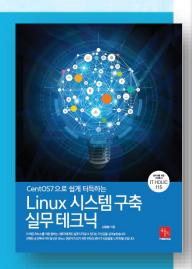
CHAPTER 7

파일시스템과 마운트 설정

Section 01 | 파일시스템의 개념과 종류

Section 02 | 파일시스템 마운트

Section **03** | 디스크 사용량 확인





파일시스템의 개념과 종류

• 파일시스템의 종류와 개념을 살펴보기



- [*******] 1. 파일시스템의 개념에 대해 이해합니다. 2 파일시스템의 종류에 대해 살펴봅니다
 - 2 파일시스템의 종류에 대해 살펴봅니다.
 - 3. 파일시스템에서 사용되는 데이터의 용량단위에 대해 살펴봅니다

파일시스템의 개념

- 하드디스크, DVD, USB 등과 같은 저장매체에 데이터를 저장하거나 삭제 또는 검색하는 방법을 제어하기 위한 시스템을 의미
- XFS 파일시스템은 RHEL ReadHat Enterprise Linux 7과 CentOS 7 에서 사용되고 있는 기본 파일시스템임
- 이 시스템은 높은 확장성과 고성능을 지원하는 64bit 파일시스템 으로
- 파일시스템을 사용하는 도중 장애가 발생하게 되면 파일시스템의
 빠른 복구를 지원하는 메타데이터 저널링을 지원하고
- 파일시스템의 온라인 통신 중 단편화 제거와 용량 확장이 용이함

파일시스템의 개념과 종류

- 저널링 기능은 파일시스템에 변경사항을 반영하기 전에 저널 안에 변경사항을 기록하고 추적하는 기능을 수행하여
- 시스템의 오류 또는 전원 문제 등으로인해 시스템이 재부팅되었을 경우 더 빠르게 파일시스템의 무결성 체크를 통해
- 신속하고 정확하게 마운트 될 수 있도록 편의성을 제공해 주는 시스템을 의미
- 저널링은 디스크에 기록되는 데이터의 복구 기능을 강화하여
 데이터를 디스크에 기록하기 전에
- 먼저 저널에 수정사항을 기록하여 문제가 발생할 경우 저널의 기록을 보고 빠르게 복구할 수 있도록 편의성을 제공

파일시스템의 개념과 종류

▋ 리눅스에서 지원하는 주요 파일시스템의 종류

[표 7-1] EXT 파일시스템의 특징

파일시스템의 유형	ext2	ext3	ext4
도입 시기	1993년	2001년	2008년
단일파일 최대사이즈	16GB~2TB	16GB~2TB	16GB~2TB
파일시스템 최대사이즈	2TB~32TB	2TB~32TB	1024PB
저널링 기능수행 여부	부	여	여
리눅스 커널버전	2.4.15 이전	2,4,15 부터	2.6.19 부터

참고로 파일의 용량 단위에 대해서도 살펴본다면 하드디스크 용량의 크기는 최근 기본 하드디스크의 용량을 의미하는 단위 $1TB^{Terabyte}$ 부터 살펴보도록 하겠습니다. 테라바이트가 모여 $2^{40}TB$ 가 되면 $1PB^{Petabyte}$ 가 되고 페타바이트가 모여 260PB가 되면 $1EB^{Exabyte}$ 가 됩니다. 그리고 엑사바이트가 모여 $2^{70}EB$ 가 되면 $1ZB^{Zettabyte}$ 가 되고 제타바이트가 모여 $2^{80}ZB$ 가 되면 $1YB^{Yottabyte}$ 가 됩니다. 요타바이트 정도는 미국의 델라웨어주에 해당하는 한 주를 다 채울 만큼의 데이터센터가 모여야 할 정도의 어마어마한 용량을 의미합니다. 이와 같은 용량 단위를 간략하게 표현하면 TB-PB-EB-ZB-YB와 같은 순으로 나열할 수 있습니다.

Section 01 파일시스템의 개념과 종류

■ 리눅스에서 지원하는 주요 파일시스템의 종류

[표 7-2] 리눅스 시스템에서 사용할 수 있도록 지원되는 파일시스템

파일시스템의 종류	가능
hpfs	https를 지원하기 위한 파일시스템
hfs	맥의 hfs 파일시스템을 지원하기 위한 파일시스템
iso9660	CD-ROM 또는 DVD의 표준 파일시스템으로 읽기 전용
msdos	MS-DOS 파티션을 사용하기 위한 파일시스템
nfs	네트워크 파일시스템으로 원격 서버와 디스크를 연결해 줌
ntfs	윈도우의 NTFS를 지원하기 위한 파일시스템
sysv	유닉스 시스템 V를 지원하기 위한 파일시스템
ufs	유닉스 표준 파일시스템을 지원하기 위한 파일시스템
vfat	윈도우 95, 98, NT를 지원하기 위한 파일시스템
xfs	레드햇과 CentOS 7의 기본 파일시스템

▋ 리눅스 가상 파일시스템

- 리눅스 시스템에서 사용되는 가상 파일시스템 VFS^{Virtual File System} 는 표준 유닉스 파일시스템과 관련된
- 모든 시스템 호출을 처리하는 커널 소프트웨어 계층과 실제 파일
 시스템의 구현을 일관된 형태로 인식하도록 해 줌
- 사용자 프로세스 사이에 존재하는 추상화 계층 및 다른 여러 종류의 파일시스템에 일반적인 공통 인터페이스를 제공하는 등의
- 특수한 용도에 따라 VFS는 존재했다가 사라지기도 함

파일시스템의 개념과 종류

[표 7-3] 리눅스 시스템에서 지원하고 있는 VFS

파일시스템의 종류	기능
proc	커널의 현재 상태를 나타내는 파일시스템 (/proc 디렉터리)
ramfs	RAM 디스크를 지원하는 파일시스템
rootfs	루트 파일시스템으로 시스템 초기화 및 관리에 필요한 내용 관리
swap	스왑 영역을 관리하기 위한 파일시스템
tmpfs	임시파일 저장을 위한 파일시스템 (/tmp 디렉터리)

VFS에 지원되는 파일시스템은 세 가지 그룹이 있습니다. 로컬 디스크 파티션의 기억장소를 관리하고 HDD, FDD, CD-ROM, DVD와 같은 블록 디바이스에 저장하기 위한 디스크 기반 파일시스템과 다른 네트워크의 컴퓨터에 속한 파일시스템에 쉽게 접근할 수 있도록 제공해 주는 네트워크 파일시스템 그리고 사용자가 커널 데이터 구조의 내용에 쉽게 접근할 수 있도록 간단한 인터페이스를 제공하는 특수 파일시스템 그룹이 있습니다.

