

# 데이터 관리 II

2024.11.12

**DAWON**  
COMPUTING

# 목차

1. 데이터 관리 중점 사항
2. Proj Quota 및 Tape 활용 방법

# 데이터 관리 중점 사항

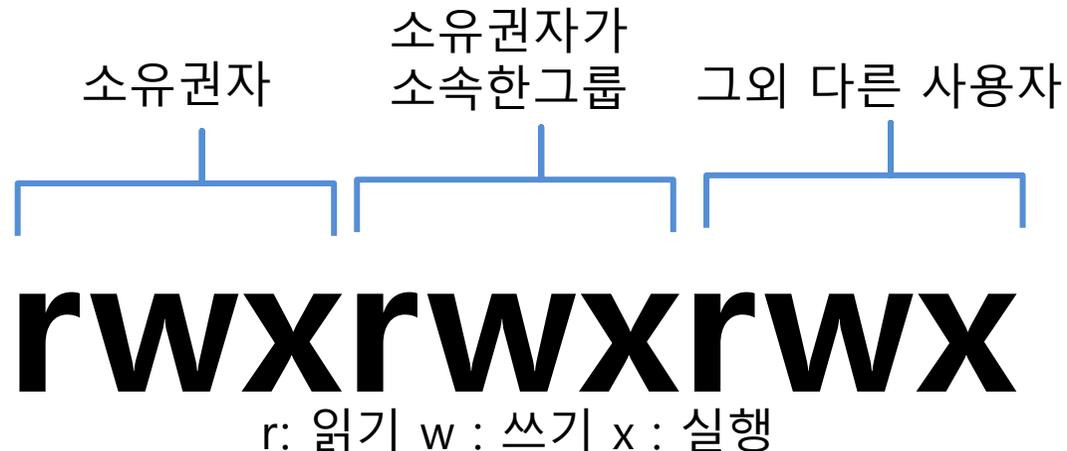
1. 권한 관리
2. 데이터 백업

# 1. 데이터 관리 중점 사항

## 1.1 파일 권한 관리

현재 기초과학연구원 초고성능 시스템은 다수의 불특정 사용자가 공유해서 사용하는 공유시스템임

- 다수의 사용자가 시스템을 활용 하다 보면, 실수 또는, 부주의로 인해, 타인에 파일 이나 디렉토리를 변경, 또는 삭제 할 수 있는 가능성이 있음
- 파일의 소유권 관리를 통해, 다른 사용자로 인해 발생하는 피해를 최소화 할 수 있음
- 파일 소유권 구성  
: 파일 권한에 따라 같은 그룹 또는 다른 사용자에게 권한을 부여할 수 있음  
아래 권한은 모든 사용자에게 전체 권한을 할당된 상태로 타인에 의해 파일이 삭제될 수 있는 위험 위험한 상태임



# 1. 데이터 관리 중점 사항

## 1.1 파일 권한 관리

아래 디렉토리 들 중 녹색으로 표시된 디렉토리가 전체로 권한이 모두 열린 상태임  
이경우, 시스템의 모든 사용자가 해당 디렉토리에 접근할 수 있고, 삭제도 가능함

```
total 173
drwxrwx---  3 adslab          adsl          4096 Apr 27  2023 adsl
drwxrwx---  6 root            ajou_ph       4096 Aug  1 14:42 ajou_ph
drwxrwx--- 13 jmchung        catholic_sbtl 4096 Sep 25 14:09 catholic_sbtl
drwxrwx---  5 root            46400        4096 Nov  5 20:35 davian
drwxrwx--- 11 tschon_hae      EnFRA         4096 Dec 18  2023 EnFRA
drwxrwx--- 10 blee            gachon_bsflaf 4096 Nov  6 20:18 gachon_bsflaf
drwxrwx---  3 root            gimpact       4096 Nov  7 14:42 gimpact
drwxrwx---  5 gistxray        gist_xray     4096 Dec 26  2023 gist_xray
drwxrwx---  5 jeon            hanyang_pl    4096 Apr 27  2023 hanyang_pl
drwxrwx---  3 root            hyntel        4096 Aug  5 15:05 hyntel
drwxrwxrwx  9 seunglae         jbnu_bc       4096 Dec 11  2023 jbnu_bc
drwxrwx---  4 heolab          kaist_heolab  4096 Apr 27  2023 kaist_heolab
drwxrwx---  7 jykang          kaist_mbpl    4096 Mar 12  2024 kaist_mbpl
drwxrwx--- 23 songj          kaist_sbe     4096 Sep 18 01:34 kaist_sbe
drwxrwx---  9 ohlab          kaist_tpdsb   16384 Mar 18  2024 kaist_tpdsb
drwxrwx---  7 hrkim          kasi          4096 Jun 14 11:10 kasi
```

```
ex ) [user@olaf]$ rm -rf /*
      [user@olaf]$ rm -rf /*
```

