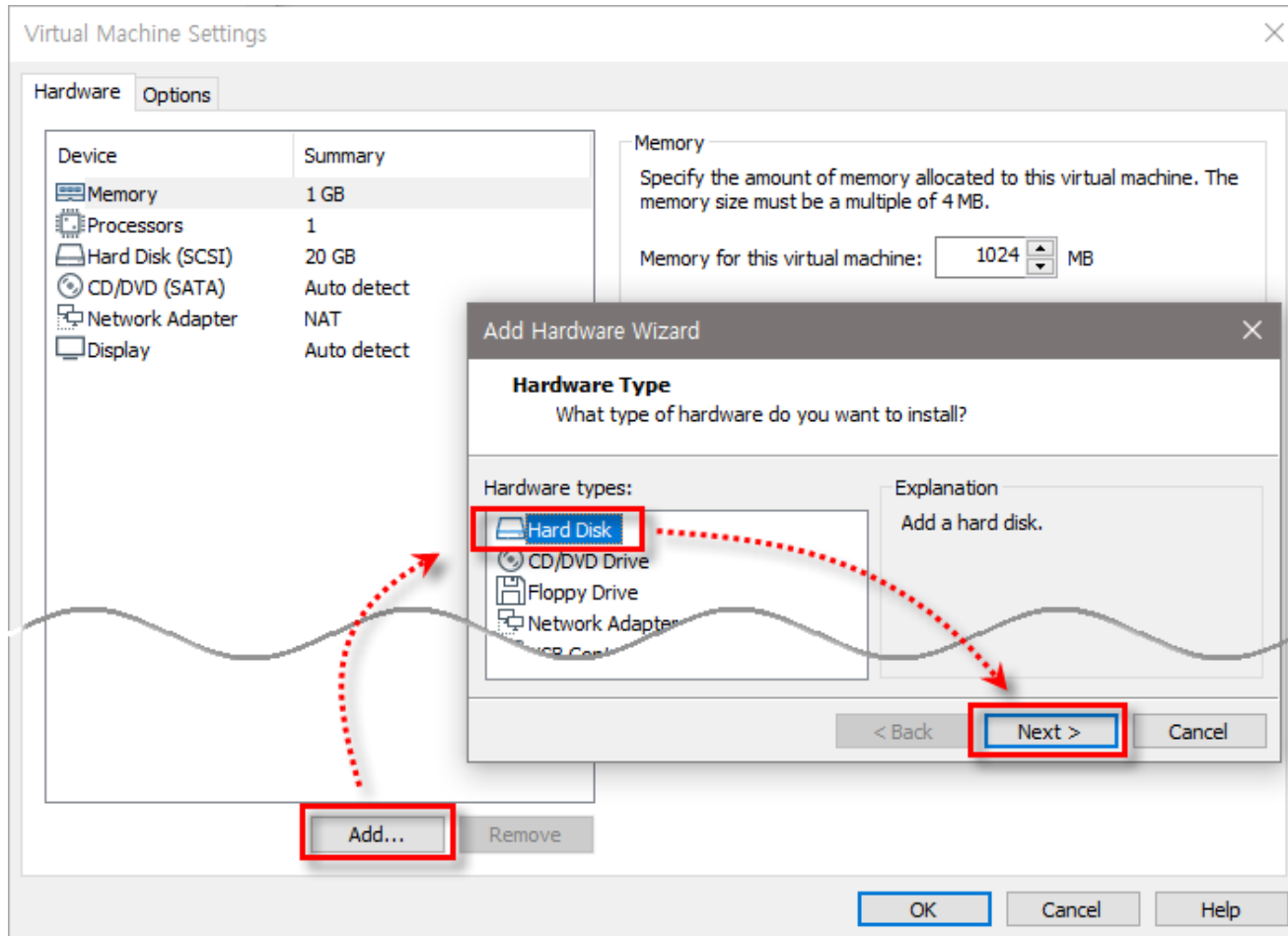
Linux 시스템에 디스크 추가 실습

- VMWare에서 하드 디스크 하나 추가



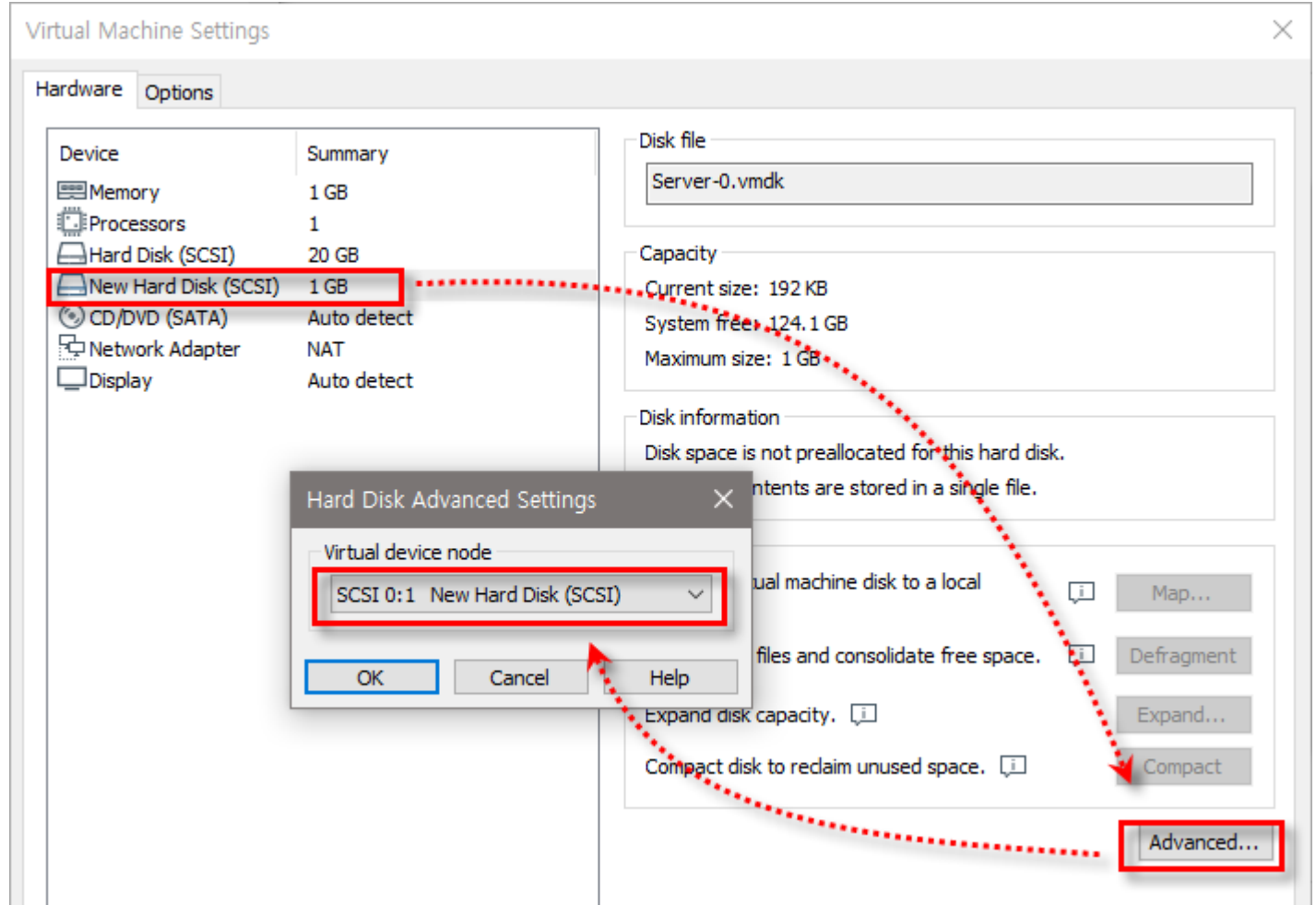
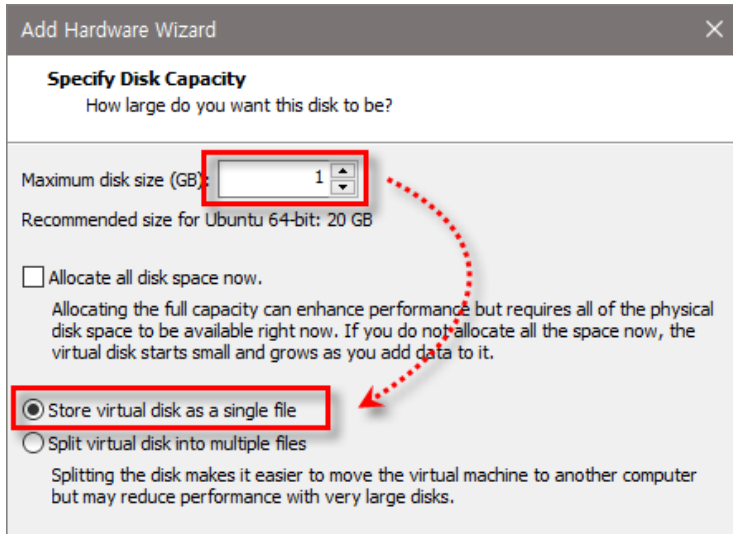
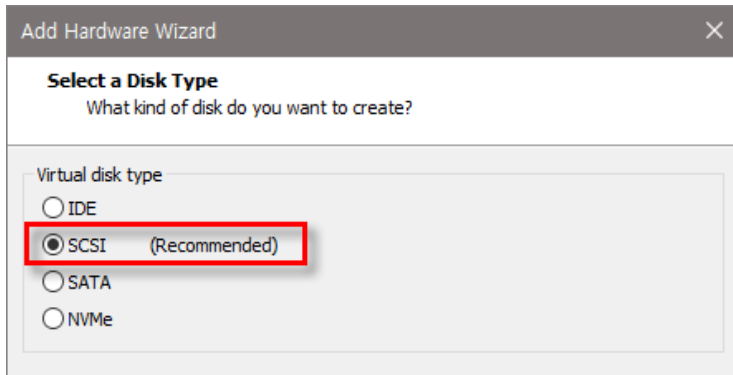
Linux 시스템에 디스크 추가 실습