파일시스템의 개념과 종류

▋ 현재 시스템이 지원하는 파일시스템

- 현재 컴퓨터에서 사용 중인 시스템에 따라 지원되는 파일시스템의
 목록을 확인하려면 cat 명령을 사용
- 현재 사용 중인 리눅스 파일시스템을 확인하기 위해서는 df 명령을 사용하며
- df 명령에 대해서는 다음 장에서 다루기로 하고 여기서는 cat 명령 에 대해서만 살펴보기로 함

Section 01 》파일시스템의 개념과 종류

| 예제 7 – 1 |

현재 사용 중인 시스템에서 지원하는 파일시스템의 목록을 cat 명령으로 출력합니다.

cat /proc/filesystems

기능 현재 시스템에서 지원하는 파일시스템 정보출력 형식 cat [옵션] 디렉터리명 또는 파일명 [Enter.]

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~] # cat /proc/filesystems
nodev
        svsfs
nodev
        root fs
nodev
       bdev
nodev
       proc
nodev
      cgroup
nodev
      cpuset
      tmpfs
nodev
        devtmpfs
nodev
nodev
       debugfs
nodev
       securityfs
nodev
       sockfs
nodev
       pipefs
nodev
       anon_inodefs
nodev
       configfs
nodev
       devpts
nodev
        ramfs
nodev
        hugetlbfs
nodev
        autofs
nodev
        pstore
nodev
        mqueue
nodev
        selinuxfs
        xfs
nodev
        rpc pipefs
nodev
        nfsd
        fuseblk
nodev
        fuse
nodev
        fusectl
[root@Localhost ~]#
```

[그림 7-1] 리눅스 시스템의 파일시스템에 대한 정보출력



파일시스템 마운트

• 파일시스템 마운트 설정 및 관리방법 이해하기

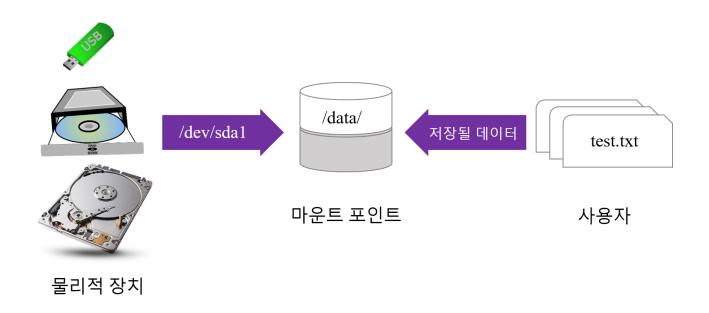


- [] 1. 마운트 포인터에 대해 이해합니다.
 - 2. 마운트 설정방법에 대해 살펴봅니다.
 - 3. 마운트 설정 및 관리방법에 대해 실습합니다.

마운트 포인트

- 리눅스 시스템은 계층적 디렉터리의 구조를 가지고 있기 때문에 파일 시스템이 디렉터리 계층구조와 연결되지 않으면 접근할 수 없음
- 특정 디렉터리와 USB 메모리, CD-ROM, DVD, HDD, FDD 등과 같은 외부장치와도 연결하는 것을 마운트라고 하며
- 디렉터리 계층 구조에서 파일시스템이 연결되는 디렉터리를 마운트 포인트라고 함

● 사용자가 사용하려고 하는 물리적인 장치를 특정 위치 즉 디렉터리에 연결시켜주는 마운트 과정을 그림으로 표현하면 다음과 같음



마운트 설정 파일

- 시스템 관리자가 일일이 마운트를 설정해야 한다면 상당한 번거로움이 초래됨
- USB 메모리나 CD-ROM, DVD와 같은 이동식 저장장치를 제외하고 는 대부분의 파일시스템 마운트 설정은 시스템 부팅 시 설정이 편리
- 리눅스 시스템이 부팅되면서 자동으로 파일시스템이 마운트 되도록 설정하려면 /etc/fstab 파일에 마운트 수행을 설정해야 함
- 시스템에 대한 마운트 설정과정은 반드시 root 계정으로 접속된 상태 에서 진행하는 것을 권장

/etc/fstab 파일의 용도

- /etc/fstab 파일은 리눅스 시스템이 부팅될 때 파일시스템에 대한 마운트 설정정보를 가지고 있는 파일이며
- 만약 이 파일에 오류가 있을 경우에는 시스템이 정상적으로 부팅되지 않음
- 리눅스 시스템이 부팅되면서 이 파일에서 설정한 내용에 따라 파일 시스템은 자동으로 마운트 되므로
- /etc/fstab 파일을 사용할 때는 파일시스템의 장치명과 마운트 포인 트. 그리고 마운트 할 때 설정할 옵션 등을 지정해 주어야 함

예제 7-2

현재 사용 중인 시스템에 설정되어 있는 /etc/fstab 파일의 구조를 살펴보기 위해 cat 명령으로 파일시스템에 대한 설정정보를 출력합니다.

cat /etc/fstab

기능 현재 시스템에 설정되어 있는 파일시스템의 정보출력

형식 cat [옵션] 디렉터리명 또는 파일명 Enter니

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~] # cat /etc/fstab
# /etc/fstab
# Created by anaconda on Wed Nov 30 21:39:35 2016
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5). findfs(8). mount(8) and/or blkid(8) for more info
/dev/mapper/centos-root /
                                                xfs
                                                        defaults usrquota
UUID=4485dbac-b5b1-451b-acbb-21089d698f9c /boot
                                                                  xfs
                                                                          defaul
         0 0
ts
/dev/mapper/centos-swap swap
                                                        defaults
                                                                        0 0
                                                swap
[root@localhost ~]#
```

[그림 7-4] 현재 시스템에 설정되어 있는 파일시스템의 정보출력

/etc/fstab 파일의 구조

● [예제 7-2]를 통해 [그림 7-4]와 같이 출력된 결과에서 /etc/fstab 파일의 구조적 측면에서 살펴보면 다음과 같음

장치명	마운트 포인트	파일시스템의 종류	옵션	덤프 관련 설정	파일 점검 옵션
/dev/mapper/centos -f00t	1	xfs	defaults	0	0