아래 명령어는 리눅스 최상위 디렉토리 밑에 있는 모든 파일을 삭제 하는 명령어로, 일반 사용자 계정으로 실행  
하면, 다른 디렉토리는 문제 없으나, 다른 사용자에게 권한이 허용된 jbnu\_bc 디렉토리는 삭제됨

# 1. 데이터 관리 중점 사항

## 1.1 리눅스 파일 권한 변경 명령어 [ chmod ]

파일의 권한을 변경하는 명령어를 이용해 기본적으로 권한을 관리해야함, 권한은 소유권자 본인만 허용하는 것이 좋으나, 협업을 위해 그룹 사용자에게 읽고 실행 권한까지 주는 것은 문제가 없음, (750 설정), 다만 타 사용자에게 권한을 열어 주는 것은 언제나 신중하게 작업해야함

사용 방법

```
[root@olaf2 ~]# chmod 700 dir1
```

: dir1 디렉토리를 소유권자에게만 전권을 부여

진수	2진수	권한	의미
0	0	---	아무 권한 없음
1	1	-x	실행 권한만 있음
2	10	-w-	쓰기 권한만 있음
3	11	-wx	쓰기, 실행 권한 있음
4	100	r--	읽기 권한만 있음
5	101	r-x	쓰기, 실행 권한 있음
6	110	rw-	읽기, 쓰기 권한 있음
7	111	rwx	모든 권한 있음

# 1. 데이터 관리 중점 사항

## 1.2 백업 (data backup) 필요성

- 디지털 데이터는 아래와 같은 다양한 원인으로 인해. 언제든지 삭제, 변경, 손망실 될 수 있음
  1. 하드웨어 고장 (proj는 다양한 데이터 보호기술이 적용되어 있어, 발생 확률은 0% 가깝게 현저히 낮음)
  2. 인적 오류 (개인 또는 팀원의 실수, 부주의, 관리자의 실수 등)
  3. 악성 소프트웨어 및 바이러스
  4. 자연 재해
  5. 기타 예상치 못한 사고
- 삭제가 된 데이터는 복구가 불가능한 경우가 많아, 그로 인해 프로젝트 진행에 큰 문제가 발생할 수 있음



백업 미 수행으로 인해, 발생한 불행한 사고 예시

# 1. 데이터 관리 중점 사항

## 1.2 백업 (data backup) 수행

- 백업은 최대한 많이 수행하여 여러 개의 백업본을 보유 하는것이 좋음
  1. 가능한 많이, 다양한 미디어(USB, CD, PC, workstation, 외장형 HDD)에, 지역을 나눠서 백업하는 것이 좋음
  2. 버전별로, 증분 백업 수행 ( 차 후 다양한 복구 시점을 가지고 복구를 수행할 수 있음
  3. 원본이 업데이트 되면 자동으로 백업본이 sync되는 방식은 편리하나, 원본에 문제가 발생했을 때, 백업 본도 같이 손상이 될 수 있어, 주의 해야함
  4. 천재지변을 대비하여 중요 데이터는 기초연, 학교 등 다양한 지역에 백업 하는 것을 권장 ( 소산백업 )
  5. 기초연은 별도의 백업 시스템을 제공하고 있지 않으므로, 수시로 자체 백업 수행을 권장함
  6. 백업된 데이터는 주기적으로 restore를 수행하여 데이터가 유효한지 확인

여러 곳에 백업을 하게 되는 경우, 보안 유출 사고 우려가 있음 .

