

11장 하드디스크 관리

강사 김영석

A top-down view of a wooden desk. On the desk, there is a silver laptop with a black keyboard, a pair of black-rimmed glasses, a white coffee cup with a yellow handle, and a small green succulent in a dark pot. The wood grain of the desk is clearly visible.

CONTENT

1 리눅스 파일 시스템 종류

2 디스크 추가 설치

3 디스크 관리

1. 리눅스 파일 시스템 종류

❖ 리눅스 고유의 디스크 기반 파일 시스템

■ ext(ext1)

- Extended File System 의 약자로 extfs 라고도 하며 기존에 파일 시스템에 비해 확장된 파일 시스템이다.
- 하지만 indode 의 수정과 데이터 수정 시간이 지원 안되는 문제와 링크드 리스트로 파일 시스템을 구성하여 파일 시스템이 복잡해지고 파편화되는 문제가 있었다. 그래서 지금은 사용하지 않는다.

■ ext2

- 공식 명칭은 Second Extended File System 으로 ext 파일 시스템의 문제를 해결하기 위해서 나왔다.
- ext2 파일 시스템은 2TB로 제한 되어 있었으나 현재는 이론적으로 32TB 까지 지원이 가능하다.
- 지금도 부팅 가능한 USB 플래시 드라이브와 기타 SSD 장치에 사용되고 있다.



■ ext3

- ext3는 ext2 를 기반으로 개발되어 호환이 가능하다. 즉 ext2 파일 시스템의 파일을 별도의 변경 없이 ext3 파일 시스템에서도 사용이 가능하다.
- ext3의 장점은 저널링 기능을 도입했다는 것이다. 저널링은 디스크에 기록 되는 데이터의 복구 기능을 강화한것이다. 저널링은 데이터를 디스크에 기록하기 전에 먼저 저널에 수정 사항을 기록한다. 저널은 쉽게 말해 로그 기록 같은 것이라고 생각하면 된다. 그래서 시스템에 문제가 발생했을 때 이 저널의 기록을 보고 빠르게 복구가 가능하다.

■ ext4

- ext4 파일 시스템은 1EB(엑사 바이트)이상의 볼륨과 16TB 이상의 파일을 지원하며, ext2와 ext3과 호환성을 유지 한다.
- ext3 에서는 서브 디렉터리의 수가 32,000개로 제한 되었으나 ext4 에서는 64,000개로 늘어 났으며, 온라인 조각 모음 기능도 지원한다.



■ XFS

- XFS 파일 시스템은 eXtended File System으로 고성능 저널링 파일 시스템이다.
- 현재 대부분의 리눅스 배포판에서 사용하고 있으며 최대 16EB까지 지원한다.
- 우분투에서 XFS 파일 시스템을 이용하려면 xfsprogs 패키지가 설치 되어 있어야 한다.

❖ 리눅스에서 지원하는 다양한 파일 시스템

파일 시스템	기능
msdos	MS-DOS 파티션을 사용하기 위한 파일 시스템
iso9660	CD-ROM, DVD의 표준 파일 시스템으로 읽기 전용으로 사용
nfs	Network File System으로 원격 서버의 디스크를 연결할 때 사용
ufs	Unix File System으로 유닉스의 표준 파일 시스템
vfat	윈도우 95, 98, NT를 지원하기 위한 파일 시스템
hpfs	HPFS를 지원하기 위한 파일 시스템
ntfs	윈도우의 NTFS 를 지원하기 위한 파일 시스템
sysv	유닉스 시스템V 를 지원하기 위한 파일 시스템
hfs	맥 컴퓨터의 hfs 파일 시스템을 지원하기 위한 파일 시스템


❖ 특수 용도의 가상 파일 시스템

파일 시스템	기능
swap	스왑 영역을 관리하기 위한 스왑 파일 시스템
tmpfs	Temporary File System 으로 메모리에서 임시 파일을 저장하기 위한 파일 시스템 시스템이 재시작할 때 마다 기존 내용은 제거함 /tmp 디렉터리가 예
proc	proc 파일 시스템으로 /proc 디렉터리 커널의 현재 상태를 나타내는 파일을 가지고 있음
ramfs	램 디스크를 지원하는 파일 시스템
rootfs	Root File System으로 / 디렉터리 시스템 초기화 및 관리에 필요한 내용을 관리 함

❖ 현재 시스템이 지원하는 파일 시스템 확인

```
user1@Server1:~$ cat /proc/filesystems
```

```
nodev    sysfs
nodev    tmpfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cgroup2
nodev    cpuset
nodev    devtmpfs
nodev    configfs
nodev    devpts
          ext3
          ext2
          ext4
          squashfs
          vfat
nodev    ecryptfs
          fuseblk
nodev    fuse
nodev    fusectl
```


- 
- 1열의 nodev 는 해당 파일 시스템이 블록 장치에 연결 되어 있지 않다는 것으로 가상 파일 시스템임을 나타낸다.
 - 2열에 나오는 것이 지원하는 파일 시스템이다.
 - 특수 용도인 가상 파일 시스템이 대부분이며, 이 중 fuseblk는 ntfs 파일 시스템을 연결 할 때 사용되는 파일 시스템이다.