[그림 7-5] /etc/fstab 파일의 마운트 설정 정보출력

[표 7-4] 파일시스템 속성설정에 따른 옵션

옵션	기능
default	일반적인 파일시스템에 지정 (rw, suid, dev, auto, nouser, async 옵션 모두 포함)
ro	읽기 전용 마운트 (read−only)
rw	읽기, 쓰기 모두 가능한 마운트 (read—write)
auto	부팅 시 자동으로 마운트 (mount –a를 사용하면 auto 옵션이 장치 자동 마운트)
noauto	부팅 시 자동으로 마운트 하지 않음
exec	실행파일의 실행이 가능한 마운트
noexec	실행파일의 실행이 불가능한 마운트
user	일반 사용자의 마운트를 허용
nouser	일반 사용자의 마운트를 불허
async	파일시스템에 대한 입출력이 비동기적으로 마운트
sync	파일시스템에 대한 입출력이 동기적으로 마운트
remount	이미 마운트 된 파일시스템을 다시 마운트 (/etc/fstab 수정한 경우 또는 마운트를 잘못 수행한 경우)
dev	파일시스템의 문자, 블록 특수 장치를 해석할 수 있는 마운트
suid	setuid, getuid의 사용을 허가하는 마운트
nosuid	setuid, getuid의 사용을 불허하는 마운트
usrquota	사용자별 디스크 쿼터설정 마운트
grpquota	그룹별 디스크 쿼터설정 마운트
umask	마운트 된 디렉터리에서 새로 생성하는 파일과 디렉터리 퍼미션의 default 값을 지정 (박탈 개념으로 755 → 022)

마운트 관리 명령

● 리눅스 시스템에서 하드디스크의 파티션, CD/DVD, USB 메모리 등을 사용하기 위해서는 mount 명령으로 해당 디렉터리와 연결

mount

기능 주변장치와 파일시스템을 마운트

형식 mount [옵션] 장치명 마운트 포인트 Enter.

옵션 -t 파일시스템의 종류: 파일시스템의 종류 지정

-。마운트 옵션: 마운트 옵션 지정

-f: 마운트 할 수 있는지에 대해서만 점검

-r: 읽기만 가능한 마운트 (-o ro와 같은 의미)

● 하드디스크의 파티션, CD/DVD, USB 메모리 등을 해당 디렉터리 와의 연결을 해제할 때는 umount 명령을 사용

umount

기능 주변장치와 연결된 파일시스템의 연결을 해제 (언마운트)

형식 umount [옵션] 장치명 또는 마운트 포인트 EnterJ

옵션 -t 파일시스템 종류: 파일시스템의 종류를 지정하여 언마운트

● 마운트 기본정보 출력

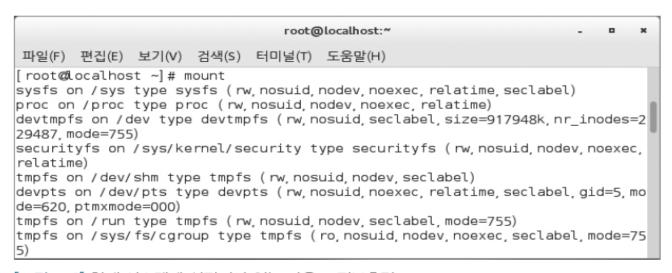
| 예제 7-3 |

현재 사용 중인 리눅스 시스템에 설정되어 있는 마운트 설정정보를 출력하기 위해서는 옵션 없이 mount 명령으로 마운트 설정 정보를 출력합니다.

mount

기능 현재 시스템에 설정되어 있는 마운트 정보 출력

형식 mount Enter↓



[그림 7-6] 현재 시스템에 설정되어 있는 마운트 정보출력

마운트 명령으로 장치 연결방법

[표 7-5] 파일시스템 또는 장치 마운트의 명령 형식

파일시스템 또는 장치	마운트 명령 형식
ext2 파일시스템	mount -t ext2 /dev/sdb1 /mnt
ext3 파일시스템	mount -t ext3 /dev/sdb1 /mnt
ext4 파일시스템	mount -t ext4 /dev/sdb1 /mnt mount /dev/sdb1 /mnt
xfs 파일시스템	mount -t xfs /dev/sdb1 /mnt mount /dev/sdb1 /mnt
CD-ROM	mount -t ISO9660 /dev/cdrom /mnt/cdrom
윈도우 디스크	mount -t vfat /dev/hdc/mnt
USB 메모리	mount /dev/sdb1 /mnt → 리눅스용 USB 메모리 mount −t vfat /dev/sdc1 /mnt → 윈도우용 USB 메모리
읽기 전용 마운트	mount -r/dev/sdb1/mnt
읽기/쓰기 마운트	mount -w /dev/sdb1 /mnt
원격 디스크 마운트	mount -t nfs 서버주소:/NFS 서버측 디렉터리 /mnt

디스크 파티션 명령

● 파티션은 fdisk 명령을 사용하여 새로 장착한 디스크의 파티션 작업 을 수행함

fdisk

기능 디스크의 파티션을 생성, 삭제, 관리하는 명령

형식 fdisk [옵션] [디스크 드라이브명] Enter니

옵션 -b 크기: 섹터의 크기 지정 (512, 1024, 2048, 4096)

-1: 디스크의 파티션 정보 출력

Section 02

파일시스템 마운트

USB 메모리 파티션 설정

│예제 7-6│

• Step 01 | 파티션을 설정하기 위해 fdisk 명령을 실행합니다. fdisk 명령 수행 후 화면에 나타 나는 Command : 프롬프트에 m을 입력하여 사용할 수 있는 명령을 살펴봅니다.

fdisk /dev/sdb1

기능 디스크의 파티션을 생성, 삭제, 관리하는 명령

형식 fdisk [옵션] [디스크 드라이브명] Enter니

옵션 -1: 디스크의 파티션 정보 출력

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~] # fdisk /dev/sdb1
Welcome to fdisk (util-linux 2,23,2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Command (m for help): m
Command action
   a toggle a bootable flag
   b edit bsd disklabel
   c toggle the dos compatibility flag
   d delete a partition
   g create a new empty GPT partition table
   G create an IRIX (SGI) partition table
  l list known partition types
   m print this menu
  n add a new partition
   o create a new empty DOS partition table
   p print the partition table
   q quit without saving changes
   s create a new empty Sun disklabel
  t change a partition's system id
  u change display/entry units
      verify the partition table
  w write table to disk and exit
  x extra functionality (experts only)
Command (m for help):
```

[그림 7-13] fdisk 명령 수행 후 사용할 수 있는 명령 리스트 출력

• Step 02 | 새로운 파티션을 생성하기 위해 Command : 프롬프트에 n을 입력합니다.

```
Command (m for help) : n

기능 새로운 파티션을 생성하는 명령
형식 Command (m for help) : n Enter

옵션 p: 기본 파티션 (primary)
e: 확장 파티션 (extended)
```

```
root@localhost:~ - ■ ×
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

Command (m for help): n
Partition type:
 p primary (1 primary, 0 extended, 3 free)
 e extended
Select (default p): ■
```

[그림 7-14] 새로운 파티션 생성 명령

• Step 03 │ 기본 파티션을 생성하기 위해 Command : 프롬프트에 p를 입력하고 Enter□키를 누르면 선택된 파티션 개수가 출력됩니다. 첫 번째 섹터의 디스크 용량을 2048로 입력하고 마지막 섹터의 디스크 용량을 나머지 용량으로 입력한 다음 Enter□를 눌러 파티션 생성을 완료합니다.