- VMWare에서 하드 디스크 하나 추가



Linux 시스템에 디스크 추가 실습

- VMWare에서 하드 디스크 하나 추가 – SCSI로 추가, disk size 선택



Linux 시스템에 디스크 추가 실습

- 파티션 할당 - fdisk
 - **fdisk [옵션] 하드디스크장치파일**
 - 하드 디스크에 새로운 파티션 생성 (기존 파티션 삭제)
 - 파티션의 타입 결정 등 초기화 작업 수행
 - 한 번에 하나의 디스크에 대해서만 작업 수행
 - 터미널에서 다음과 같이 입력

```
# fdisk /dev/sdb -- SCSI 0:1 디스크 선택
Command: n -- 새로운 파티션 분할
Select: p -- Primary 파티션 선택
Partition number: 1 -- 파티션 1번 선택(Primary 파티션은 최대 4개까지 생성 가능)
First sector:  -- 시작 섹터 번호 입력(파티션 하나만 할당하므로 첫 섹터로 설정)
Last sector:  -- 마지막 섹터 번호 입력(파티션 하나만 할당하므로 마지막 섹터로 설정)
Command: p -- 설정 내용 확인
Command: w -- 설정 내용 저장
```

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x6a85bf8d.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-2097151, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151):

Created a new partition 1 of type 'Linux' and of size 1023 MiB.

Command (m for help): p
Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6a85bf8d

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1             2048 2097151 2095104 1023M 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@server:~#
```

- 새로운 파티션 이름: /dev/sdb1

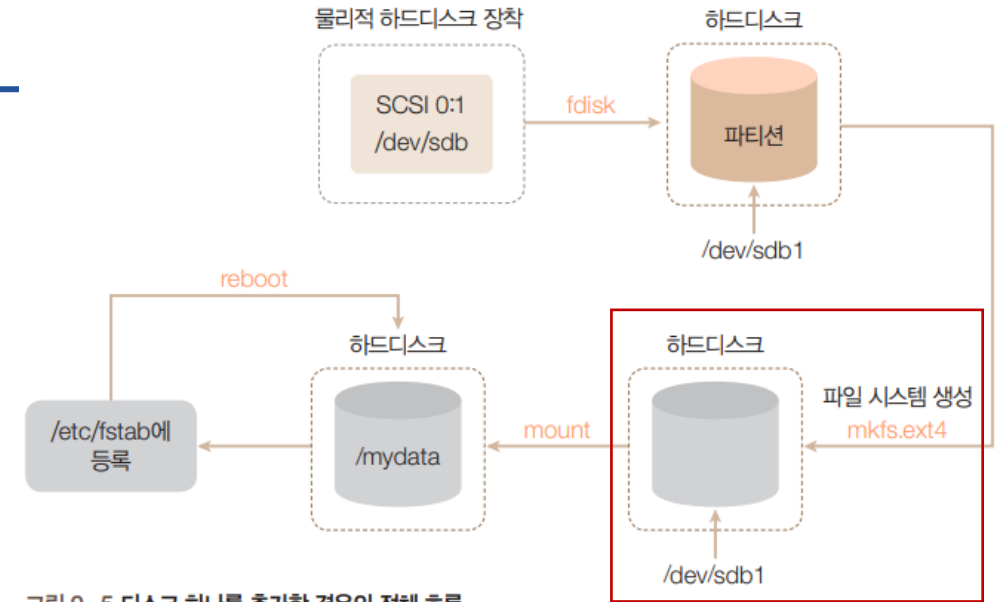
Linux 시스템에 디스크 추가 실습

- 파티션 할당 - fdisk
 - fdisk 옵션 메뉴얼

Command	action	command	action
a	부팅파티션 설정	p	현재 설정된 파티션 정보를 보여줌
b	디스크 라벨 설정	q	설정된 정보를 저장하지 않고 fdisk 빠져나감
c	마운트 가능한 파티션설정	s	SUN 디스크라벨을 생성
d	파티션 삭제	t	파티션 타입을 변경
l	설정가능한 파티션 타입을 보여줌	u	유닛 정보를 열람하거나 변경 가능
m	파티션 설정 도움말	v	지정된 파티션을 검사
n	새로운 파티션을 생성	w	현재까지 설정한 파티션정보를 저장하고 fdisk를 빠져나감.
o	도스 파티션 테이블을 생성	x	파티션 설정 전문가 모드로 들어감

Linux 시스템에 디스크 추가 실습

- 파일 시스템 생성 - mkfs
 - **mkfs [옵션] [파일시스템유형] 장치이름**
 - 하드 디스크의 파티션을 Linux 파일 시스템으로 만들기
 - **# mkfs -t ext4 /dev/sdb1** 또는
 - **# mkfs.ext4 /dev/sdb1**



```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# mkfs.ext4 /dev/sdb1
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 261888 4k blocks and 65536 inodes
Filesystem UUID: a458eb81-2957-4bc8-a3f9-b2f9a9bfff547
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

root@server:~#
```

Linux 시스템에 디스크 추가 실습

- 디렉토리에 마운트 – **mount**
 - 마운트할 디렉토리 생성 (디렉토리명: mydata)
 - `# mkdir /mydata`
 - 테스트용 파일을 하나 복사
 - `# /boot/vmTab /mydata/file1`
 - `# ls -al /mydata`

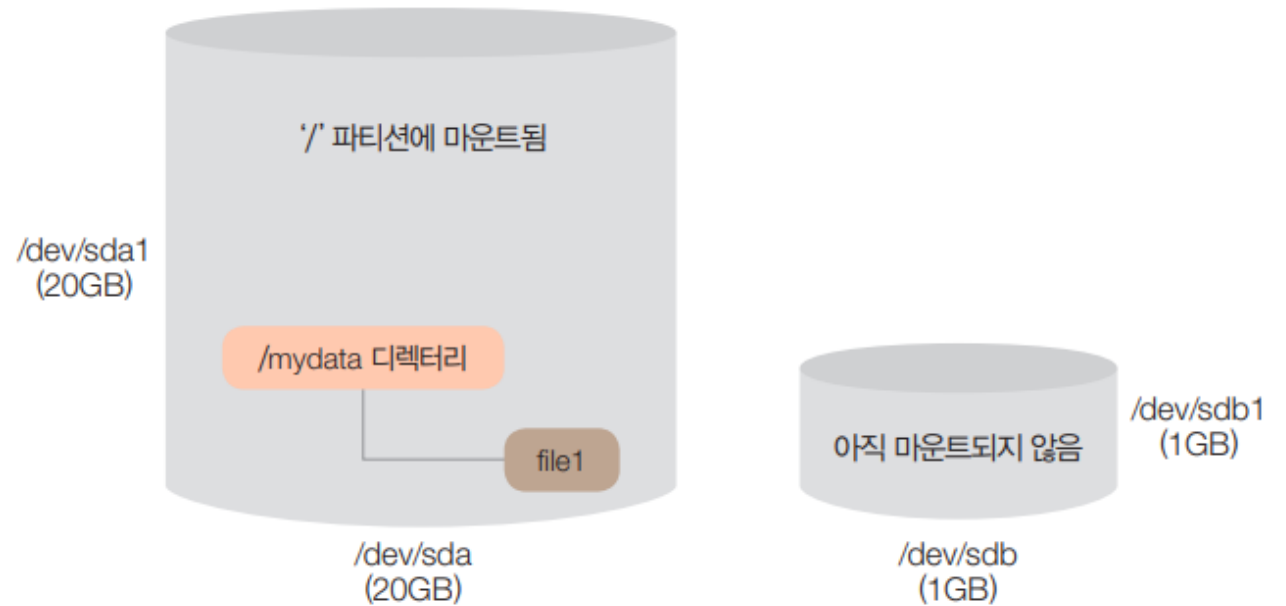
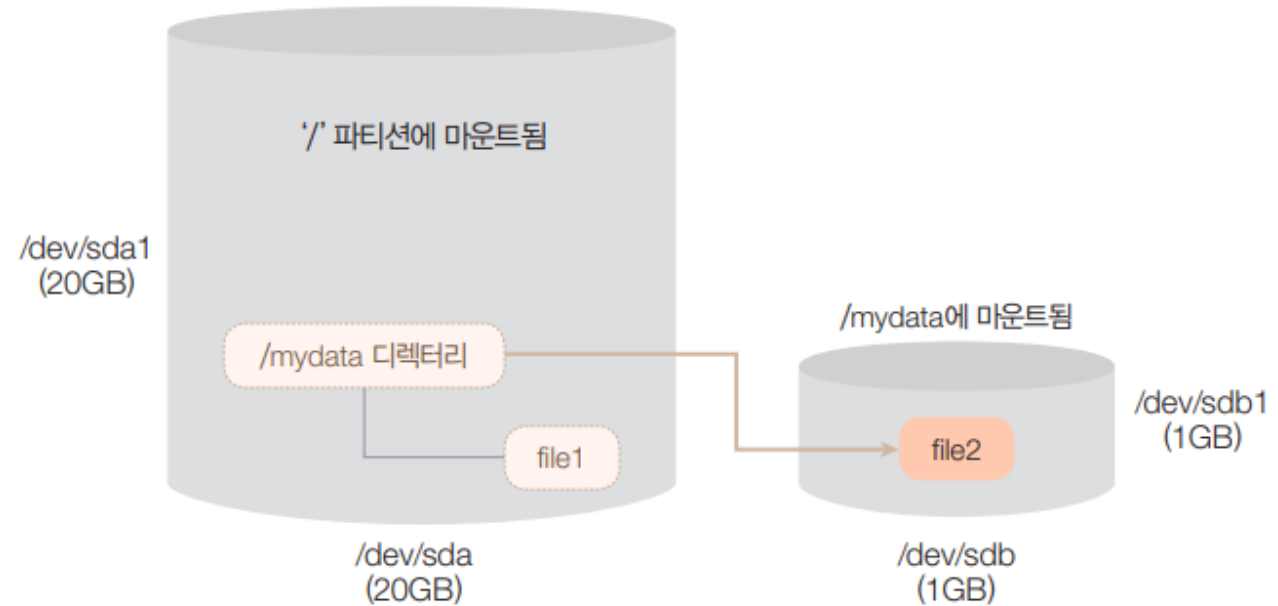


그림 9-17 마운트되기 이전의 디스크 내용

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# mkdir /mydata
root@server:~# cp /boot/vmlinuz-4.18.0-15-generic /mydata/file1
root@server:~# ls -l /mydata
합계 8344
-rw-r--r-- 1 root root 8543992 7월 20 22:20 file1
root@server:~#
```