백업 수행 시 백업 본에 대한 리스트를 관리하고, 만료된 데이터 또는 미디어는 반드시 확인 후 파기 해야함

# 목차

1. Proj 소개
2. 파일시스템 Quota
3. Tape 활용방법

## 2. 시스템 구성

### 2.1 Proj 파일시스템 구성 현황

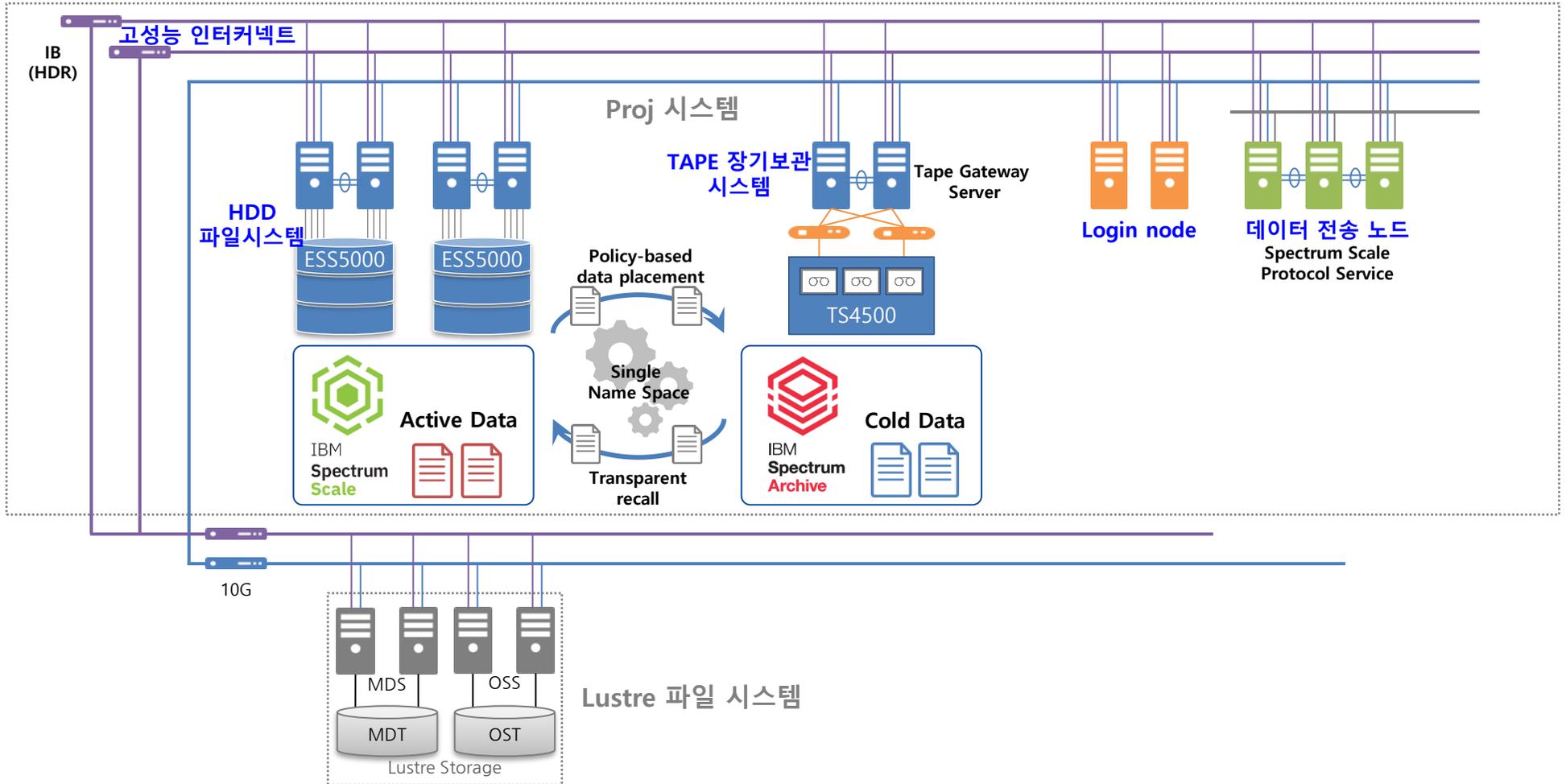
IBS 연구데이터 분석 및 저장을 위한 , proj 파일 시스템은 DISK 가용 용량 12.9PB, 백업 테이프 12TB \* 1000 EA로 시스템이 구성되어 있습니다..