```
Command (m for help) : p
기능 기본 파티션을 생성하는 명령
형식 Command (m for help): p Enter---
번호 선택된 파티션의 개수
```

```
root@localhost:~ - 및 ★
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
Select (default p): p
Selected partition 3
First sector (2048-63505055, default 2048): 2048
Last sector, +sectors or +size{K, M, G} (2048-63505055, default 63505055): 63505055
Partition 3 of type Linux and of size 30.3 GiB is set
Command (m for help):
```

[그림 7-15] 각 파티션 별 디스크 용량 설정

• Step 04 | 파티션이 제대로 생성되었는지를 확인하기 위해 p를 입력합니다.

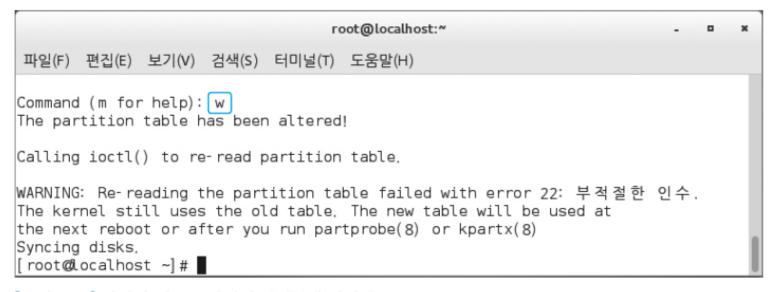
```
Command (m for help) : p
기능 새로 생성된 파티션을 확인하는 명령
형식 Command (m for help): p Enter
출력 파티션 생성 정보
```

```
root@localhost:"
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
Command (m for help): p
Disk /dev/sdb1: 32.5 GB. 32514588672 bytes. 63505056 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
Disk label type: dos
Disk identifier: 0x500a0dff
This doesn't look like a partition table
Probably you selected the wrong device.
                     Start
                                           Blocks Id System
     Device Boot
                                   End
/dev/sdb1p1 ? 1948285285 3650263507
                                        850989111+ 6e Unknown
/dev/sdb1p2 ?
                                                   74 Unknown
                      2048
                                                    83 Linux
/dev/sdb1p3
                              63505055
                                          31751504
/dev/sdb1p4
            28049408
                              28049848
                                              220+
                                                     0 Empty
Partition table entries are not in disk order
Command (m for help):
```

[그림 7-16] 새로 생성된 파티션 정보출력

• Step 05 | 파티션 정보를 파티션 테이블에 저장하고 종료하기 위해 w를 입력합니다.

```
Command (m for help) : w
기능 파티션 정보를 파티션 테이블에 저장하는 명령
형식 Command (m for help): w Enter니
```



[그림 7-17] 파티션 정보를 파티션 테이블에 저장하고 종료

Section 02

파일시스템 마운트

USB 메모리 파일시스템 생성 1: mkfs 명령

| 예제 7-7 |

• Step ①1 | USB 메모리를 포맷하여 파일시스템을 생성하기 위해 mkfs^{make file system} 명령과 파일 형식을 지정하여 xfs 파일시스템을 생성합니다. 리눅스 파일시스템을 생성하기 위해서는 mkfs 명령으로 파일시스템을 생성하기 전에 반드시 fdisk로 파티션을 생성시켜야 합니다. mkfs.xfs 명령은 mkfs -t xfs 명령과 동일한 의미입니다.

mkfs.xfs -f /dev/sdb1

기능 새로 생성된 파티션을 확인하는 명령

형식 mkfs.[파일시스템 타입] [디스크 드라이브명] Enter니

옵션 출력화면 참조

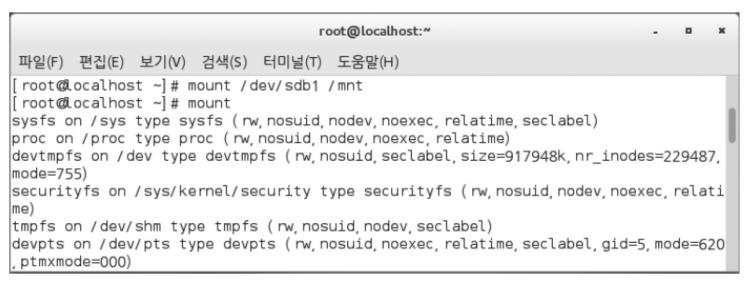
```
root@localhost:"
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~] # mkfs.xfs - f /dev/sdb1
mkfs.xfs: /dev/sdb1 contains a mounted filesystem
Usage: mkfs,xfs
/* blocksize */
                         [-b loa=n|size=num]
/* metadata */
                         [-m crc=0| 1, finobt=0| 1, uuid=xxx]
/* data subvol */
                        [-d agcount=n, agsize=n, file, name=xxx, size=num,
                             (sunit=value, swidth=value| su=num, sw=num| noalign),
                             sectlog=n|sectsize=num
/* force overwrite */
                         [ - f]
/* inode size */
                         [-i log=n|perblock=n|size=num, maxpct=n, attr=0|1|2,
                             projid32bit=0|1|
                         [ - K]
/* no discard */
                         [-l agnum=n, internal, size=num, logdev=xxx, version=n
/* log subvol */
                             sunit=value| su=num, sectlog=n| sectsize=num,
                             lazy-count=0|1|
                         [-L label (maximum 12 characters)]
/* label */
/* naming */
                         [-n log=n| size=num, version=2| ci, ftype=0| 1]
/* no-op info only */
                         [ - N]
/* prototype file */
                        [-p fname]
/* quiet */
                         [-a]
/* realtime subvol */ [-r extsize=num, size=num, rtdev=xxx]
/* sectorsize */
                         [-s log=n|size=num]
/* version */
                         [ - V]
                         devicename
(devicename) is required unless - d name=xxx is given.
(num) is xxx (bytes), xxxs (sectors), xxxb (fs blocks), xxxk (xxx KiB),
      xxxm (xxx MiB), xxxg (xxx GiB), xxxt (xxx TiB) or xxxp (xxx PiB).
(value) is xxx (512 byte blocks).
[root@localhost ~]#
```

• Step 02 | USB 메모리에 파일시스템을 생성하였으면 이 장치를 디렉터리에 연결하는 작업을 수행하기 위해 mount 명령을 사용합니다.

mount /dev/sdb1 /mnt

기능 주변장치와 파일시스템을 마운트

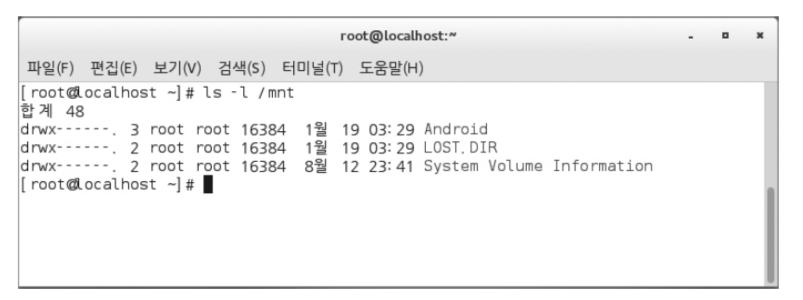
형식 mount [옵션] 장치명 마운트 포인트 Enter니



[그림 7-19] USB 메모리 장치를 /mnt 디렉터리에 마운트

• Step 03 | USB 메모리를 마운트한 디렉터리에 존재하는 디렉터리와 파일목록을 ls 명령으로 출력합니다.