Linux 시스템에 디스크 추가 실습

- 디렉토리에 마운트 – mount
 - /dev/sdb1 장치를 /mydata 디렉토리에 마운트
 - `# mount /dev/sdb1 /mydata`
 - /mydata 디렉토리 확인
 - `# ls -al /mydata`
 - /mydata 디렉토리에 테스트용 파일 복사
 - `# /boot/vmlinuz /mydata/file2`
 - `# ls -al /mydata`

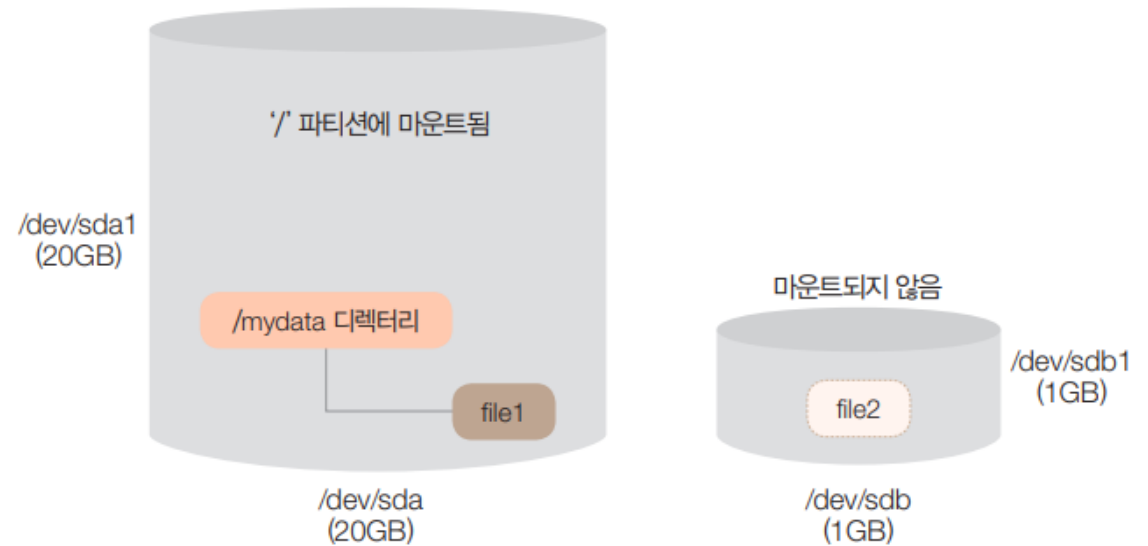


- file1의 행방은?

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# mount /dev/sdb1 /mydata
root@server:~# ls -l /mydata
합계 16
drwx----- 2 root root 16384  7월  20 22:18 lost+found
root@server:~# cp /boot/vmlinuz-4.18.0-15-generic /mydata/file2
root@server:~# ls -l /mydata
합계 8360
-rw-r--r--  1 root root 8543992  7월  20 22:24 file2
drwx----- 2 root root  16384  7월  20 22:18 lost+found
root@server:~#
```

Linux 시스템에 디스크 추가 실습

- 언마운트 (연결해제) – `umount`
 - `/dev/sdb1`의 마운트 해제
 - `# umount dev/sdb1`
 - `file1`의 유무 확인
 - `# ls -al /mydata`



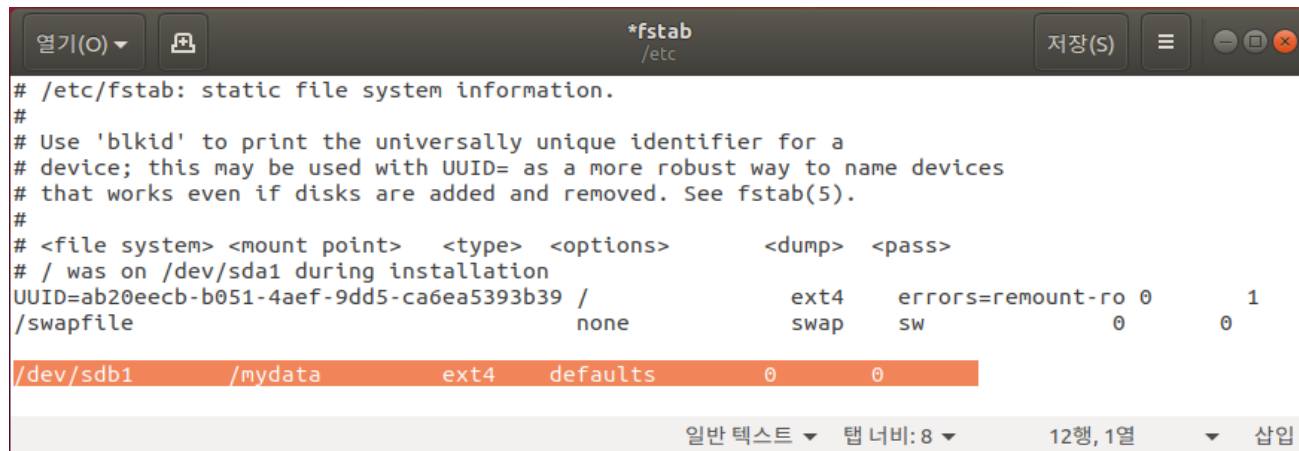
➔ 마운트 전/후로 디렉토리 내 파일이 변경됨

```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
root@server:~# umount /dev/sdb1  
root@server:~# ls -l /mydata  
합계 8344  
-rw-r--r-- 1 root root 8543992 7월 20 22:20 file1  
root@server:~#
```

Linux 시스템에 디스크 추가 실습

- 자동 마운트
 - 일반적으로 마운트 후 별도의 설정없이 재부팅 → 자동으로 마운트 해제됨
 - /etc/fstab 파일에 아래 내용을 추가하여 재부팅하여도 자동으로 마운트되도록 변경할 수 있음

```
/dev/sdb1 /mydata ext4 defaults 0 0
```



```
*fstab
/etc
저장(S)

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=ab20eecb-b051-4aef-9dd5-ca6ea5393b39 / ext4 errors=remount-ro 0 1
/swapfile none swap sw 0 0

/dev/sdb1 /mydata ext4 defaults 0 0

일반 텍스트 탭 너비: 8 12행, 1열 삽입
```

- 파일 수정 후 reboot

파일 시스템 사용량 확인

- **df [옵션] [파일시스템유형]**
 - 현재 사용 중인 파일 시스템의 사용량 확인
 - 전체 용량, 사용한 용량, 사용 가능한 용량, 사용율, 마운트 정보 등
 - 옵션
 - -k: KB 단위로 표시
 - -m: MB 단위로 표시
 - -T: 파일 시스템의 종류 표시
 - -h: 사람이 보기 편한 형태로 표시

3. 사용자별 공간 할당

공간 할당과 쿼터

- 공간 할당
 - 디스크가 꽉 차면 시스템 전체가 가동되지 않는 치명적인 문제가 발생
 - → 사용자별로 할당된 공간만 사용하도록 용량을 제한
- 쿼터
 - 각 사용자가 사용할 수 있는 파일의 용량을 제한하는 것



그림 9-24 쿼터 실습 진행 순서

사용자에게 공간 할당 실습

- 사용자 생성
 - 두 명의 사용자 생성 (linux1, linux2)
 - 비밀번호도 생성 (사용자 명과 똑같이)

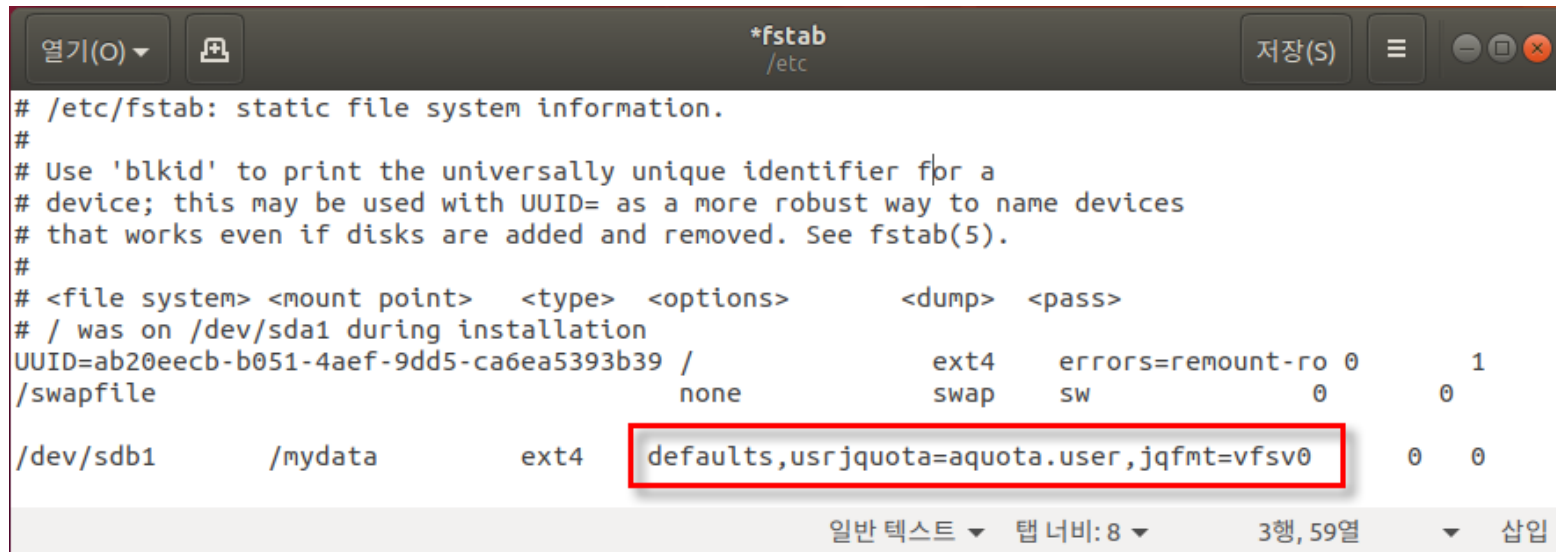
```
adduser --home /mydata/linux1 linux1 -- 암호는 linux1, 나머지는 기본 값으로 설정
adduser --home /mydata/linux2 linux2 -- 암호는 linux2, 나머지는 생략
```

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# adduser --home /mydata/linux1 linux1
'linux1' 사용자를 추가 중...
새 그룹 'linux1' (1001) 추가 ...
새 사용자 'linux1' (1001) 을(를) 그룹 'linux1' (으)로 추가 ...
'/mydata/linux1' 홈 디렉터리를 생성하는 중...
'/etc/skel'에서 파일들을 복사하는 중...
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
linux1의 사용자의 정보를 바꿉니다
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요
이름 []:
방 번호 []:
직장 전화번호 []:
집 전화번호 []:
기타 []:
정보가 올바릅니까? [Y/n]
root@server:~#
```