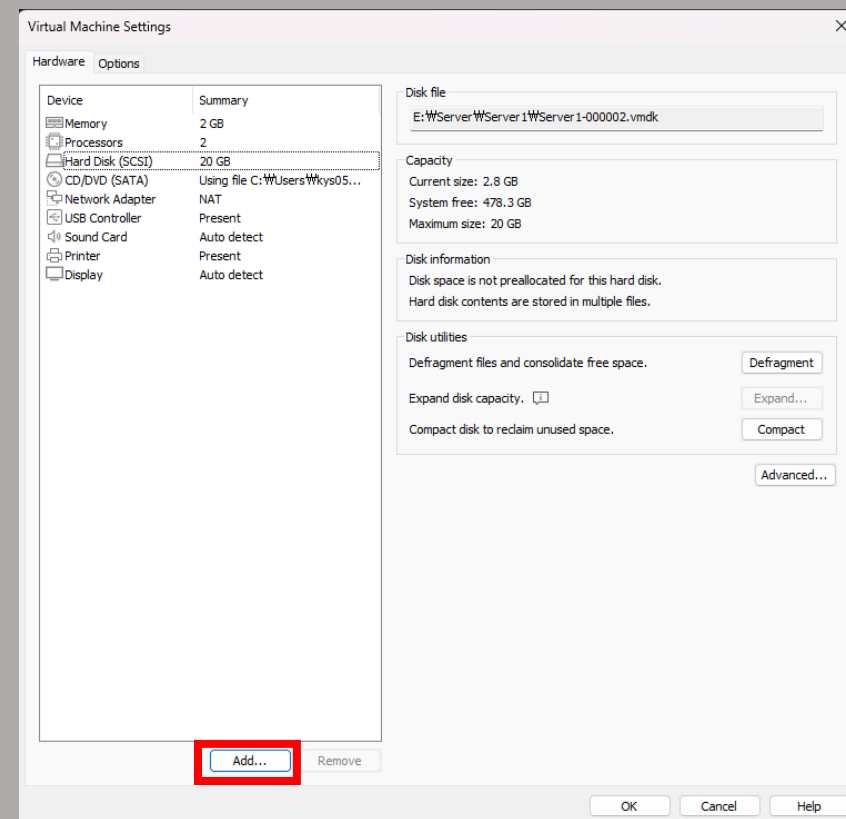
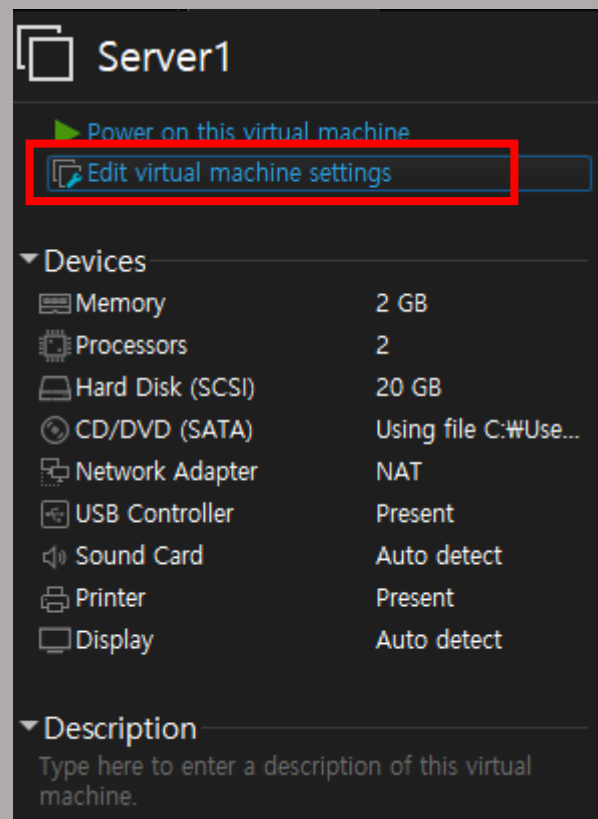
2. 디스크 추가 설치