```
# 1s -1 /mnt
기능 주변장치와 파일시스템을 마운트
형식 mount [옵션] 장치명 마운트 포인트 Enter↓
```



[그림 7-20] USB 메모리에 저장되어 있는 디렉터리 및 파일목록

USB 메모리 파일시스템 생성 2: mke2fs 명령

● mkfs가 확장된 mke2fs 명령어의 사용형식

mke2fs

기능 리눅스 확장개정판 파일시스템 생성

형식 mke2fs [옵션] 장치명 마운트 포인트 Enter→

옵션 -t 파일시스템의 종류: 파일시스템의 종류 지정 (기본 파일형식은 ext2)

-b 블록 크기: 블록 크기를 byte 수로 지정

-c: 배드블록 체크

-f 프래그먼트 크기 : 프래그먼트의 크기를 byte 수로 지정

-i I-node당 byte 수 : 기본 값은 4,096 byte

-m 예약블록 퍼센트: 슈퍼유저에게 예약해 둘 블록의 퍼센트 지정 (기본값 5%)

예제 7-8

mkfs 명령과 mke2fs 명령은 /sbin 디렉터리에 있습니다. ls /sbin/mk* 명령을 수행하면 쉽게 명령어를 찾을 수 있습니다.

1s /sbin/mk* 기능 mk로 시작하는 파일을 /sbin 디렉터리에서 찾기 형식 1s [옵션] 디렉터리명 또는 파일명 Enter.

```
root@localhost:~ - 미 *
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

[root@localhost ~] # ls /sbin/mk*
/sbin/mkdict /sbin/mkfs.btrfs /sbin/mkfs.fat /sbin/mkhomedir_helper
/sbin/mkdosfs /sbin/mkfs.cramfs /sbin/mkfs.minix /sbin/mklost+found
/sbin/mkdumprd /sbin/mkfs.ext2 /sbin/mkfs.msdos /sbin/mkswap
/sbin/mke2fs /sbin/mkfs.ext3 /sbin/mkfs.vfat
/sbin/mkfs /sbin/mkfs.ext4 /sbin/mkfs.xfs
[root@localhost ~] # ■
```

[그림 7-21] mk로 시작하는 파일목록 출력

| 예제 7-9 |

mke2fs 명령이 제공하는 별도의 설정파일은 /etc/mke2fs.conf 파일이며 이 파일에 대한 내용 은 cat 명령으로 살펴볼 수 있습니다.

cat /etc/mke2fs.conf

기능 /etc/mke2fs.conf 파일의 정보출력

형식 cat 디렉터리명 또는 파일명 Enter

Section 02 〉파일시스템 마운트

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~] # cat /etc/mke2fs.conf
[defaults]
        base features = sparse super, 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, uninit_bg, dir_nl
ink, extra_isize, 64bit
                inode size = 256
        ext4dev = {
                features = has journal, extent, huge file, flex bg, uninit bg, dir nl
ink,extra_isize
                inode_size = 256
                options = test fs=1
        small = {
                blocksize = 1024
                inode_size = 128
                inode_ratio = 4096
        floppy = {
                blocksize = 1024
                inode size = 128
                inode ratio = 8192
        biq = {
                inode ratio = 32768
        huge = {
                inode ratio = 65536
        news = {
                inode ratio = 4096
```

[그림 7-22] mke2fs 명령에서 제공하는 설정파일내용

Section 02 **파일/**

파일시스템 마운트

| 예제 7-10 |

• Step 01 | mke2fs 명령으로 USB 메모리의 파일시스템을 생성하기 위해 다음과 같이 명령을 수행합니다. mke2fs 명령을 사용할 때 파일시스템의 타입을 설정해 주지 않으면 기본값인 ext2 파일시스템 형식으로 생성됩니다. 이 부분을 실습하면서 주의깊게 살펴보기를 바랍니다.

mke2fs /dev/sdb1

기능 리눅스 확장개정판 파일시스템 생성

형식 mke2fs [옵션] 장치명 마운트 포인트 [Enter↓]

Section 02 기파일시스템 마운트

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# mke2fs /dev/sdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=O blocks Stripe width=O blocks
1986768 inodes, 7938132 blocks
396906 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
243 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000
Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
[root@localhost ~]#
```

[그림 7-23] mke2fs 명령으로 USB 메모리의 파일시스템 생성하기

Section 02 기 파일시스템 마운트

• Step 02 | USB 메모리에 파일시스템을 생성하였으면 이 장치를 디렉터리에 연결하는 작업을 수행하기 위해 mount 명령을 사용합니다.

mount /dev/sdb1 /mnt 기능 주변장치와 파일시스템을 마운트 형식 mount [옵션] 장치명 마운트 포인트 Enter니

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~] # mount /dev/sdb1 /mnt
[root@localhost ~]# mount
sysfs on /sys type sysfs (rw.nosuid.nodev.noexec.relatime.seclabel)
proc on /proc type proc (rw.nosuid.nodev.noexec.relatime)
devtmpfs on /dev type devtmpfs (rw.nosuid.seclabel.size=917948k.nr inodes=229487
mode=755)
securityfs on /sys/kernel/security type securityfs (rw.nosuid.nodev.noexec.relat
ime)
tmpfs on /dev/shm type tmpfs (rw.nosuid.nodev.seclabel)
devpts on /dev/pts type devpts (rw.nosuid.noexec.relatime.seclabel.gid=5.mode=62
0. ptmxmode=000)
tmpfs on /run type tmpfs (rw.nosuid.nodev.seclabel.mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro.nosuid.nodev.noexec.seclabel.mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw.nosuid.nodev.noexec.relatime.xa
ttr_release agent=/usr/lib/systemd/systemd-cgroups-agent_name=systemd)
```