사용자에게 공간 할당 실습

- /etc/fstab 파일 편집
 - /dev/sdb1 마운트하는 부분에 다음을 추가

```
/dev/sdb1 /mydata ext4 defaults,usrjquota=aquota.user,jqfmt=vfsv0 0 0
```



The screenshot shows a text editor window titled '*fstab /etc'. The content of the file is as follows:

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=ab20eecb-b051-4aef-9dd5-ca6ea5393b39 / ext4 errors=remount-ro 0 1
/swapfile none swap sw 0 0
/dev/sdb1 /mydata ext4 defaults,usrjquota=aquota.user,jqfmt=vfsv0 0 0
```

The line `defaults,usrjquota=aquota.user,jqfmt=vfsv0` in the new entry is highlighted with a red box. The editor status bar at the bottom shows '일반 텍스트', '탭 너비: 8', '3행, 59열', and '삽입'.

사용자에게 공간 할당 실습

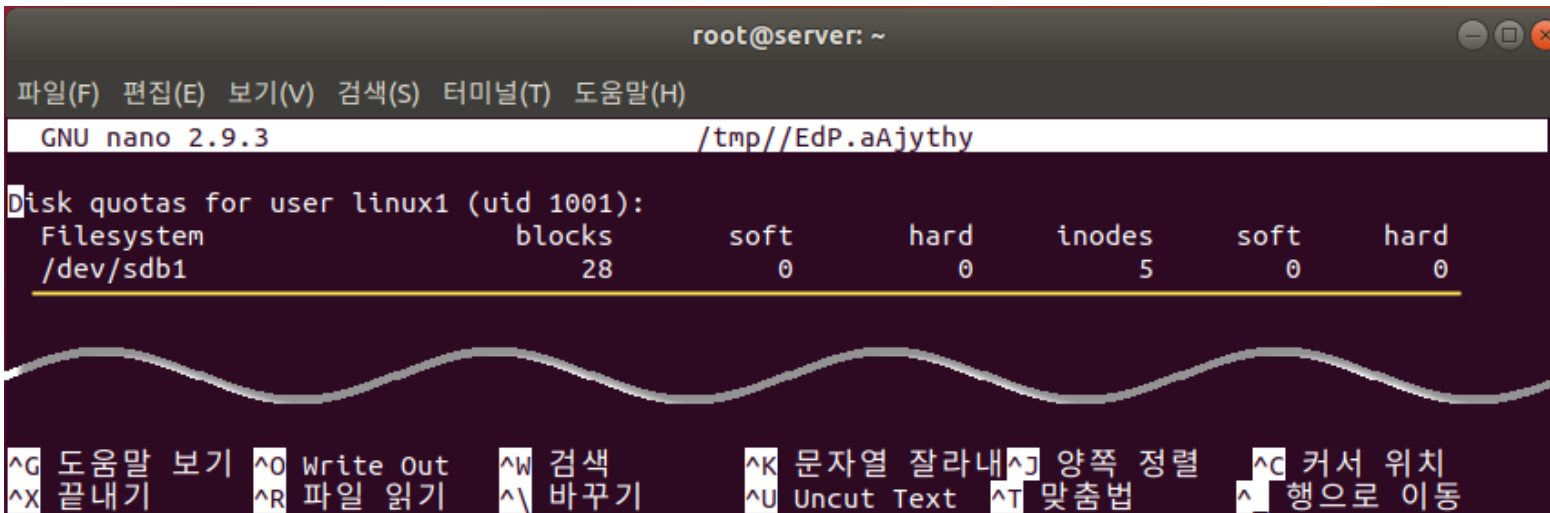
- 쿼터 DB 생성

<code>cd /mydata</code>	-- 쿼터용 파일 시스템이 마운트된 디렉터리로 이동
<code>quotaoff -avug</code>	-- 일단 쿼터 종료
<code>quotacheck -augmn</code>	-- 파일 시스템의 쿼터 관련 체크
<code>rm -f aquota.*</code>	-- 생성된 쿼터 관련 파일 삭제
<code>quotacheck -augmn</code>	-- 파일 시스템의 쿼터 관련 체크
<code>touch aquota.user aquota.group</code>	-- 쿼터 관련 파일 생성
<code>chmod 600 aquota.*</code>	-- 보안을 위해 소유자(root) 외에는 접근 금지
<code>quotacheck -augmn</code>	-- 파일 시스템의 쿼터 관련 체크
<code>quotaon -avug</code>	-- 설정된 쿼터 시작

```
root@server: /mydata
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# cd /mydata
root@server:/mydata# quotaoff -avug
/dev/sdb1 [/mydata]: user quotas turned off
root@server:/mydata# quotacheck -augmn
root@server:/mydata# rm -f aquota.*
root@server:/mydata# quotacheck -augmn
root@server:/mydata# touch aquota.user aquota.group
root@server:/mydata# chmod 600 aquota.*
root@server:/mydata# quotacheck -augmn
root@server:/mydata# quotaon -avug
/dev/sdb1 [/mydata]: user quotas turned on
root@server:/mydata# ls -l
합계 8376
-rw----- 1 root root 0 7월 20 22:58 aquota.group
-rw----- 1 root root 7168 7월 20 22:58 aquota.user
-rw-r--r-- 1 root root 8543992 7월 20 22:24 file2
drwxr-xr-x 2 linux1 linux1 4096 7월 20 22:48 linux1
drwxr-xr-x 2 linux2 linux2 4096 7월 20 23:00 linux2
drwx----- 2 root root 16384 7월 20 22:18 lost+found
root@server:/mydata#
```

사용자에게 공간 할당 실습

- 사용자별 공간 할당
 - linux1, linux2 사용자가 사용할 수 있는 공간을 각각 20MB씩 할당
 - `# edquota -u linux1`
 - nano 편집기 실행됨 → 사용자별 또는 그룹별 할당량 편집 가능



```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
GNU nano 2.9.3 /tmp//EdP.aAlythy
Disk quotas for user linux1 (uid 1001):
Filesystem      blocks      soft      hard    inodes      soft      hard
/dev/sdb1         28           0         0         5           0         0
```

^G 도움말 보기 ^O Write Out ^W 검색 ^K 문자열 잘라내기 ^J 양쪽 정렬 ^C 커서 위치
^X 끝내기 ^R 파일 읽기 ^\ 바꾸기 ^U Uncut Text ^T 맞춤법 ^_ 행으로 이동

사용자에게 공간 할당 실습

- Filesystem

- 사용자별 쿼터를 할당하는 파일 시스템



```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
GNU nano 2.9.3 /tmp//EdP.aAjythy
Disk quotas for user linux1 (uid 1001):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/sdb1        28          0         0          5           0         0
```

The screenshot shows a terminal window with the GNU nano 2.9.3 editor. The title bar indicates the user is root@server and the current file is /tmp//EdP.aAjythy. The main content displays disk quotas for user linux1 (uid 1001) on the /dev/sdb1 filesystem. The table shows that 28 blocks are currently used, with no soft or hard limits set for blocks or inodes. The bottom of the window shows various keyboard shortcuts for nano, such as ^G for help, ^O for write out, and ^X for exit.

- blocks, soft, hard

- 현재 사용자가 사용하는 블록 (KB 단위), 소프트 사용 한도, 하드 사용 한도를 의미
- blocks: 현재 사용 중인 용량 (현재 28KB를 사용 중)
- soft, hard: 0 → 사용 한도를 제한하지 않는다는 의미
- → linux1 사용자는 제한없이 /dev/sdb1 파일 시스템(/mydata) 사용 가능

- inodes, soft, hard

- inodes: inodes의 개수
- → 현재 linux1 사용자는 5개의 파일을 사용하며, 한도를 제한하지 않는다는 의미

사용자에게 공간 할당 실습

- 사용자별 공간 할당

- linux1 사용자의 사용 한도를 soft = 15360KB(15MB), hard = 20480KB(20MB)로 수정

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
GNU nano 2.9.3 /tmp//EdP.aApythy 변경됨.
Disk quotas for user linux1 (uid 1001):
Filesystem blocks soft hard inodes soft hard
/dev/sdb1 28 15360 20480 5 0 0
```