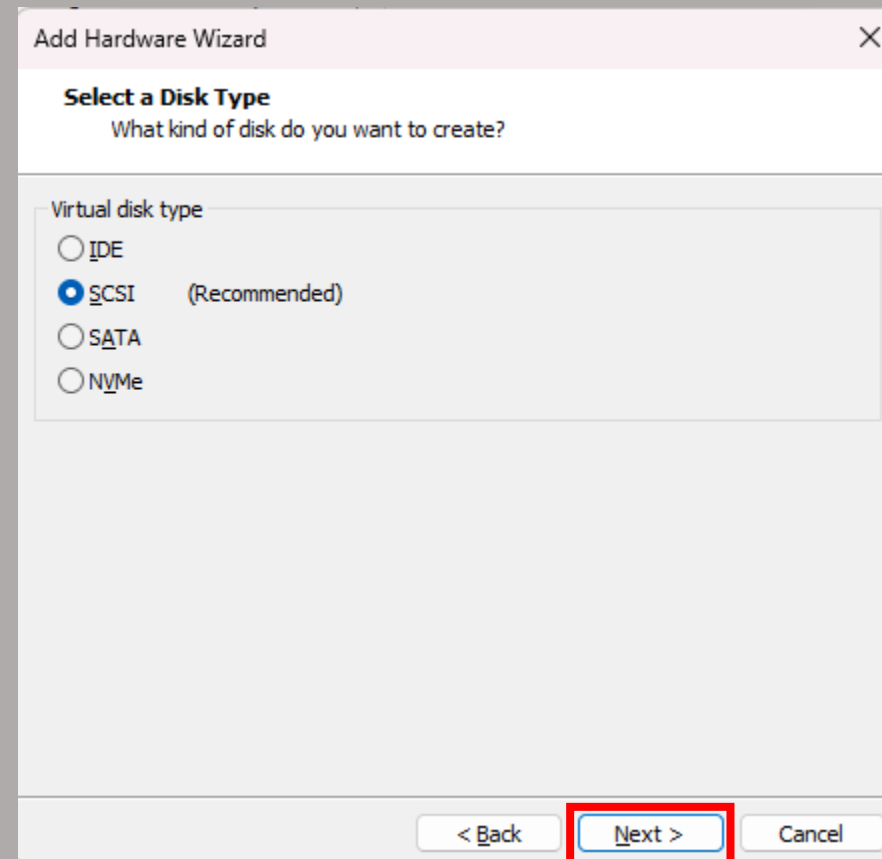
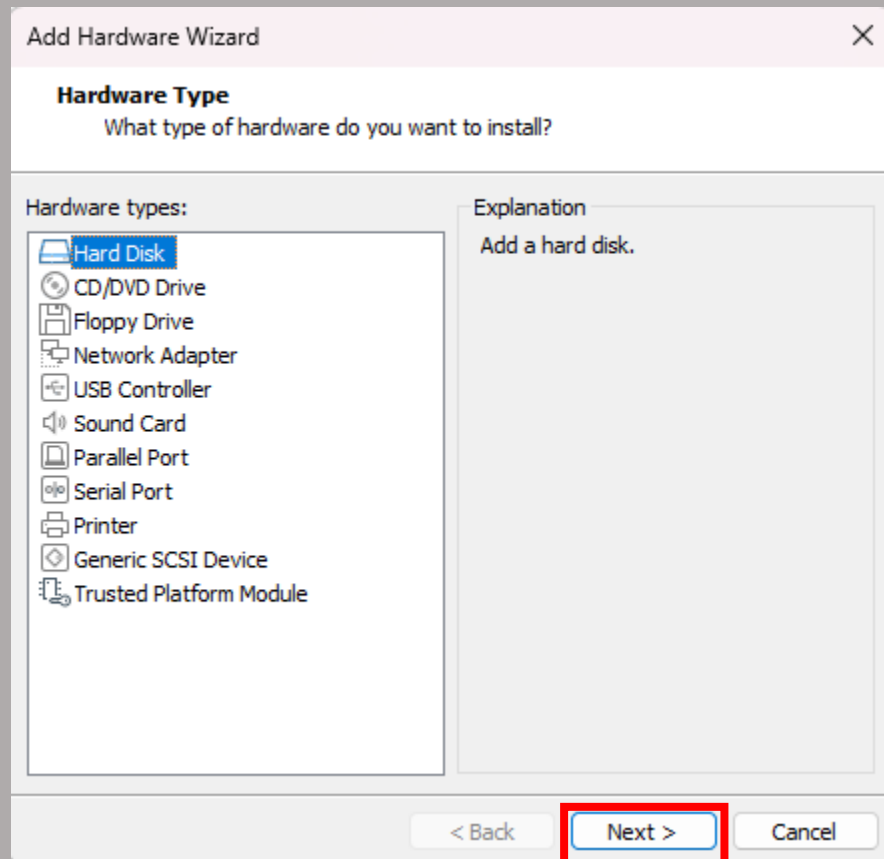
❖리눅스 디스크 보기

```
user1@Server1:~$ ls -l /dev/sd*  
brw-rw---- 1 root disk 8, 0  8월 20 14:47 /dev/sda  
brw-rw---- 1 root disk 8, 1  8월 20 14:47 /dev/sda1  
brw-rw---- 1 root disk 8, 2  8월 20 14:47 /dev/sda2  
brw-rw---- 1 root disk 8, 3  8월 20 14:47 /dev/sda3
```