Section 02 파일시스템 마운트

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
debugfs on /sys/kernel/debug type debugfs (rw.relatime)
mqueue on /dev/mqueue type mqueue (rw.relatime.seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw.relatime.seclabel)
sunrpc on /var/lib/nfs/rpc pipefs type rpc pipefs (rw.relatime)
nfsd on /proc/fs/nfsd type nfsd (rw.relatime)
/dev/sda1 on /boot type xfs (rw. relatime seclabel attr2 inode64 noguota)
tmpfs on /run/user/O type tmpfs (rw.nosuid.nodev.relatime.seclabel.size=186728k.
mode=700)
gvfsd-fuse on /run/user/0/gvfs type fuse.gvfsd-fuse (rw.nosuid.nodev.relatime.us
er id=0 group id=0)
fusectl on /sys/fs/fuse/connections type fusectl (rw.relatime)
/dev/sr0 on /run/media/root/VMware Tools type iso9660 (ro.nosuid.nodev.relatime.
uid=0.gid=0.iocharset=utf8.mode=0400.dmode=0500.uhelper=udisks2)
/dev/sdb1 on /mnt type ext2 (rw.relatime.seclabel)
[root@localhost ~]#
```

(b) USB 메모리 마운트 명령-2

[그림 7-24] USB 메모리 ext2 형식의 파일시스템 생성결과

Section 02 기 파일시스템 마운트

• Step 03 | USB 메모리에 존재하는 디렉터리와 파일목록을 출력합니다.

```
# 1s -1 /mnt
기능 USB 메모리에 존재하는 디렉터리와 파일목록 출력
형식 1s [옵션] 디렉터리명 Enter니
```

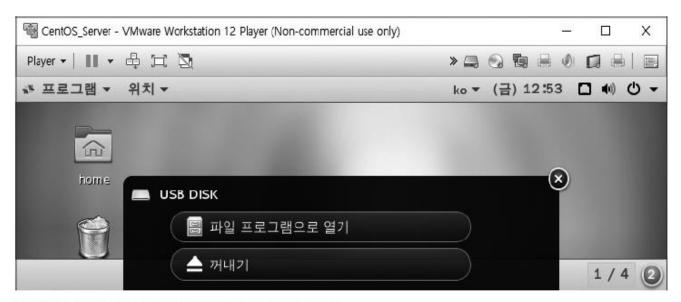


[그림 7-25] USB 메모리에 저장되어 있는 디렉터리 및 파일목록

X윈도 환경에서 USB 메모리 포맷

| 예제 7-11 |

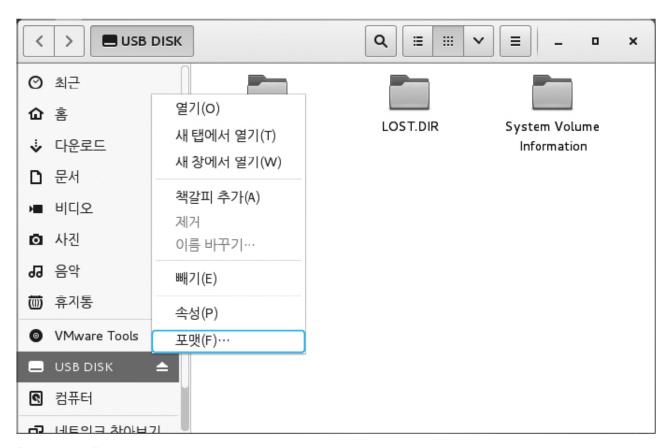
• Step 🔰 | 바탕화면 아래쪽에 연결 창이 나타나면 [파일 프로그램으로 열기]를 클릭합니다.



[그림 7-26] USB 메모리 장치에 있는 파일 열기

Section 02 과일시스템 마운트

• Step 02 | 왼쪽 목록의 [USB DISK]에서 마우스 오른쪽 버튼을 클릭하여 [포맷]을 클릭합니다.



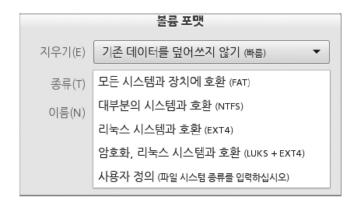
[그림 7-27] X윈도 환경에서 USB 메모리 포맷하기

Section 02 기 파일시스템 마운트

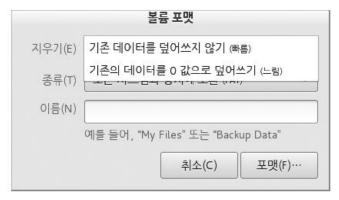
• Step 03 | 볼륨 포맷 형식의 항목을 각각 설정한 다음 〈포맷〉버튼을 클릭하면 USB 메모리에 대한 포맷이 진행됩니다.



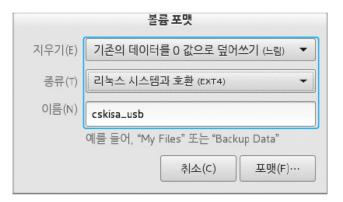
(a) 볼륨 포맷 화면



(c) 포맷 파일종류 선택



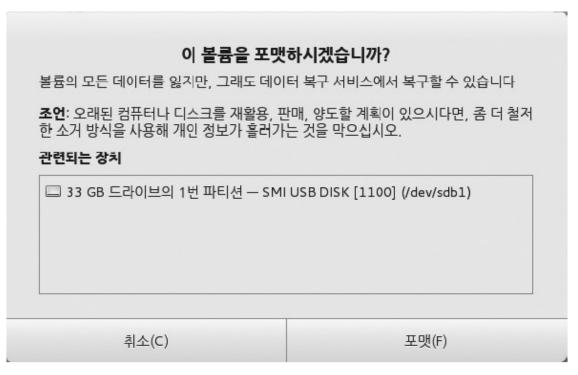
(b) 포맷 지우기 형식 설정



(d) 볼륨 이름 입력

Section 02 파일시스템 마운트

• Step 04 기정한 볼륨 포맷 형식으로 포맷을 진행할 것인지에 재차 확인하는 화면에서 〈포맷〉 버튼을 클릭하면 포맷이 진행됩니다. 여기서는 포맷하는 방법에 대해서 살펴보는 정도만 이해 하기로 하고 〈취소〉버튼을 클릭합니다.



[그림 7-29] USB 메모리 포맷진행 확인화면

Section 02

파일시스템 마운트

USB 메모리 장치 연결 해제

| 예제 7-12 |

USB 메모리 장치의 연결을 해제함에 있어서 먼저 체크해야 할 사항은 터미널 창에서의 프롬프트 가 어느 디렉터리에 있는지를 확인하는 것이 우선입니다. 반드시 루트 디렉터리에서 umount 명령을 수행해야 합니다.

umount /mnt

기능 주변장치와 연결된 파일시스템의 연결을 해제 (언마운트)

형식 umount [옵션] 장치명 또는 마운트 포인트 Enter↓

옵션 -t 파일시스템 종류: 파일시스템의 종류를 지정하여 언마운트

Section 02 과일시스템 마운트

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# umount /mnt
[root@localhost ~]# mount
sysfs on /sys type sysfs (rw.nosuid.nodev.noexec.relatime.seclabel)
proc on /proc type proc (rw.nosuid.nodev.noexec.relatime)
devtmpfs on /dev type devtmpfs (rw.nosuid.seclabel.size=917948k.nr inodes=229487.
mode=755)
securityfs on /sys/kernel/security type securityfs (rw.nosuid.nodev.noexec.relati
me)
tmpfs on /dev/shm type tmpfs (rw.nosuid.nodev.seclabel)
devpts on /dev/pts type devpts (rw.nosuid.noexec.relatime.seclabel.gid=5.mode=620
ptmxmode=000)
tmpfs on /run type tmpfs (rw.nosuid.nodev.seclabel.mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro nosuid nodev noexec seclabel mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw.nosuid.nodev.noexec.relatime.xat
tr, release agent=/usr/lib/systemd/systemd-cgroups-agent, name=systemd)
```

