



4. 파일 다루기

홍필두 교수
(리눅스프로그래밍)

● 주요내용

1. 파일 필터

- 1) 파이프와 grep 명령
- 2) 리다이렉션

2. 파일 비교, 정렬, 탐색

- 1) 파일 비교 (파일 2개의 내용 비교)
- 2) 파일 정렬
- 3) 파일 검색

3. 파일 보관 압축

- 1) tar ✓ (gzip를 관수 있음) - 묶은 룬드 (압축도 마크 관수 있음)
- 2) compress / uncompress (리눅스에는 패키지를 따르 선택해야해서 잘 안씀)
- 3) gzip / gunzip - 압축 룬드

● 1. 강의 들어가기

✓ 학습내용 소개

- 앞 강의에서는 리눅스 운영체제에서 파일에 대한 간단한 명령어를 다루었다. 여러분이 리눅스나 유닉스 시스템을 다루는 관리자이거나 프로그램, 데이터베이스를 다루는 업무의 담당자라면 파일을 깊게 다루는 경우가 많다. 각종 서버시스템의 운영 상황들이 실시간으로 기록되는 파일인 로그파일을 분석하기 위하여 파일을 여러 방면으로 다루어야 하며, 프로그램이나 데이터베이스의 입출력으로 일반적인 파일이 많이 사용되기 때문이다. 이번 강의에서는 리눅스에서 파일과 디렉토리 부분을 이해 후 실습하도록 한다.

✓ 학습목표 제시

- 파일 필터를 이해하고 사용하는 방법을 알 수 있다.
- 파일을 비교하고, 정렬하고, 필요한 파일을 찾을 수 있다.
- 파일 여러 개를 하나의 파일로 묶어서 보관할 수 있다.
- 파일의 크기를 줄이기 위한 압축과 복원 명령을 실행시킬 수 있다.

● 2. 생각해볼 문제 및 용어

✓ 학습전 생각해볼 문제

- 윈도우에서 파일을 다루는 유틸리티 프로그램이나 워드, 아래아 한글 등에서 기능 등을 사용하여 두 개의 파일의 내용을 비교하는 방법을 알아봅니다.
- 윈도우 엑셀 등의 프로그램에서 행을 정렬(sort)하는 방법에 대하여 알아봅니다.

✓ 용어 (강의 정리 시 필기 할 것)

- Root

● 3. 이해하기

2. 파일 다루기

앞 강의에서는 리눅스 운영체제에서 파일에 대한 간단한 명령어를 다루었다. 여러분이 리눅스나 유닉스 시스템을 다루는 관리자이거나 프로그램, 데이터베이스를 다루는 업무의 담당자라면 파일을 깊게 다루는 경우가 많다. 각종 서버시스템의 운영 상황들이 실시간으로 기록되는 파일인 로그파일을 분석하기 위하여 파일을 여러 방면으로 다루어야 하며, 프로그램이나 데이터베이스의 입출력으로 일반적인 파일이 많이 사용되기 때문이다. 이번 강의에서는 리눅스에서 파일과 디렉토리 부분을 이해 후 실습하도록 한다.

Tip

윈도우에서 파일을 다루는 유틸리티 프로그램이나 워드, 아래 아 한글 등에서 기능 등을 사용하여 두 개의 파일의 내용을 비교하는 방법을 알아보고, 학습 준비도를 높이도록 한다.

● 3. 이해하기

(1) 파일 필터

파일의 내용 중 원하는 부분을 검색하거나 파일에 내용을 쓰거나 출력하고 또는 기존파일에 추가하여 내용을 쓸 수 있는 데, 이는 파일 필터를 활용한 명령어를 이용한다

1) 파이프 (|)와 grep 명령

파일내 지정된 패턴이 있는지 찾아내기 위하여 파이프 필터(|)와 **grep** 명령을 사용 한다

① grep 검색 기능

· **grep**(Globally find Regular-Expression and Print : 지정된 표현식이 전체에 있는지 찾아서 프린트 함) 명령의 사용형식

· **grep patten filename** : 주어진 패턴이 있는지 해당 지정된 파일이나 또는 확장파일형식으로 검색

Tip

예1) `grep "gh" abc.txt` : abc.txt파일안에 gh라는 문자가 있으면 출력

Tip

예2) `grep "gh" *.txt` : 모든 .txt로 명칭이 끝나는 파일을 대상으로 파일안에 gh라는 문자가 있으면 출력하라

· **~ | grep patten** : 필터를 이용하여 검색하는 방식으로 주어진 패턴을 검색하는 방법은 동일함

Tip

예) `cat abc.txt | grep gh` : abc.txt파일을 전부출력하고, 그 출력을 받아서 gh라는 글자 를 찾음. 결국 같은 결과가 나옴

② grep 주요 옵션

● 3. 이해하기

- **w** : 전체 단어가 일치되는 경우출력
- **n** : 라인넘버 출력
- **v** : 단어가 일치하지 않는 경우출력
- **l** : 해당되는 파일명을 출력

Tip

예) `cat xinetd.conf | grep -nw telnetd` : xinetd.conf파일내용 중 “telnetd”라는 단어가 정확히 일치하게 들어간 라인 수를 출력

● 3. 이해하기

```
kopoctc@kopoctc:~/mydir$ cat abc.txt
abc
def
ghi
jkl
mno
p

kopoctc@kopoctc:~/mydir$ grep "gh" abc.txt
ghi
kopoctc@kopoctc:~/mydir$ cat abc.txt | grep gh
ghi
kopoctc@kopoctc:~/mydir$ cat abc.txt | grep -w gh
kopoctc@kopoctc:~/mydir$ cat abc.txt | grep -n gh
3:ghi
kopoctc@kopoctc:~/mydir$ cat abc.txt | grep -v gh
abc
def
jkl
mno
p

kopoctc@kopoctc:~/mydir$
```

) 랑금

<그림 1-47> 파일필터 명령어 사용

● 3. 이해하기

2) 리다이렉션(Redirection)

명령어의 결과를 다른 명령어의 입력으로 사용하는 경우나, 명령어의 결과를 파일에 기록하는 경우에 리다이렉션 기호를 사용한다.