- /dev/sd* 로 시작하는 것이 물리적인 하드 디스크다.
- /dev/sd*n 은 /dev/sd* 에서 논리적으로 분리된 파티션의 숫자다.
- 실제로 사용하는 것은 /dev/sd*n 으로 마운트 포인트가 된다.

❖ 하드 디스크 추가하기





Add Hardware Wizard

Select a Disk
Which disk do you want to use?

Disk

☒ Create a new virtual disk

A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.

☐ Use an existing virtual disk

Choose this option to reuse a previously configured disk.

☐ Use a physical disk (for advanced users)

Choose this option to give the virtual machine direct access to a local hard disk. Requires administrator privileges.

< Back

Next >

Cancel

Add Hardware Wizard

Specify Disk Capacity
How large do you want this disk to be?

Maximum disk size (GB): 20.0

Recommended size for Ubuntu: 20 GB

☐ Allocate all disk space now.

Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.

☐ Store virtual disk as a single file

☒ Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

< Back


Next >

Cancel

Add Hardware Wizard ✕

Specify Disk File
Where would you like to store the disk file?

☐ Disk file
A 20 GB virtual disk be created using multiple disk files. The disk files will be automatically named based on this file name.



```
user1@Server1:~/바탕화면$ ls -l /dev/sd*
brw-rw---- 1 root disk 8,  0  8월 23 19:01 /dev/sda
brw-rw---- 1 root disk 8,  1  8월 23 19:01 /dev/sda1
brw-rw---- 1 root disk 8,  2  8월 23 19:01 /dev/sda2
brw-rw---- 1 root disk 8,  3  8월 23 19:01 /dev/sda3
brw-rw---- 1 root disk 8, 16  8월 23 19:01 /dev/sdb
```

- /dev/sdb 로 물리적 하드 디스크가 추가 된 것을 확인 할 수 있다.
- 물리적 하드 디스크는 영문자가 증가되고 논리적인 파티션은 숫자가 증가 된다.


Quiz

❖ Server2 에 하드 디스크 하나 추가해 보세요

❖디스크 파티션 나누기

fdisk	
기능	디스크의 파티션 생성, 삭제, 보기 등 파티션을 관리 한다.
형식	fdisk [옵션] [장치명]
옵션	-b [크기]: 섹터 크기를 지정한다. (512, 1024, 2048, 4096) -l: 파티션 테이블을 출력한다.
사용 예	fdisk /dev/sdb fdisk -l

```
user1@Server1:~/바탕화면$ sudo fdisk -l
[sudo] user1 암호:
Disk /dev/loop0: 63.45 MiB, 66531328 bytes, 129944 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
```



```
user1@Server1:~/바탕화면$ sudo fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.37.2).
```

```
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 0x556837e3.
```

```
Command (m for help): m
```

```
Help:
```

DOS (MBR)

- a toggle a bootable flag
- b edit nested BSD disklabel
- c toggle the dos compatibility flag

Generic

- d delete a partition
- F list free unpartitioned space
- l list known partition types
- n add a new partition

- n add a new partition
- p print the partition table
- t change a partition type
- v verify the partition table
- i print information about a partition

Misc

- m print this menu
- u change display/entry units
- x extra functionality (experts only)

Script

- I load disk layout from sfdisk script file
- O dump disk layout to sfdisk script file

Save & Exit

- w write table to disk and exit
- q quit without saving changes

Create a new label

- g create a new empty GPT partition table
- G create a new empty SGI (IRIX) partition table
- o create a new empty DOS partition table
- s create a new empty Sun partition table