- linux1 사용자의 사용 한도 설정 확인

```
su - linux1
$ pwd
$ cp /boot/vm* test1 -- 약 8.34MB 사용
$ cp /boot/vm* test2 -- 약 16.68MB 사용, soft 한도(15MB) 초과
$ ls -l -- test2 파일의 경우 soft 한도를 초과했지만 정상적으로 복사됨
$ cp /boot/vm* test3 -- 약 25.02MB 사용, hard 한도(20MB)를 초과하여 사용 불가
$ ls -l -- test3 파일의 경우 hard 사용 한도의 남은 용량(약 3.7MB)만큼만 파일이 생성
됨, test3 파일은 정상적인 파일이 아님
```

```
linux1@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# su - linux1
linux1@server:~$ pwd
/mydata/linux1
linux1@server:~$ cp /boot/vm* test1
linux1@server:~$ cp /boot/vm* test2
linux1@server:~$ ls -l
합계 16700
-rw-r--r-- 1 linux1 linux1 8980 7월 20 22:48 examples.desktop
-rw-r--r-- 1 linux1 linux1 8543992 7월 20 23:24 test1
-rw-r--r-- 1 linux1 linux1 8543992 7월 20 23:24 test2
linux1@server:~$ cp /boot/vm* test3
cp: 'test3'에 쓰는 도중 오류 발생: 디스크 할당량이 초과됨
linux1@server:~$ ls -l
합계 20464
-rw-r--r-- 1 linux1 linux1 8980 7월 20 22:48 examples.desktop
-rw-r--r-- 1 linux1 linux1 8543992 7월 20 23:24 test1
-rw-r--r-- 1 linux1 linux1 8543992 7월 20 23:24 test2
-rw-r--r-- 1 linux1 linux1 3854336 7월 20 23:24 test3
linux1@server:~$
```

사용자에게 공간 할당 실습

- 사용자에게 할당된 디스크 공간 확인
 - linux1 사용자에게 할당된 디스크 공간확인
 - \$ quota

```
linux1@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
linux1@server:~$ quota
Disk quotas for user linux1 (uid 1001):
  Filesystem  blocks   quota  limit  grace  files   quota  limit  grace
   /dev/sdb1  20480*  15360  20480  6days     8       0     0
linux1@server:~$
```

- limit = 20480: hard 기준 사용량
- quota = 15360: soft 기준 사용량
- grace: 초과 분 (20480 - 15360)KB는 6일만 사용가능 → 이후 자동 삭제됨

사용자에게 공간 할당 실습

- 사용자별 사용량 확인
 - # repquota /mydata
 - root 권한으로 실행해야 함

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
linux1@server:~$ exit
로그아웃
root@server:~# repquota /mydata
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days

```

User		Block limits				File limits			
		used	soft	hard	grace	used	soft	hard	grace
root	--	8364	0	0		4	0	0	
linux1	+-	20480	15360	20480	6days	9	0	0	
linux2	--	28	0	0		5	0	0	

사용자에게 공간 할당 실습

- 사용자 간 동일하게 사용량 할당
 - linux1 사용자의 사용한도를 linux2 사용자에게 할당
 - `# edquota -p linux1 linux2`
- 사용자별 사용량 확인
 - `# repquota /mydata`

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# edquota -p linux1 linux2
root@server:~# repquota /mydata
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days

```

User		Block limits				File limits			
		used	soft	hard	grace	used	soft	hard	grace
root	--	8364	0	0		4	0	0	
linux1	+-	20480	15360	20480	6days	9	0	0	
linux2	--	28	15360	20480		5	0	0	

4. RAID

RAID의 개념

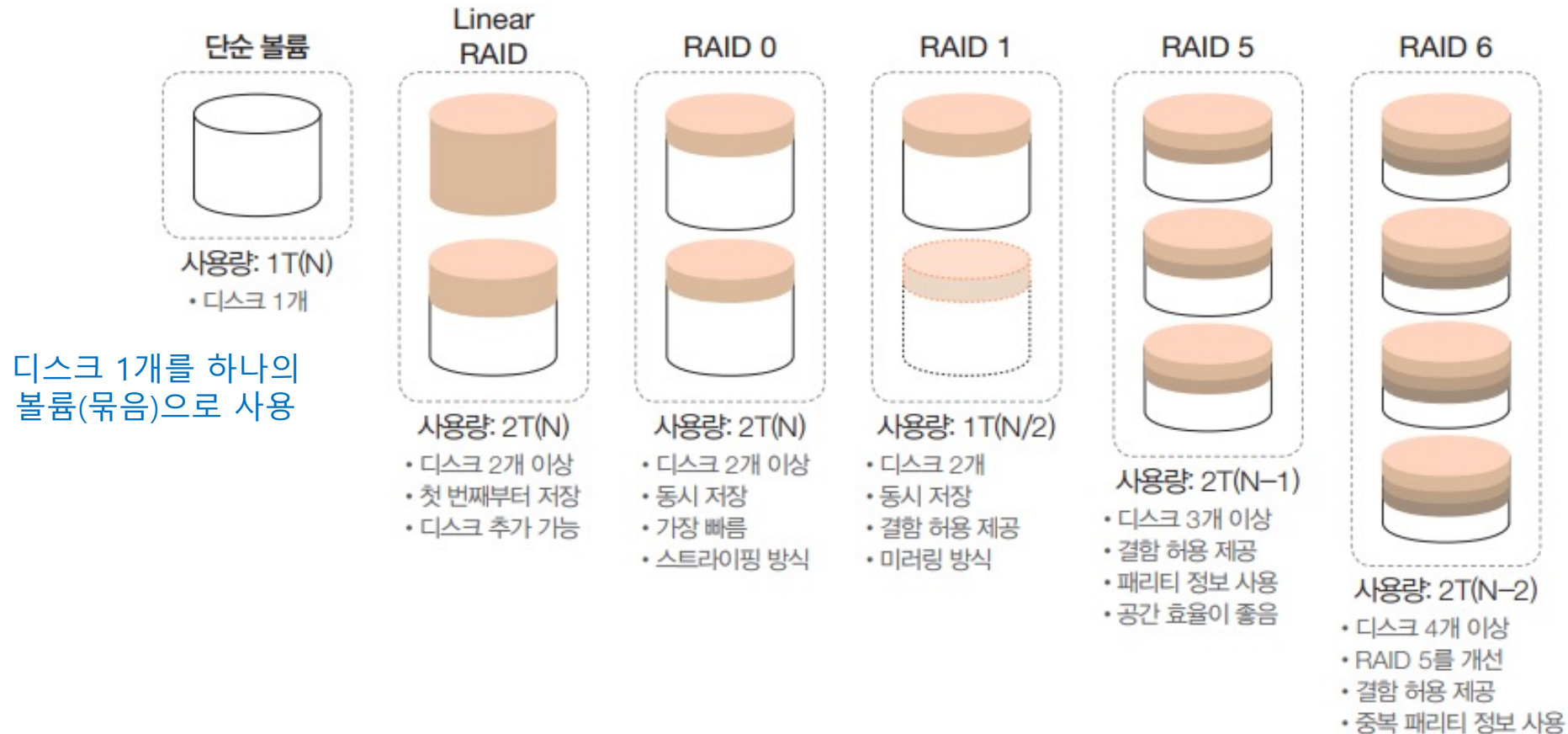
- 8TB 용량의 디스크가 필요할 때
 - (1) 2TB * 2개 + 4TB * 1개 vs (2) 8TB * 1개
 - (1) → 각 디스크 용량을 초과하지 않도록 데이터 관리 필요
- Redundant Array of Inexpensive/Independent Disks (RAID)
 - 여러 개의 디스크를 하나처럼 사용 가능

RAID의 개념

- 하드웨어 RAID
 - 하드웨어 제조 업체가 여러 개의 디스크를 연결한 장비를 만들어 공급하는 것
 - 안정적인 운용 가능
 - 제조 업체의 기술 지원 가능
 - 비싼 가격 및 제조 업체에 따라 조작 방법이 상이
- 소프트웨어 RAID
 - 고가인 하드웨어 RAID의 대안
 - 운영체제 안에서 구현되어 디스크 관리 수행
 - 하드웨어 RAID와 비교하여 신뢰성, 속도 등이 낮음

RAID의 레벨

- RAID 구성 방식
 - Linear RAID, RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5로 구분
 - Linear RAID, RAID 0, RAID 1, RAID 5 주로 사용



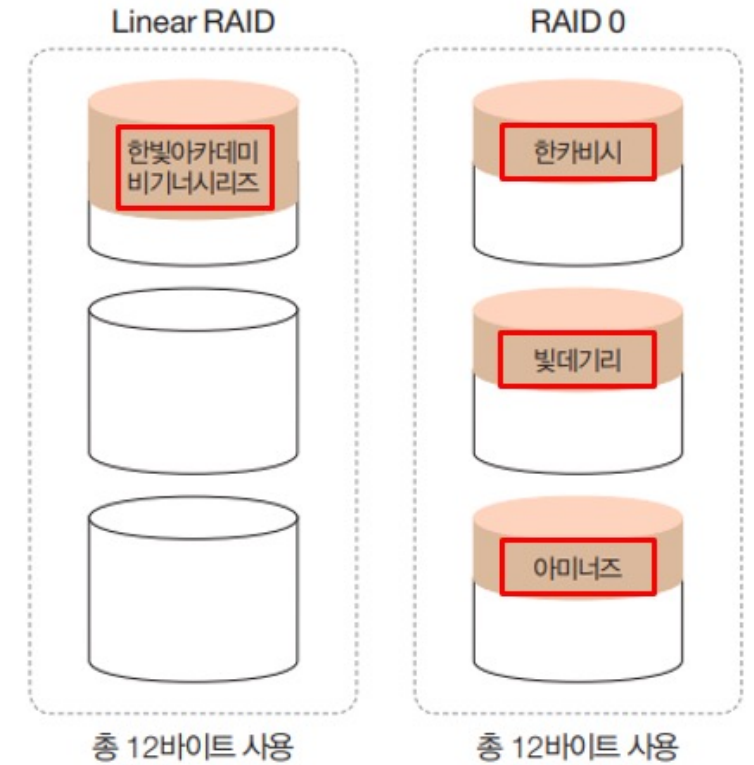
RAID의 레벨

- Linear RAID

- 2개 이상의 디스크를 하나의 볼륨으로 사용
- 앞 디스크에 데이터를 완전히 저장한 후, 다음 디스크에 저장
- ‘한빛아카데미비기너시리즈’ → 한 글자에 1초씩 총 12초 걸린다고 가정
- 한 디스크의 효율성 100%, 안전

- RAID 0

- 모든 디스크를 동시에 사용
- ‘한빛아카데미비기너시리즈’ → 동시에 디스크 3개를 사용하여 4초에 저장 가능 (striping 방식)
 - 여러 개의 디스크에 동시에 저장하는 방법
- 빠른 속도, but 디스크 하나만 고장나도 전체 데이터를 잃게됨



6. 다음 중 구성된 디스크 중에 한 개라도 오류가 발생하면 데이터 복구가 불가능한 RAID 구성법으로 알맞은 것은?