[그림 7-30] USB 메모리 장치 연결 해제

·

VMware에서 USB 메모리 장치 연결/해제

| 예제 7-13 |

• Step 01 | USB 메모리를 컴퓨터에 꽂으면 VMware 오른쪽 위에 나타나는 여러 개의 아이 콘 중에서 USB 아이콘에 커서를 올려놓고 마우스 오른쪽 버튼을 클릭하여 나타난 메뉴에서 [Connect(Disconnect from host)]를 클릭하면 USB 메모리가 마운트 됩니다.



[그림 7-31] VMware에서 USB 메모리 장치 연결

Section 02 과일시스템 마운트

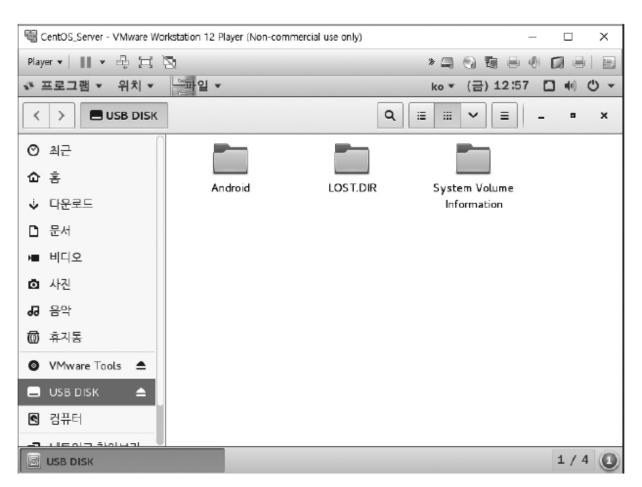
• Step 02 | 잠시 기다리면 바탕화면 아래쪽에 연결 창이 나타나면 [파일 프로그램으로 열기]를 클릭합니다.



[그림 7-32] USB 메모리 장치에 있는 파일 열기

Section 02 파일시스템 마운트

• Step 03 | USB 메모리에 저장되어 있는 디렉터리와 파일목록이 나타납니다.



[그림 7-33] USB 메모리에 저장되어 있는 디렉터리와 파일목록

Section 02 파일시스템 마운트

• Step 04 | VMware 오른쪽 위의 USB 아이콘에 커서를 올려놓고 마우스 오른쪽 버튼을 클릭하여 나타난 메뉴에서 [Disconnect (Disconnect from host)]를 클릭하면 USB 메모리 장치의 연결이 해제되어 가상머신에서 USB 메모리의 마운트가 완전히 해제됩니다.



[그림 7-34] VMware에서 USB 메모리 장치 해제

USB 메모리 장치가 VMware 가상머신에서 언마운트 되었음을 확인하였다면 컴퓨터 슬롯에서 USB 메모리를 제거해도 됩니다.



• 현재 사용 중인 디스크 사용량 확인하기



- [1. 현재 설치된 시스템에서 사용 중인 하드디스크 용량을 살펴봅니다. 2 파일시스템병 디스크 사용량 확인에 대해 실습합니다
 - 2. 파일시스템별 디스크 사용량 확인에 대해 실습합니다.
 - 3. 특정 디렉터리 또는 사용자별 디스크 사용량에 대해 실습합니다.

■ 파일시스템별 디스크 사용량 확인 : df

\$ df

기능 하드디스크의 남은 용량에 대한 정보를 출력

형식 df [옵션] [파일명] Enter↓

옵션 -a: 모든 파일시스템에서 사용 중인 디스크의 용량에 대한 정보출력

-B: 지정한 크기를 블록단위로 디스크의 용량을 출력

-h: 디스크 사용량을 보기 쉬운 단위(GB, MB, KB)로 출력

-H: 디스크의 용량을 1024Byte가 아닌 1000Byte 단위로 출력

-i:inode의 남은 공간과 사용 공간, 사용 퍼센트를 출력

-k: 디스크 사용량을 KB 단위로 출력

-m: 디스크 사용량을 MB 단위로 출력

-t 파일타입 : 지정한 파일타입의 종류에 해당하는 디스크 사용량 출력

-T: 파일시스템의 형태를 추가하여 각각의 파티션 정보 출력

-x 파일타입: 지정한 파일타입을 제외한 나머지 모든 파일시스템의 정보출력

옵션없이 df 명령만 사용하기: 1024byte 블록 단위로 출력

| 예제 7-14 |

이 실습도 반드시 root 계정으로 접속된 상태에서 수행해야 합니다. 리눅스 시스템에 접속된 상태가 root 계정임을 확인하였다면 df 명령으로 현재 사용 중인 리눅스 시스템의 디스크 사용량에 대한 정보를 출력합니다.