■ fdisk 내부 명령

명령	기능	명령	기능
a	부팅 파티션을 설정한다.	p	파티션 테이블을 출력한다.
b	BSD 디스크 라벨을 편집한다.	q	작업 내용을 저장하지 않고 종료한다.
c	도스 호환성을 설정한다.	s	새로운 빈 Sun 디스크 라벨을 생성한다.
d	파티션을 삭제한다.	t	파티션의 시스템 ID를 변경한다. (파일 시스템 종류 변경)
l	사용 가능한 파티션의 종류를 출력한다.	u	항목 정보를 변경 및 출력한다.
m	도움말을 출력한다.	v	파티션 테이블을 검사한다.
n	새로운 파티션을 추가한다.	w	파티션 정보를 디스크에 저장하고 종료한다.
o	새로운 빈 DOS 파티션을 생성한다.	x	실린더 개수 변경 등 전문가를 위한 부가적 기능이다.



```
user1@Server1:~/바탕화면$ sudo fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.37.2).
```

```
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 0x0ca9e544.
```

```
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):
```

```
First sector (2048-41943039, default 2048):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943039, default 41943039): +500M
```

```
Created a new partition 1 of type 'Linux' and of size 500 MiB.
```




```
Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (1026048-41943039, default 1026048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1026048-41943039, default 41943039):

Created a new partition 2 of type 'Linux' and of size 19.5 GiB.

Command (m for help): p
Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VMware Virtual S
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: 0x0ca9e544

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1                2048  1026047  1024000  500M 83 Linux
/dev/sdb2          1026048 41943039 40916992 19.5G 83 Linux
```



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

Quiz

- ❖ Server2 에 디스크 파티션을 아래와 같이 나누어 보세요
 - 3개의 디스크 파티션을 나눈다.
 - 첫번째 파티션은 500 메가
 - 두번째 파티션은 200 메가
 - 세번째 파티션은 나머지 용량

❖ 파일 시스템 생성 명령

mkfs	
기능	리눅스 파일 시스템을 만든다.
형식	mkfs [옵션][장치명]
옵션	-t [파일 시스템 종류]: 파일 시스템의 종류를 지정한다.

mke2fs	
기능	리눅스 개정판 확장 파일 시스템을 만든다. (ext2, ext3, ext4)
형식	mke2fs [옵션][장치명]
옵션	-t [파일 시스템 종류]: 파일 시스템의 종류를 지정한다. -b [블록크기]: 블록 크기를 바이트 수로 지정한다. -c: 대드 블록을 체크한다. -f [프래그먼트 크기]: 프래그먼트 크기를 바이트 수로 지정한다. -i [inode 바이트 수]: inode 당 바이트 수를 지정한다. -m [예약 블록 퍼센트]: 슈퍼유저에게 예약해 둘 블록의 퍼센트를 지정한다.



```
user1@Server1:~/바탕화면$ cat /etc/mke2fs.conf
```

```
[defaults]
```

```
base_features = sparse_super,large_file,filetype,resize_inode,dir_index,ext_attr
default_mntopts = acl,user_xattr
enable_periodic_fsck = 0
blocksize = 4096
inode_size = 256
inode_ratio = 16384
```

```
[fs_types]
```

```
ext3 = {
    features = has_journal
}
ext4 = {
    features = has_journal,extent,huge_file,flex_bg,metadata_csum,64bit,dir_nlink,extra_isize
}
small = {
    inode_ratio = 4096
}
floppy = {
    inode_ratio = 8192
}
big = {
    inode_ratio = 32768
}
huge = {
    inode_ratio = 65536
}
```


user1@Server1:~/바탕화면\$ ls -l /sbin/mk*

```
lrwxrwxrwx 1 root root      8 8월 17 16:28 /sbin/mkdosfs -> mkfs.fat
-rwxr-xr-x 1 root root 133752 6월 2 2022 /sbin/mke2fs
-rwxr-xr-x 1 root root 14712 2월 21 2022 /sbin/mkfs
-rwxr-xr-x 1 root root 22904 2월 21 2022 /sbin/mkfs.bfs
-rwxr-xr-x 1 root root 35128 2월 21 2022 /sbin/mkfs.cramfs
lrwxrwxrwx 1 root root      6 8월 17 16:28 /sbin/mkfs.ext2 -> mke2fs
lrwxrwxrwx 1 root root      6 8월 17 16:28 /sbin/mkfs.ext3 -> mke2fs
lrwxrwxrwx 1 root root      6 8월 17 16:28 /sbin/mkfs.ext4 -> mke2fs
-rwxr-xr-x 1 root root 52048 3월 23 2022 /sbin/mkfs.fat
-rwxr-xr-x 1 root root 43400 2월 21 2022 /sbin/mkfs.minix
lrwxrwxrwx 1 root root      8 8월 17 16:28 /sbin/mkfs.msdos -> mkfs.fat
lrwxrwxrwx 1 root root      6 8월 17 16:28 /sbin/mkfs.ntfs -> mkntfs
lrwxrwxrwx 1 root root      8 8월 17 16:28 /sbin/mkfs.vfat -> mkfs.fat
-rwxr-xr-x 1 root root 22704 2월 2 2023 /sbin/mkhomedir_helper
-rwxr-xr-x 1 root root 12453 6월 14 15:54 /sbin/mkinitramfs
-rwxr-xr-x 1 root root 14648 6월 2 2022 /sbin/mklost+found
-rwxr-xr-x 1 root root 72072 11월 1 2022 /sbin/mkntfs
-rwxr-xr-x 1 root root 47488 2월 21 2022 /sbin/mkswap
```