구분	구분	구성	수량(식)
병렬 파일시스템	Spectrum Scale Appliance	IBM ESS5000 SC5 <ul style="list-style-type: none"> <li>Data Server : Power L922 x 2EA</li> <li>Storage : ESS 4U106 x 5EA</li> <li>Spectrum Scale 5.1.1</li> </ul>	2
		<b>가용 용량 : 12.9PB, 16TB NL-SAS HDD * 1,056개, 800GB SSD * 4EA</b>	
장기 파일보관 시스템	Tape Library	IBM TS4500 HD2 <ul style="list-style-type: none"> <li>BASE Frame + Expansion Frame</li> <li>물리 Slot : 2,050 Slot(24PB 확장 가능), <b>2,050 Slot 라이선스 포함</b></li> <li>TS1080 <b>LTO 8 TAPE DRIVE x 12EA</b></li> <li>비압축/고용량 12PB Tape 제공, 12TB Ultrium 8 Tape x 1,000개</li> </ul>	1
	Tape Gateway 서버	Lenovo ThinkSystem SR650 <ul style="list-style-type: none"> <li>Intel Xeon 4214R 12C 2.4GHz x 2EA</li> <li>128GB Memory(8EA x 16GB)</li> </ul>	2
	SAN Switch	Brocade G610	2
	아카이빙 솔루션	IBM Spectrum Archive Enterprise Edition	1
인프라 노드	로그인 서버	Lenovo ThinkSystem SR650 <ul style="list-style-type: none"> <li>Intel Xeon 6230R 26C 2.1GHz x 2EA</li> <li>192GB Memory(12EA x 16GB)</li> </ul>	2
	데이터 전송 노드	Lenovo ThinkSystem SR650 <ul style="list-style-type: none"> <li>Intel Xeon 4214R 12C 2.4GHz x 2EA</li> <li>128GB Memory(8EA x 16GB)</li> </ul>	<b>3</b>
고성능 인터커넥트	Infiniband	Mellanox QM8700 Switch <ul style="list-style-type: none"> <li>HDR 40-ports, 대역폭 : 100Gbps, 단자간 스위치 대기시간 : 90ns</li> </ul>	2
		3m EDR Cable	32
		100m EDR Cable	4

# 2. 시스템 구성

## 2.2 시스템 구성도

Proj 파일시스템 구성도



## 2. 시스템 구성

### 2.2 사용자 접속

접속을 한 후 `df -h` 명령어로 용량 확인 가능

용량은 DISK로 구성된 용량만 표시되며 Tape 용량은 별도로 표시 되지 않음

POSIX 기반 파일 시스템으로 리눅스 에서 제한 없이 사용 가능함, 다만 일부 용량제한(Quota)이 적용 되어있음

```
tmpfs 19742600 4 19742596 1% /run
[dawon@olaf1 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        95G   0    95G   0% /dev
tmpfs           95G   2.4G  92G   3% /dev/shm
tmpfs           95G   3.7G  91G   4% /run
tmpfs           95G   0    95G   0% /sys/fs/cgroup
/dev/mapper/r1-root 441G  115G  326G  26% /
/dev/sda2       2.0G  259M  1.8G  13% /boot
/dev/sda1       599M   5.8M  594M   1% /boot/efi
proj            12P   5.2P   6.5P  45% /proj
tmpfs           19G   24K   19G   1% /run/user/14901
tmpfs           19G   72K   19G   1% /run/user/0
tmpfs           19G   8.0K   19G   1% /run/user/1000
100.100.100.10:/mnt/lustre 2.3P  1.4P  906T  61% /mnt/lustre
tmpfs           19G   28K   19G   1% /run/user/14903
tmpfs           19G   0    19G   0% /run/user/31018
tmpfs           19G   48K   19G   1% /run/user/45001
```

## 2. 파일 시스템 쿼타(Quota) 기능

### 2.3 쿼타 Quota의 정의

쿼타란, 특정 사용자, 그룹 또는 파일 시스템에 대한 저장 용량 및 파일 개수를 제한하는 기능을 의미함  
한정적인 스토리지 자원의 효율적 분배 및 관리 공정한 사용을 위해 필요한 기능임

#### - 쿼타의 목적

자원 관리: 시스템 내의 저장 공간을 효율적으로 분배.