- ① RAID-0
- ② RAID-1
- ③ RAID-5
- ④ RAID-6

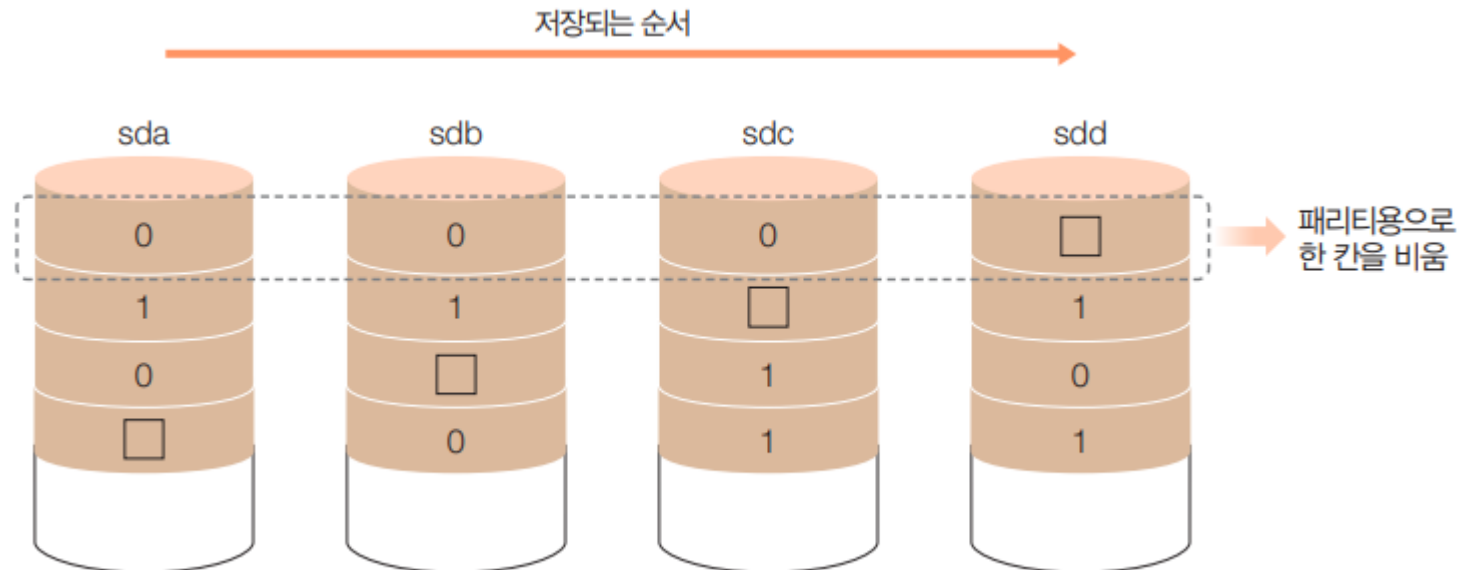
RAID의 레벨

- RAID 1
 - 미러링 (mirroring) 개념
 - 똑같은 데이터를 각 디스크에 저장
 - 공간 효율성 50%
 - 중요 데이터를 저장하기에 적당한 방식



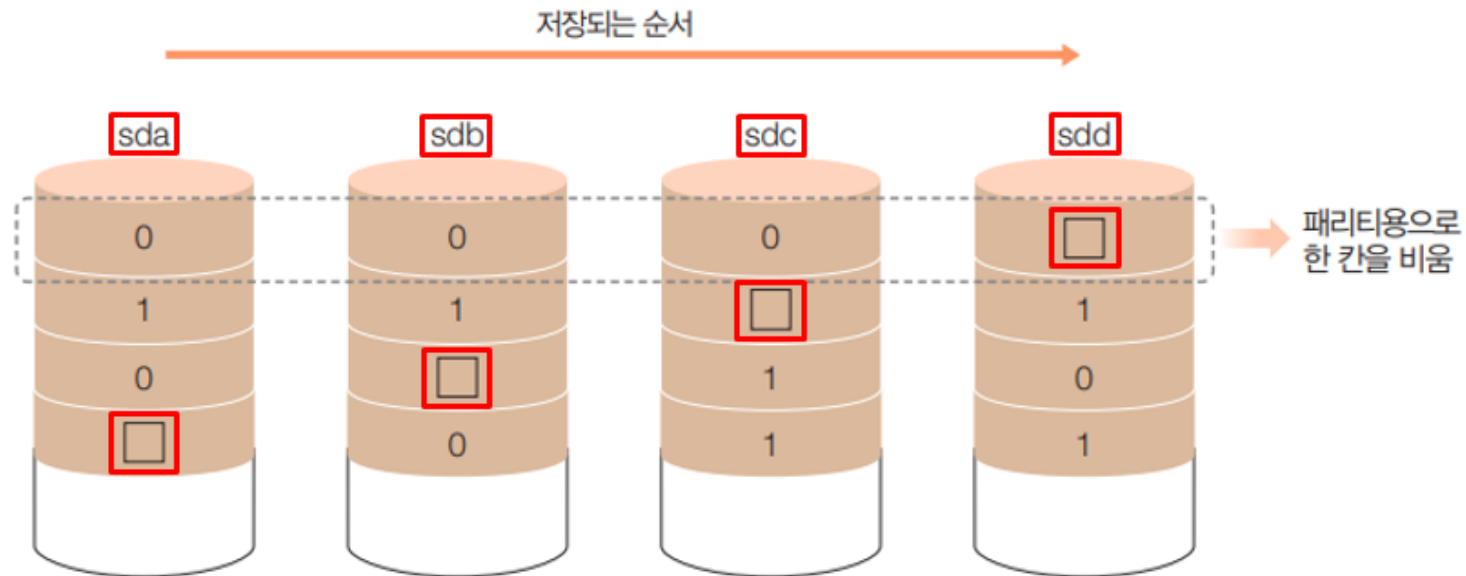
RAID의 레벨

- RAID 5
 - RAID 0과 RAID 1의 특징점 혼합
 - 디스크에 오류가 발생하면 parity(패리티) 데이터를 이용하여 데이터 복구 가능
 - 패리티: 데이터의 이동 시 데이터의 유실/손상 여부를 점검하는 기술
 - RAID 5는 최소 3개 이상의 디스크가 있어야 구성 가능 (주로 5개 이상 사용)
 - 데이터 분실 시 유용



RAID의 레벨

- RAID 5
 - 각 행이 짝수가 되도록 숫자를 채워 넣는 짝수 패리티 사용
 - 첫 번째 행의 $0+0+0+P=$ 짝수가 되어야 하므로 $P=0$
 - 세 번째 행의 $0+P+1+0=$ 짝수가 되어야 하므로 $P=1$
 - 4개의 디스크 중 하나가 고장나도 패리티를 이용하여 원래 데이터 복원 가능
 - 어느 정도의 결함 허용 + 저장 공간 효율성 확보