❖ 파일 시스템 생성하기


```
user1@Server1:~/바탕화면$ sudo mkfs /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 128000 4k blocks and 128000 inodes
Filesystem UUID: 0ae49968-899c-4551-857f-58ddaecc6812
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```




```
user1@Server1:~/바탕화면$ sudo mkfs.ext3 -v /dev/sdb2
mke2fs 1.46.5 (30-Dec-2021)
fs_types for mke2fs.conf resolution: 'ext3'
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1281120 inodes, 5114624 blocks
255731 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
157 block groups
32768 blocks per group, 32768 fragments per group
8160 inodes per group
Filesystem UUID: 5c94c146-f9d9-4a37-8985-1a8d4a7bc9be
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
user1@Server1:~/바탕화면$ sudo mke2fs -t ext3 /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
/dev/sdb1 contains a ext2 file system
        created on Wed Aug 23 19:43:15 2023
Proceed anyway? (y,N) y
Creating filesystem with 128000 4k blocks and 128000 inodes
Filesystem UUID: a0db16a0-e76d-4a72-b925-2ef49c2a6c25
Superblock backups stored on blocks:
        32768, 98304

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



```
user1@Server1:~/바탕화면$ sudo mke2fs -t ext4 -b 4096 /dev/sdb2
mke2fs 1.46.5 (30-Dec-2021)
/dev/sdb2 contains a ext3 file system
        created on Wed Aug 23 19:43:59 2023
Proceed anyway? (y,N) y
Creating filesystem with 5114624 4k blocks and 1281120 inodes
Filesystem UUID: 02944434-c26b-4580-be7e-f202618f7f15
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```



❖ Server2 에서 아래와 같이 파일 시스템을 구현해 본다.

- 첫번째 파티션은 mkfs 의 기본 파일 시스템으로 구현
- 두번째 파티션은 mkfs 로 ext3 으로 파일 시스템 구현
- 세번째 파티션은 mke2fs 로 ext2 로 파일 시스템 구현

❖디스크 마운트 하기

```
user1@Server1:~/바탕화면$ sudo mount /dev/sdb1 /sdb_1
user1@Server1:~/바탕화면$ sudo mount /dev/sdb2 /sdb_2
user1@Server1:~/바탕화면$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=909952k,nr_inodes=227488,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=190320k,mode=755,inode64)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sr0 on /media/user1/Ubuntu 22.04.3 LTS amd64 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,icharset=utf8,uhelper=udisks2)
/dev/sdb1 on /sdb_1 type ext3 (rw,relatime)
/dev/sdb2 on /sdb_2 type ext4 (rw,relatime)
```

❖파일 시스템 사용하기

```
user1@Server1:~/바탕화면$ sudo cp /etc/hosts /sdb_1
user1@Server1:~/바탕화면$ ls /sdb_1
hosts  lost+found
```


❖ 언마운트 하기