사용 패턴 제어: 특정 사용자가 지나치게 많은 공간을 사용하지 않도록 제한.

비용 절감: 더 많은 저장 공간이 필요할 경우 효율적 계획 수립 가능.

#### - 쿼타의 유형

블록 쿼타: 사용자 또는 그룹이 사용할 수 있는 최대 저장 용량을 정의.

파일 수 쿼타: 사용자 또는 그룹이 생성할 수 있는 파일의 최대 수를 정의.

#### - 쿼타 관리

쿼타 조회: `repquota` 또는 `quota` 명령어로 현재 사용과 쿼타 현황 확인.

모니터링: 시스템 관리자에 의해 정기적으로 쿼타 사용량을 모니터링하여 조정.

## 2. 파일 시스템 Quota 확인

### 2.3 mmlsquota 명령어

Proj에서 개인 또는 그룹 quota를 확인 하는 명령어

명령어 위치 : /usr/lpp/mmfs/bin/mmlsquota

그룹 Quota 확인

**/usr/lpp/mmfs/bin/mmlsquota -j 그룹명 --block-size=T /dev/proj**

-----  
TB로 용량표시    proj 지정

**Ex ) /usr/lpp/mmfs/bin/mmlsquota -j cup\_amore --block-size=T /dev/proj**

```
cupamore@daTape1 ~]$ /usr/lpp/mmfs/bin/mmlsquota -j cup_amore --block-size=T /dev/proj
      Block Limits
filesystem type      TB      quota      limit  in_doubt  grace |      File Limits
proj      FILESET    485        500        500        1      none |  files  quota  limit in_doubt  grace  Remarks
proj: Quota management has been enabled but quota accounting information is outdated. Run mmcheckquota to correct and update quota
formation.
```

## 2. 파일 시스템 Quota 확인

### 2.3 mmlsquota 명령어 스크립트

명령어 위치가 라눅스 기본 위치가 아니기에 경로를 모두 작성하여야 하고, 명령어가 쉽지 않아 기억하기 어려움

보다 손쉽게 quota를 확인할 수 있는 스크립트 활용

임의의 실행명을 주고 vim 실행 후 아래 내용 붙여넣기

```
[dawon@olaf1 ~]$ vim ./myquota
```

```
#!/bin/bash
```

```
# -j 옵션뒤에 확인하고자 하는 그룹 지정
```

```
output=$(/usr/lpp/mmfs/bin/mmlsquota -j cup_amore --block-size=T /dev/proj
```

```
# 파일 권한 부여
```

```
echo "$output" | awk 'NR==3 {print "USED: " $3, " TB " " Quota: " $5 " TB "}'
```

저장 후 실행 권한 주기

```
chmod 770 ./myquota
```

```
[dawon@olaf1 ~]$ chmod 770 myquota
[dawon@olaf1 ~]$ ./myquota
USED: 485 TB Limit: 500TB
```

## 2. Tape 드라이브 활용 (Archiving)

### 2.3 Tape 드라이브 활용 목적

#### 1. 데이터 아카이빙

- 대량의 데이터를 장기간 보관하는 데 적합
- 데이터 접근 빈도가 낮은 경우 비용 효율적인 대안
- 파일 시스템 용량에 포함되지 않기 때문에 Quota에 제한을 받지 않음

#### 2. 비용 효율성

- HDD 또는 SSD 대비 저장 용량 대비 낮은 비용으로 대량 데이터 저장 가능

#### 3. 내구성 및 수명

- 데이터의 보존 기간이 길고, 일반적으로 20년 이상의 저장 수명 제공

#### 4. 높은 저장 밀도

- 소형 물리적 공간에서 대량의 데이터를 저장 가능, 1개의 테이프에 12TB 저장 가능

#### 5. 데이터 보안성

- 물리적 분리로 인해 랜섬웨어 및 사이버 공격에 대한 불안감 감소

#### 6. 환경적 측면

- 비활성 상태에서 전력을 거의 소모하지 않음

## 2. Tape 드라이브 활용

### 2.3. Tape 드라이브로 파일을 옮기는 방법

각 그룹 디렉토리 안에 특정 위치에 파일을 옮겨 놓으면 일 1회 (새벽 01시) 자동으로 테이프로 이동

- 현재 Tape 드라이브로 이동 하는 디렉토리는 두 군데 디렉토리에 하위 디렉토리임
- /proj/internal\_group
- /proj/external\_group