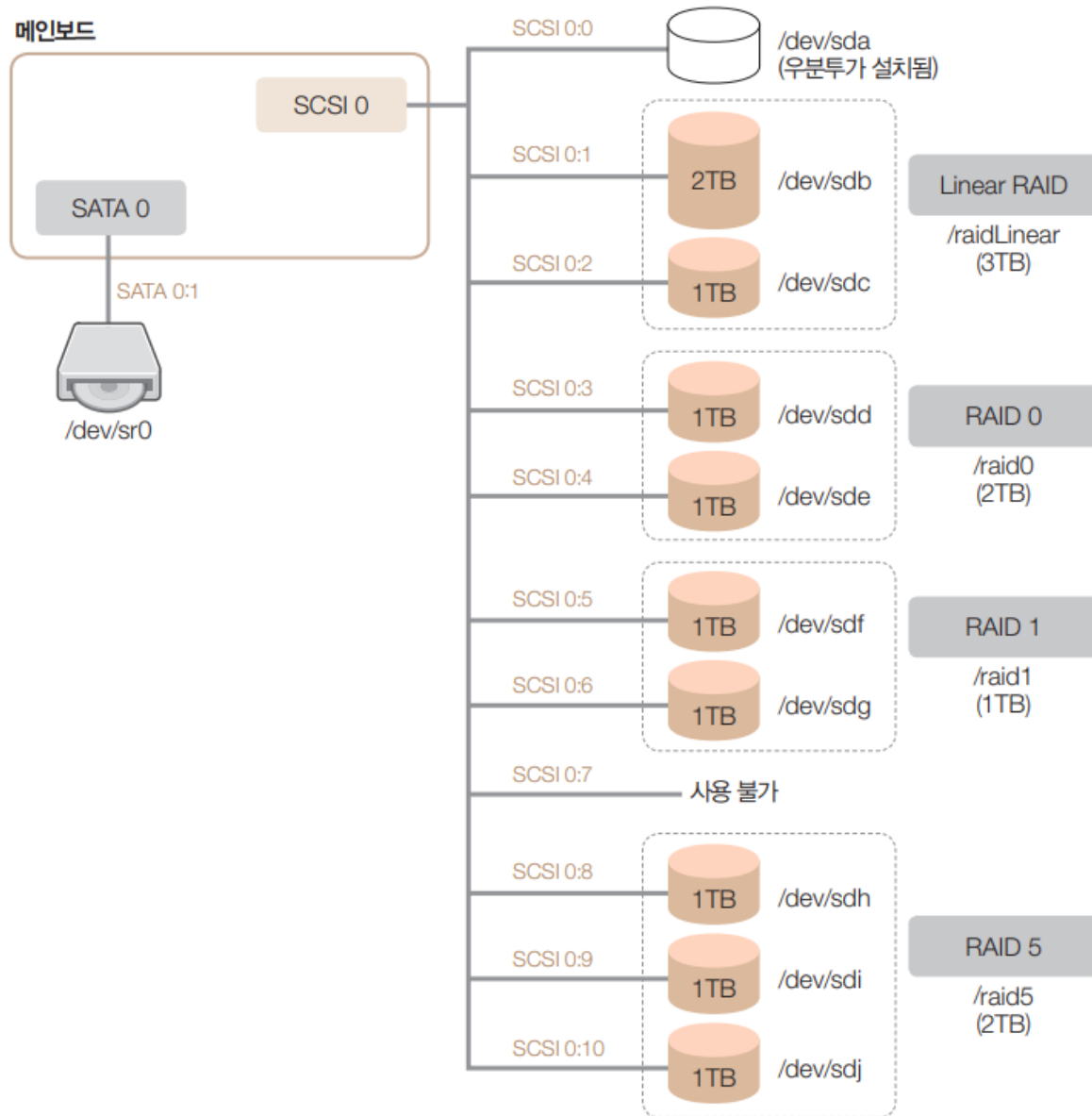


RAID의 레벨

- RAID 6
 - RAID 5를 일부 개선하여 2개의 패리티 사용
 - 디스크 10개로 구성된 RAID 5에서 동시에 2개 가 고장난다면 복원 불가능하다는 단점을 보완
 - 내부적인 쓰기 알고리즘이 복잡, 성능(속도)가 떨어짐

RAID 실습 환경 구성

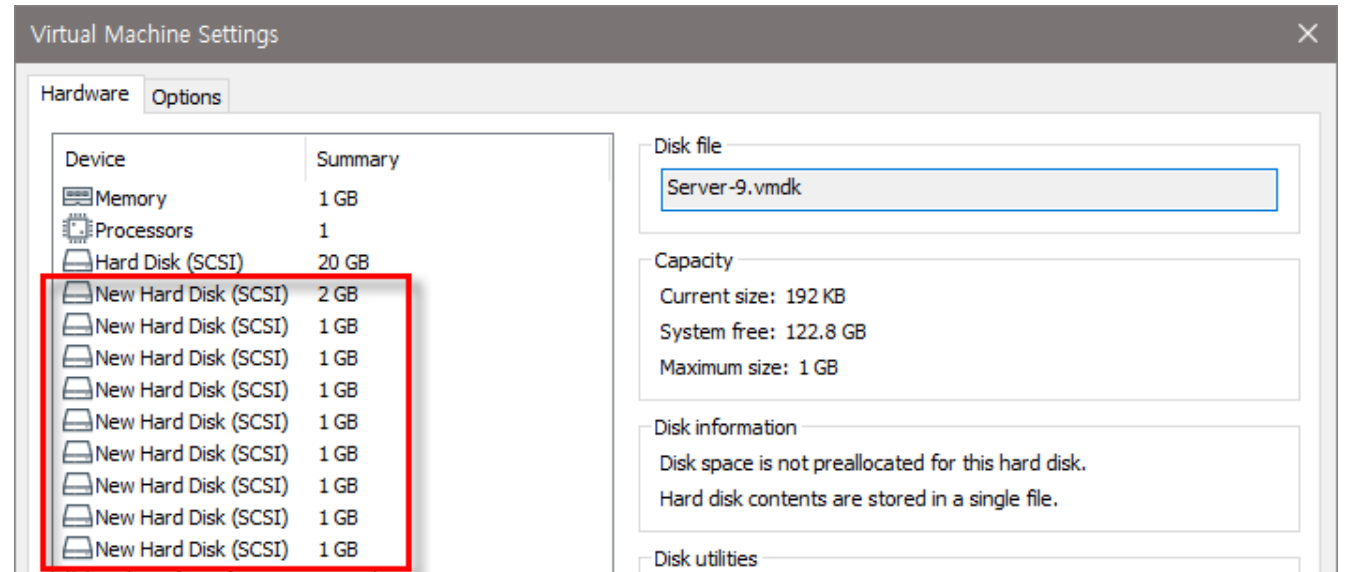
- 하드웨어 구성



RAID 실습 환경 구성

- VMWare에서 디스크 추가

장치 이름	디스크 크기	파일 이름	비고
SCSI 0:1	2GB	자동	Linear RAID
SCSI 0:2	1GB	자동	
SCSI 0:3	1GB	자동	RAID 0
SCSI 0:4	1GB	자동	
SCSI 0:5	1GB	자동	RAID 1
SCSI 0:6	1GB	자동	
SCSI 0:7	사용할 수 없음(VMware에서 예약되어 있음)		
SCSI 0:8	1GB	자동	RAID 5
SCSI 0:9	1GB	자동	
SCSI 0:10	1GB	자동	



RAID 실습 환경 구성

- 추가된 디스크 확인

- `# ls -l /dev/sd*`



```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# ls -l /dev/sd*
brw-rw---- 1 root disk 8,  0  7월 21 10:59 /dev/sda
brw-rw---- 1 root disk 8,  1  7월 21 10:59 /dev/sda1
brw-rw---- 1 root disk 8, 16  7월 21 10:59 /dev/sdb
brw-rw---- 1 root disk 8, 32  7월 21 10:59 /dev/sdc
brw-rw---- 1 root disk 8, 48  7월 21 10:59 /dev/sdd
brw-rw---- 1 root disk 8, 64  7월 21 10:59 /dev/sde
brw-rw---- 1 root disk 8, 80  7월 21 10:59 /dev/sdf
brw-rw---- 1 root disk 8, 96  7월 21 10:59 /dev/sdg
brw-rw---- 1 root disk 8,112  7월 21 10:59 /dev/sdh
brw-rw---- 1 root disk 8,128  7월 21 10:59 /dev/sdi
brw-rw---- 1 root disk 8,144  7월 21 10:59 /dev/sdj
root@server:~#
```

RAID 실습 환경 구성

- RAID용 파티션 생성
 - /dev/sdb 장치에 /dev/sdb1 파티션 생성
 - # fdisk /dev/sdb
 - 별도의 파일 시스템 지정
 - fd (Linux raid autodetect)

```
# fdisk /dev/sdb    -- [SCSI 0:1] 디스크 선택
Command: n         -- 새로운 파티션 분할
Select: p          -- Primary 파티션 선택
Partition number: 1 -- 파티션 1번 선택
First sector:  -- 시작 섹터 번호
Last sector:  -- 마지막 섹터 번호
Command: t         -- 파일 시스템의 유형 선택
Hex code: fd       -- 'Linux raid autodetect' 유형 번호 선택(L을 입력하면 전체 유형이 출력됨)
Command: p         -- 설정 내용 확인
Command: w         -- 설정 내용 저장
```

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# fdisk /dev/sdb
fdisk (util-linux 2.31.1)
Created a new disklabel with identifier 0x1234567890

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-4194303, default 2048):
Last sector, +sectors or +size[K,M,G,T,P] (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): fd
Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): p
Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Used blocks: 0 of 1 * 512 bytes
Disk identifier: 0x1234567890

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1           2048 4194303 4192256    2G fd Linux raid autodetect

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

RAID 실습 환경 구성

- RAID용 파티션 생성 (반복)

- /dev/sdc ~ /dev/sdj 장치에 /dev/sdb1 ~ /dev/sdj1 파티션 생성

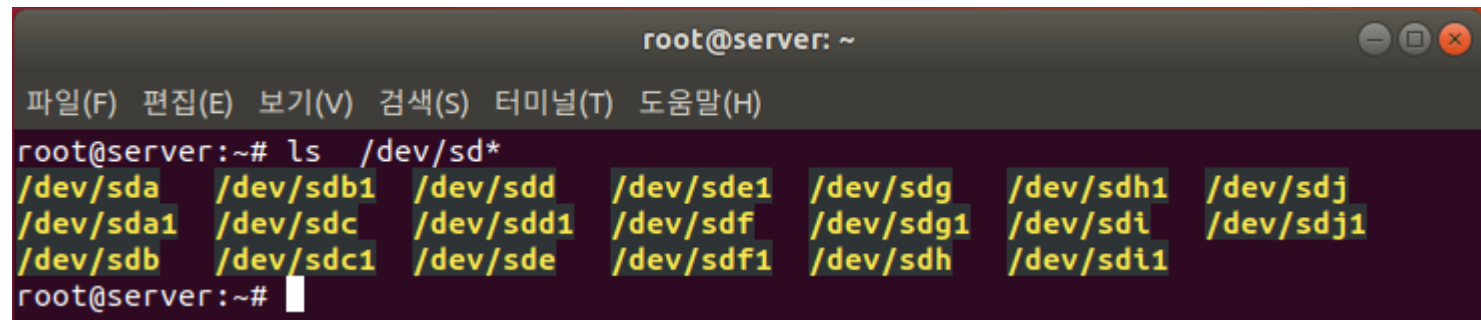
- `# fdisk /dev/sdb ~ # fdisk /dev/sdj`

- 별도의 파일 시스템 지정

- `fd` (Linux raid autodetect)