```
user1@Server1:~/바탕화면$ sudo umount /dev/sdb1
user1@Server1:~/바탕화면$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=909952k,nr_inodes=227488,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=190320k,mode=755,inode64)
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro)
/var/lib/snapd/snaps/gnome-3-38-2004_143.snap on /snap/gnome-3-38-2004/143 type squashfs (ro,nodev,relatime,errors=continue,threads=single,x-gdu.hide)
/dev/sda1 on /boot type ext4 (rw,relatime)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/snapd/ns type tmpfs (rw,nosuid,nodev,noexec,relatime,size=190320k,mode=755,inode64)
nsfs on /run/snapd/ns/snapd-desktop-integration.mnt type nsfs (rw)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=190316k,nr_inodes=47579,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sr0 on /media/user1/Ubuntu 22.04.3 LTS amd64 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,iocharset=utf8,uhelper=udisks2)
/dev/sdb2 on /sdb_2 type ext4 (rw,relatime)
```


Quiz

- ❖ Server2 에서 아래와 같이 구현해 본다.
 - 디렉터리 newdsk1, newdsk2, newdsk3 을 생성한다.
 - 각 파티션을 마운트 해본다.
 - 마운트 되었는지 확인한다.
 - 마운트를 해제 해 본다.

3. 디스크 관리

❖ 디스크 사용량 확인하기

- 파일 시스템별 디스크 사용량 확인 하기

df	
기능	디스크의 남은 공간에 대한 정보를 출력한다.
형식	df [옵션] [파일 시스템]
옵션	-a: 모든 파일 시스템을 대상으로 사용량을 확인한다. -k: 디스크 사용량을 KB 단위로 출력한다. -m: 디스크 사용량을 MB 단위로 출력한다. -h: 디스크 사용량을 알기 쉬운 단위(GB, MB, KB 등)로 출력한다. -t[파일 시스템 종류]: 지정한 파일 시스템 종류에 해당하는 디스크의 사용량을 출력한다. -T: 파일 시스템 종류를 출력한다.

```
user1@Server1:~/바탕화면$ df
```

파일 시스템	1K-블록	사용	가용	사용%	마운트위치
tmpfs	190320	1704	188616	1%	/run
/dev/sda3	16574272	9412056	6294940	60%	/
tmpfs	951584	0	951584	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
/dev/sda1	1882556	104472	1664116	6%	/boot
tmpfs	190316	96	190220	1%	/run/user/1000
/dev/sr0	4919592	4919592	0	100%	/media/user1/Ubuntu 22.04.3 LTS amd64
/dev/sdb2	19960656	24	18921324	1%	/sdb_2

```
user1@Server1:~/바탕화면$ df -h
```


파일 시스템	크기	사용	가용	사용%	마운트위치
tmpfs	186M	1.7M	185M	1%	/run
/dev/sda3	16G	9.0G	6.1G	60%	/
tmpfs	930M	0	930M	0%	/dev/shm
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
/dev/sda1	1.8G	103M	1.6G	6%	/boot
tmpfs	186M	96K	186M	1%	/run/user/1000
/dev/sr0	4.7G	4.7G	0	100%	/media/user1/Ubuntu 22.04.3 LTS amd64
/dev/sdb2	20G	24K	19G	1%	/sdb_2

user1@Server1:~/바탕화면\$ df -Th

파일 시스템	형식	크기	사용	가용	사용%	마운트위치
tmpfs	tmpfs	186M	1.7M	185M	1%	/run
/dev/sda3	ext4	16G	9.0G	6.1G	60%	/
tmpfs	tmpfs	930M	0	930M	0%	/dev/shm
tmpfs	tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
/dev/sda1	ext4	1.8G	103M	1.6G	6%	/boot
tmpfs	tmpfs	186M	96K	186M	1%	/run/user/1000
/dev/sr0	iso9660	4.7G	4.7G	0	100%	/media/user1/Ubuntu 22.04.3 LTS amd64
/dev/sdb2	ext4	20G	24K	19G	1%	/sdb_2

❖ 파일 시스템 검사 하기


fsck	
기능	리눅스의 파일 시스템을 점검한다.
형식	fsck [옵션] [장치명]
옵션	-f: 강제로 점검한다. -b [슈퍼블록]: 슈퍼블록으로 지정한 백업 슈퍼블록을 사용한다. -y: 모든 질문에 yes로 대답한다. -a: 파일 시스템 검사에서 문제를 발견했을 때 자동으로 복구한다.
사용 예	fsck /dev/sdb1 fsck -f /dev/sdb1



```
user1@Server1:~/바탕화면$ sudo fsck /dev/sdb1
fsck from util-linux 2.37.2
e2fsck 1.46.5 (30-Dec-2021)
/dev/sdb1: clean, 12/128000 files, 12215/128000 blocks
user1@Server1:~/바탕화면$ sudo fsck -f /dev/sdb1
fsck from util-linux 2.37.2
e2fsck 1.46.5 (30-Dec-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb1: 12/128000 files (0.0% non-contiguous), 12215/128000 blocks
user1@Server1:~/바탕화면$ sudo fsck.ext3 /dev/sdb1
e2fsck 1.46.5 (30-Dec-2021)
/dev/sdb1: clean, 12/128000 files, 12215/128000 blocks
```



e2fsck	
기능	리눅스의 파일 시스템을 점검한다.
형식	e2fsck [옵션] [장치명]
옵션	-f: 강제로 점검한다. -b [슈퍼블록]: 슈퍼블록으로 지정한 백업 슈퍼블록을 사용한다. -y: 모든 질문에 yes로 대답한다. -j ext3/ext4: ext3 이나 ext4 파일 시스템을 검사할 때 지정한다.
사용 예	e2fsck /dev/sdb1 e2fsck -f /dev/sdb1