df

기능 현재 시스템에서 사용 중인 하드디스크의 사용량에 대한 정보출력

형식 df Enter→

Section 03 디스크 사용량 확인

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# df
Filesystem
                      1K-blocks
                                  Used Available Use% Mounted on
/dev/mapper/centos-root
                       28784916 5391612
                                        23393304 19%/
devtmpfs
                         926928
                                     0
                                          926928
                                                  0%/dev
tmpfs
                         942088
                                    88
                                          942000 1%/dev/shm
tmpfs
                         942088
                                  8876 933212 1%/run
tmpfs
                         942088
                                        942088
                                                  0%/sys/fs/cgroup
                                     0
                                 214528 294060 43%/boot
/ dev/ sda1
                         508588
tmpfs
                         188420
                                    16 188404 1%/run/user/0
[root@localhost ~]#
```

[그림 7-35] 현재 사용 중인 리눅스 시스템의 디스크 사용량에 대한 정보출력

사람이 이해하기 쉬운 단위로 디스크 사용량 출력 : -h 옵션

| 예제 7-15 |

현재 사용 중인 시스템의 디스크 사용량을 GB와 MB 단위로 출력되도록 df 명령과 함께 옵션 h를 함께 사용합니다.

df -h

기능 현재 시스템에서 사용 중인 하드디스크의 사용량에 대한 정보출력

형식 df [옵션] Enter↓

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# df -h
Filesystem
                               Used Avail Use% Mounted on
                         Size
/dev/mapper/centos-root
                               5. 6G
                          28G
                                      22G 21%/
                         906M
                                     906M
devtmpfs
                                  0
                                            0%/dev
                                     920M
                                            1%/dev/shm
tmpfs
                         921M
                                88K
                               8.7M 912M
tmpfs
                         921M
                                           1% / run
                                     921M
                                            0%/sys/fs/cgroup
tmpfs
                         921M
                                  0
/ dev/ sda1
                                     288M 43%/boot
                         497M
                               210M
tmpfs
                         185M
                                20K
                                           1% / run/ user/ 0
                                    184M
[root@localhost ~]#
```

[그림 7-36] 시스템의 디스크 사용량을 사용자가 이해하기 쉽도록 출력

Section 03 기다크 사용량 확인

파일시스템의 타입 출력: -T 옵션

| 예제 7-16 |

현재 사용 중인 시스템의 디스크 사용량을 파일시스템의 타입 별로 디스크 사용량에 대한 정보를 GB와 MB 단위로 출력하기 위해서 df 명령과 함께 옵션 - T옵션과 -h 옵션을 한꺼번에 사용합니다.

df -Th

기능 파일시스템을 타입 별로 디스크 사용량에 대한 정보출력

형식 df [옵션][옵션] Enter↓

```
root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# df - Th
Filesystem
                                  Size
                                        Used Avail Use% Mounted on
                        Type
/dev/mapper/centos-root xfs
                                   28G
                                        5. 6G
                                                22G 21%/
devtmpfs
                        devtmpfs
                                  906M
                                           0
                                              906M
                                                      0%/dev
tmpfs
                        tmpfs
                                  921M
                                         88K
                                              920M
                                                     1%/dev/shm
tmpfs
                        tmpfs
                                  921M
                                        8. 7M
                                              912M
                                                      1% / run
tmpfs
                        tmpfs
                                  921M
                                           0
                                              921M
                                                      0%/sys/fs/cgroup
/ dev/ sda1
                        xfs
                                  497M
                                        210M
                                              288M
                                                     43%/boot
tmpfs
                        tmpfs
                                  185M
                                         24K
                                              184M
                                                      1% / run/user/0
[root@localhost ~]#
```

[그림 7-37] 파일 타입 별 디스크 사용량 출력

실습 7-1 다음 항목에서 주어진 지시사항을 수행하시오.

- 1. 파일시스템의 디스크 사용량을 MB 단위로 출력하기
- 2 파일시스템의 디스크 사용량을 GB와 MB 단위로 출력하기
- 3. 현재 사용 중인 디스크 사용량을 파일시스템 타입 별로 GB와 MB 단위로 출력하기
- 4. tmpfs 파일시스템의 디스크 사용량 출력하기
- 5. 파일의 크기가 0인 파일시스템까지 출력하기

Section 03 디스크 사용량 확인

```
# df -m
# df -h
# df –Th
# df —t tmpfs
# df -a
```

Section 03 기 디스크 사용량 확인

■ 디렉터리 또는 사용자별 디스크 사용량 확인 : du

● du 명령은 특정 디렉터리 또는 사용자 별로 디스크 사용량을 확인 할 때 사용하는 명령

\$ du

기능 디렉터리 또는 사용자별 디스크 사용량에 대한 정보출력

형식 du [옵션] [디렉터리명 또는 계정이름] Enter→

옵션 -s: 모든 파일시스템에서 사용 중인 디스크의 용량에 대한 정보출력

-h: 디스크 사용량을 보기 쉬운 단위(GB, MB, KB)로 출력

옵션 없이 디렉터리 또는 사용자별 디스크 사용량 확인

| 예제 7-17 |

현재 디렉터리가 있는 위치가 어느 디렉터리 인지에 대해 pwd 명령으로 확인한 다음 du 명령을 사용해서 현재 디렉터리의 디스크 사용량에 대한 정보를 출력합니다.

pwd

기능 현재 위치한 디렉터리의 위치 정보출력

형식 pwd Enter↓

du

기능 현재 위치한 디렉터리의 디스크 사용량에 대한 정보출력

형식 du Enter↓

Section 03 디스크 사용량 확인

[그림 7-38] 현재 위치한 디렉터리의 디스크 사용량에 대한 정보출력

전체 디스크 또는 특정 디렉터리의 디스크 사용량 확인 : -s 옵션

| 예제 7-18 |

현재 위치한 디렉터리의 전체 사용량과 특정 디렉터리의 디스크 사용량을 확인하기 위해 du 명령과 -s 옵션 및 옵션 -h를 함께 선언하여 디스크 사용량에 대한 정보와 특정 사용자 계정을 지정하여 사용자 계정이 사용하고 있는 디스크 사용량에 대해서도 출력되도록 수행합니다.

du -s

기능 현재 위치한 디렉터리의 전체 사용량 또는 특정 디렉터리의 디스크 사용량 출력

형식 du [옵션] [디렉터리명 또는 계정이름] Enter↓

```
root@localhost:~ - □ ×
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

[root@localhost ~] # du -s
20820 .
[root@localhost ~] # du -sh /etc
32M /etc
[root@localhost ~] # du -sh /home/cskisa
29M /home/cskisa
[root@localhost ~] # ■
```

[그림 7-39] 특정 디렉터리와 특정 사용자 계정의 디스크 사용량 정보출력

실습 7-2 다음 항목에서 주어진 지시사항을 수행하시오.

- 1. 현재 위치한 디렉터리의 위치 출력하기
- 2. 현재 위치한 디렉터리의 전체 디스크 사용량을 MB 단위로 출력하기
- 3. /tmp 디렉터리의 서브 디렉터리까지 디스크 사용량 출력하기
- 4. /usr 디렉터리가 사용하고 있는 디스크의 용량을 GB와 MB 단위로 출력하기
- 5. 홈 디렉터리에 존재하는 사용자 계정이름 확인하기
- 6. 특정 사용자 계정 1개에 대한 디스크 사용량 출력하기

```
# pwd
# du -sh
# du /tmp
# du -sh /usr
# ls /home
# du -sh /home/cskisa
```

Chapter 07

최상의 노력에 따른 인고의 가치는 반드시 증명될 수 있습니다!

Thank You