- 파티션 확인

- `# ls /dev/sd*`



```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
root@server:~# ls /dev/sd*  
/dev/sda /dev/sdb1 /dev/sdd /dev/sde1 /dev/sdg /dev/sdh1 /dev/sdj  
/dev/sda1 /dev/sdc /dev/sdd1 /dev/sdf /dev/sdg1 /dev/sdi /dev/sdj1  
/dev/sdb /dev/sdc1 /dev/sde /dev/sdf1 /dev/sdh /dev/sdi1  
root@server:~#
```

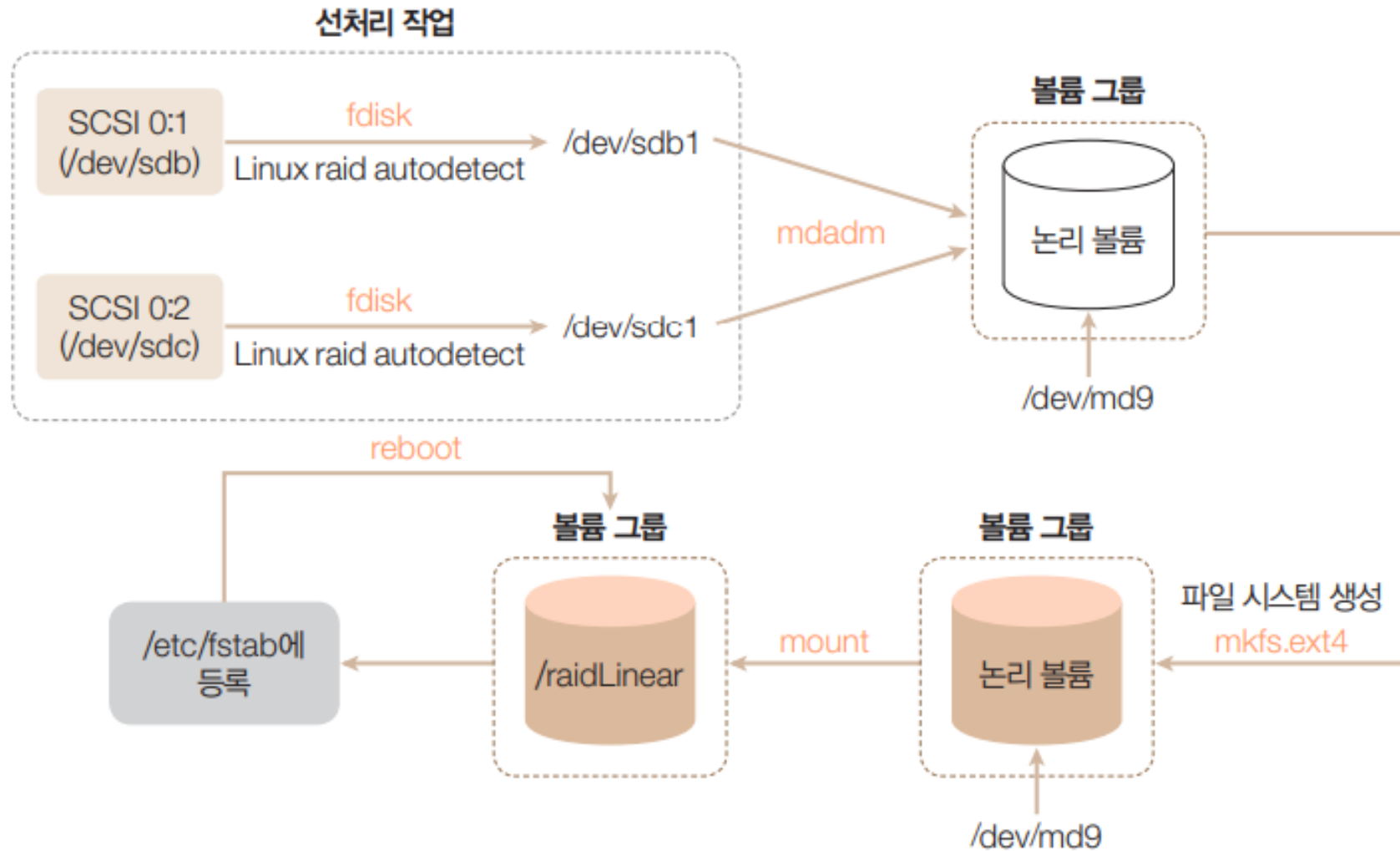
- RAID 관련 패키지 설치

- `# apt-get -y install mdadm`

RAID 실습 환경 구성

- VMWare 가상 환경 백업 (꼭!)

Linear RAID 구축



Linear RAID 구축

- 선처리 작업
 - /dev/sdb, /dev/sdc 파티션 상태 확인
 - `# fdisk -l /dev/sdb ; fdisk -l /dev/sdc`

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# fdisk -l /dev/sdb ; fdisk -l /dev/sdc
Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Disk identifier: (logical) 09345d3

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1   2048 4194303 4192256   2G fd Linux raid autodetect

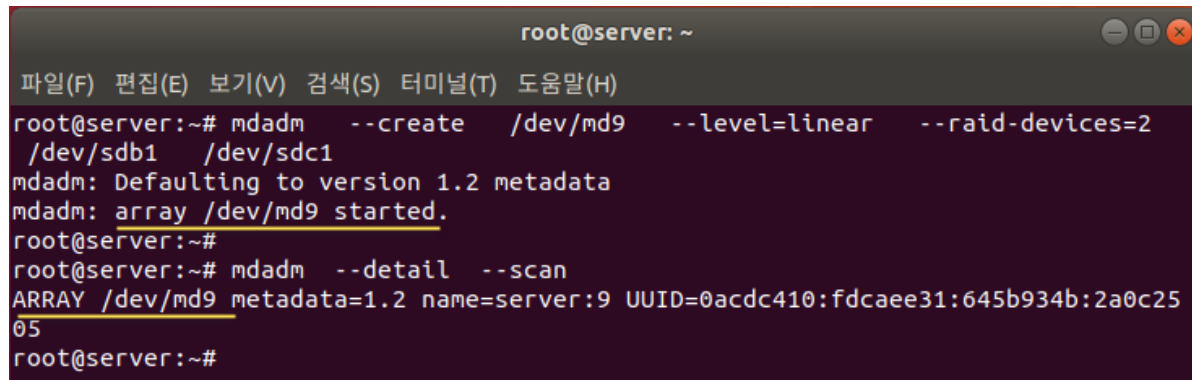
Disk /dev/sdc: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Disk identifier: (logical) 2990364

Device      Boot Start      End Sectors  Size Id Type
/dev/sdc1   2048 2097151 2095104 1023M fd Linux raid autodetect
root@server:~#
```

Linear RAID 구축

- RAID 구축 (mdadm 활용)
 - /dev/sdb1과 /dev/sdc1을 Linear RAID 장치인 /dev/md9로 생성

```
mdadm --create /dev/md9 --level=linear --raid-devices=2 /dev/sdb1 /dev/sdc1 -- RAID 생성
mdadm --detail --scan -- RAID 확인
```



```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# mdadm --create /dev/md9 --level=linear --raid-devices=2
/dev/sdb1 /dev/sdc1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md9 started.
root@server:~#
root@server:~# mdadm --detail --scan
ARRAY /dev/md9 metadata=1.2 name=server:9 UUID=0acdc410:fdcaee31:645b934b:2a0c25
05
root@server:~#
```

- /dev/md9 파티션 장치의 파일 시스템 생성
 - # mkfs.ext4 /dev/md9

Linear RAID 구축

- 마운트
 - 마운트할 디렉토리 생성 (디렉토리명: /raidLinear)

- # mkdir /raidLinear

- 마운트

- # mount /dev/md9 /raidLinear

- 확인

- # df -h

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# mount /dev/md9 /raidLinear
root@server:~# df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            460944         0    460944   0% /dev
tmpfs           98492         1484    97008   2% /run
/dev/sda1       20509264    6162288 13282120 32% /
/dev/loop1      492624        90624    492448 100% /snap/shm/270
/dev/loop15     35456         35456         0 100% /snap/gtk-common-themes/818
tmpfs           98488          24    98464   1% /run/user/0
/dev/md9        3026704        9216 2844024   1% /raidLinear
root@server:~#
```

- 자동 마운트 등록

```
/dev/md9 /raidLinear ext4 defaults 0 0
```

```
*fstab
/etc

# /etc/fstab: static file system information.
#
# 'blkid' to print the universally unique identifier for a
# <file system>, and use it with this syntax:
# / was on /dev/sda1 during installation
UUID=ab20eecb-b051-4aef-9dd5-ca6ea5393b39 / ext4 errors=remount-ro 0 1
/swapfile none swap sw 0 0

/dev/md9 /raidLinear ext4 defaults 0 0

일반 텍스트 탭 너비: 8 12행, 1열 삽입
```


Linear RAID 구축

- 구축한 Linear RAID 확인
 - # mdadm --detail /dev/md9

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# mdadm --detail /dev/md9
/dev/md9:
  Version : 1.2
  Creation Time : Sun Jul 21 11:49:09 2019
  Raid Level : linear
  Array Size : 3141632 (3.00 GiB 3.22 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

               ver:9 (100% good) server)
  UUID : 0acdc410:fdcae31f:045b934b:2a0c2505
  Events : 0

Number Major Minor RaidDevice State
  0       8      17         0    active sync  /dev/sdb1
  1       8      33         1    active sync  /dev/sdc1
root@server:~#
```

RAID 0 구축

- RAID 0 장치 생성 (/dev/md0)

- `# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdd1 /dev/sde1`

- 확인 → `# mdadm --detail --scan`

- 파티션 장치 포맷

- `# mkfs.ext4 /dev/md0`

- 마운트할 디렉토리 생성 및 마운트

- `# mkdir /raid0`

- `# mount /dev/md0 /raid0`

- 구축한 RAID 0 확인

- `# mdadm --detail /dev/md0`

End of slide