```
user1@Server1:~/바탕화면$ sudo e2fsck /dev/sdb1
e2fsck 1.46.5 (30-Dec-2021)
/dev/sdb1: clean, 12/128000 files, 12215/128000 blocks
user1@Server1:~/바탕화면$ sudo e2fsck -f /dev/sdb1
e2fsck 1.46.5 (30-Dec-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb1: 12/128000 files (0.0% non-contiguous), 12215/128000 blocks
```

❖ 배드 블록 검사하기

badblocks	
기능	장치의 배드 블록을 검색한다.
형식	badblocks [옵션] [장치명]
옵션	-v : 검색 결과를 자세하게 출력한다. -o 출력 파일 : 검색한 배드 블록 목록을 지정한 출력 파일에 저장한다.
사용 예	badblocks -v /dev/sdb1 badblocks -v -o bad.out /dev/sdb1

```
user1@Server1:~/바탕화면$ sudo badblocks -v /dev/sdb1
Checking blocks 0 to 511999
Checking for bad blocks (read-only test): done
Pass completed, 0 bad blocks found. (0/0/0 errors)
user1@Server1:~/바탕화면$ sudo badblocks -v -o bad.out /dev/sdb1
Checking blocks 0 to 511999
Checking for bad blocks (read-only test): done
Pass completed, 0 bad blocks found. (0/0/0 errors)
user1@Server1:~/바탕화면$ ls
bad.out
```


❖ 백업 슈퍼블록을 이용하여 파일 시스템 복구하기

dumpe2fs	
기능	파일 시스템의 정보를 출력한다.
형식	dumpe2fs [장치명]
사용 예	dumpe2fs /dev/sdb1

```
user1@Server1:~/바탕화면$ sudo dumpe2fs /dev/sdb1
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem volume name:   <none>
Last mounted on:         /sdb_1
Filesystem UUID:         a0db16a0-e76d-4a72-b925-2ef49c2a6c25
Filesystem magic number: 0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              128000
Block count:              128000
```


❖ 파일 시스템 복구하기

dd	
기능	지정한 블록 크기만큼 파일을 복사한다.
형식	dd [if=파일] [of=파일] [bs=바이트 수] [count=블록수]
옵션	-if=파일 : 표준 입력 대신 지정한 파일에서 읽어온다. -of=파일 : 표준 출력 대신 지정한 파일을 복사한다. -bs=바이트수 : 한 번에 읽어보고 기록할 바이트 수다. -count=블록수 : 블록 수만큼만 복사한다.



```
user1@Server1:~/바탕화면$ sudo dd if=/dev/zero of=/dev/sdb1 bs=4096 count=20
```

```
20+0 레코드 들어옴
```

```
20+0 레코드 나감
```

```
81920 bytes (82 kB, 80 KiB) copied, 0.000454066 s, 180 MB/s
```

```
user1@Server1:~/바탕화면$ sudo mount /dev/sdb1 /sdb_1
```

```
mount: /sdb_1: wrong fs type, bad option, bad superblock on /dev/sdb1, missing codepage or helper program, or other error.
```

```
user1@Server1:~/바탕화면$ sudo e2fsck -b 32768 -y /dev/sdb1
```

```
e2fsck 1.46.5 (30-Dec-2021)
```

```
/dev/sdb1 was not cleanly unmounted, check forced.
```

```
Resize inode not valid.  Recreate? yes
```

```
Pass 1: Checking inodes, blocks, and sizes
```

```
Pass 2: Checking directory structure
```

```
Pass 3: Checking directory connectivity
```

```
Pass 4: Checking reference counts
```

```
Pass 5: Checking group summary information
```

```
Free inodes count wrong for group #0 (31989, counted=31988).
```

```
Fix? yes
```

```
Free inodes count wrong (127989, counted=127988).
```

```
Fix? yes
```

```
/dev/sdb1: ***** FILE SYSTEM WAS MODIFIED *****
```

```
/dev/sdb1: 12/128000 files (0.0% non-contiguous), 12215/128000 blocks
```

A photograph of a server room with rows of server racks on both sides of a central aisle. The racks have glass doors and internal components are visible, with many small blue lights glowing. The ceiling has several long, rectangular light fixtures. The overall atmosphere is dimly lit, emphasizing the blue light from the servers.

수고하셨습니다.