위 두 디렉토리 아래 각 사용자 그룹 디렉토리 안에 tape\_archiving 디렉토리에 데이터를 넣어두면, 1일 1회 (새벽 1시) 정책에 따라 자동으로 Tape으로 파일이 이동함

#### Tape 으로 옮기는 파일 기준

- 1MB 이상 용량 파일
- 단일 파일 사이즈가 Tape 미디어 1장 용량 (12TB)를 넘지 않아야함
- : tape은 작은 파일에 효율이 매우 낮음, 작은 파일이 많은 경우 tar로 파일을 묶어서 처리
  
- Ex ) `tar cvfz tape_arch.tar tape_move_dic`
- `mv tape_arch.tar tape_archiving`

## 2. Tape 드라이브 활용

### 2.3. Tape 드라이브로 파일이 옮겨졌는지 확인 하는 방법

`ls -sh` 명령어로 확인해서 용량이 0으로 표시되면, Tape로 이동한 데이터 이고, 용량이 표시되면 스토리지에 있는 파일임

Ls명령어 중 `-s` 옵션을 사용하지 않은경우, 파일의 용량이 표시됨

```
[root@daTape1 tape_archiving]# ls -alth
total 3.0K
drwxr-xr-x 3 cupamore cup_amore 4.0K Sep 10 11:04 AMoRE-I_RAW
-rw-r--r-- 1 cupamore cup_amore 4.1T Sep 10 01:15 AMoRE-I_ONLANA.tar
drwxr-xr-x 3 cupamore cup_amore 4.0K Sep 9 17:55 .
drwxrwx--- 8 cupamore cup_amore 4.0K Feb 26 2024 ..
-rw-r--r-- 1 cupamore cup_amore 2.2T Dec 27 2023 2022_2nd_RODY_2.tar
-rw-r--r-- 1 cupamore cup_amore 2.2T Dec 27 2023 2022_2nd_RODY_1.tar
-rw-r--r-- 1 cupamore cup_amore 2.5T Dec 27 2023 2021_2nd_RODY_2.tar
-rw-r--r-- 1 cupamore cup_amore 2.4T Dec 27 2023 2021_2nd_RODY_1.tar
-rw-r--r-- 1 cupamore cup_amore 8.4T Dec 23 2023 2022_1st_RODY.tar
-rw-r--r-- 1 cupamore cup_amore 5.5T Dec 22 2023 2021_1st_RODY.tar
-rw-r--r-- 1 cupamore cup_amore 2.6T Dec 21 2023 2020_2nd_RODY.tar
-rw-r--r-- 1 cupamore cup_amore 7.4T Dec 18 2023 2020_1st_RODY.tar
-rw-r--r-- 1 cupamore cup_amore 1017G Dec 17 2023 2019_1st_RODY.tar
```

`ls-sh`로 확인 테이프로 이동한 파일 확인, 좌측에 용량이 0으로 표시됨

```
[root@daTape1 tape_archiving]# ls -sh
total 1.0K
 0 2019_1st_RODY.tar      0 2021_1st_RODY.tar      0 2022_1st_RODY.tar      0 AMoRE-I_ONLANA.tar
 0 2020_1st_RODY.tar      0 2021_2nd_RODY_1.tar    0 2022_2nd_RODY_1.tar  1.0K AMoRE-I_RAW
 0 2020_2nd_RODY.tar      0 2021_2nd_RODY_2.tar    0 2022_2nd_RODY_2.tar
```

## 2. Tape 드라이브 활용

### 2.3. Tape 드라이브로 파일이 옮겨진 파일 다시 파일 시스템으로 가져오기

일반 리눅스 시스템 명령어로 파일을 읽거나, 쓰면 자동으로 Tape에서 파일 시스템으로 가져오게 되어 있음

```
Ex ) mv backup.tar ../  
      cp backup.tar ../
```

다만 스토리지 대비 Tape이 매우 느리기 때문에, 파일 크기에 따라, restore 시간이 현저히 오래 걸릴 수 있음

**감사합니다**