① 파이프(|)

·명령 1 | 명령2 : (파이프) 어떤 명령의 결과를 받아 다른 명령을 실행

Tip

예) cat abc.txt | grep gh : 첫 번째 명령 abc.txt파일을 전부출력하고, 그 출력을 받아 서 두 번째 명령인 gh라는 글자를 찾음.

② 리다이렉션, 꺾쇠 (>, >>)

·명령 > filename : 어떤 명령의 결과를 지정된 명칭의 파일을 새로 생성하여 기록

·명령 >> filename : 어떤 명령의 결과를 지정된 명칭의 파일 뒤로 계속 붙여서 기록함

·명령 < filename : 어떤 명령의 입력으로 지정된 명칭의 파일을 사용함 - 잘 안씀

·예 1) cat xinetd.conf > a.file : xinetd.conf파일을 출력하고 이 결과를 a.file에 기록, 만일 a.file의 명칭으로 파일이 존재 하였다면, 이를 무시하고 새로 생성하고 기록함

Tip

예 1) cat xinetd.conf > a.file : xinetd.conf파일을 출력하고 이 결과를 a.file에 기록, 만일 a.file의 명칭으로 파일이 존재하였다면, 이를 무시하고 새로 생성하고 기록함

Tip

예 2) cat xinetd.conf >> a.file : xinetd.conf파일을 출력하고 이 결과를 a.file에 기록 하는데 기존 a.file의 맨끝부터 추가하여 기록함. 만일 a.file이 없다면 에러가 발생

● 3. 이해하기

Tip

예 3) `cat xinetd.conf | grep telnet > a.file` : 여러 가지 명령을 계속해서 연결하여 사용 할 수 있음. `xinetd.conf` 파일을 출력하고, 이 결과중 `telnet`이라는 문자열이 있으면 그 줄 만 출력하는데, 그 결과를 `a.file`에 기록함. 이때 `telnet`이라는 문자열이 있는지 화면에는 출력하지 않고 파일에 만 기록됨.

Tip

예 4) `cat > aa` : `cat` 명령어 뒤에 파일명이 지정되지 않으면 기본 키보드 입력을 받 음. `cat > aa`이라고 명령후 엔터를 치면 화면이 대기모드가 되며 키보드로 입력되는 값 을 받음. 키보드 입력을 하고 최종 `ctrl+c`로 입력을 종료하면 지금까지 키보드로 입력한 내용이 `aa`이라는 파일에 저장되어 있음 <그림 1-xx>

● 3. 이해하기

```
kopoctc@kopoctc:~/mydir$ cat aa
aaa
sasa
kopoctc@kopoctc:~/mydir$ cat > aa
aa
aa
^C
kopoctc@kopoctc:~/mydir$ cat aa
aa
aa
kopoctc@kopoctc:~/mydir$ cat >> aa
sdasa
sasa
^C
kopoctc@kopoctc:~/mydir$ cat aa
aa
aa s
dasa
sasa
```

<그림 1-48> 리다이렉션 명령어 사용

Tip

예 5) `grep gh > abc.txt : grep gh` 명령을 실행하는데 `abc.txt`가 입력으로 사용. 결국 `cat abc.txt |grep gh` 와 동일한 명령이 됨

● 4. 실습하기(1)

1)파일필터

- ① `grep hello abc.txt`
- ② `grep hello *`
- ③ `cat abc.txt | grep hello`
- ④ `ps -ef |grep http`
- ⑤ `grep -wn hello abc.txt`
- ⑥ `grep -wnv hello abc.txt`
- ⑦ `grep -l hello abc.txt`
- ⑧ `grep -l hello *`
- ⑨ `cat aa ; cat >aa; cat aa; cat >>aa;cat aa`

● 3. 이해하기

(2) 파일 비교, 정렬, 탐색

두 파일이 동일한 파일인지, 또는 다른 곳이 어디인지 알기 위하여 파일을 비교하거나, 원하는 파일이 어느 디렉토리에 위치해 있는지 파일을 찾기위한 명령어를 다룬다.

1) 파일 비교

파일을 비교하는 명령어는 **cmp, diff**의 명령이 있다. <그림 I -49>

① cmp

· **cmp file1 file2** : 두 개의(file1,file2) 파일을 비교하는 명령.

- 셸 스크립트 프로그래밍에서 사용하기 적합함
- 같으면 **exit code =0**을 반환 (셸 스크립트에서 사용)
- 다르면 **exit code =1**을 반환 일치하지 않는 첫째 **byte** 출력

② diff

· **diff file1 file2** : 두 개의(file1,file2) 파일의 차이를 보여 줌

- 두 파일 간 다른 부분을 각각 보여 주기 때문에 프로그램 작성시 프로그램 소스를 비교하는데 편리함

● 3. 이해하기

```
kopoctc@kopoctc:~/mydir$ cmp aa bb
aa bb differ: byte 4, line 2
kopoctc@kopoctc:~/mydir$ diff aa bb
2c2
< aa
___
>bb
4c4
< sasa
___
>cwsasa
kopoctc@kopoctc:~/mydir$
```

<그림 1-49> cmp,diff명령어의 차이

● 3. 이해하기

2) 파일 정렬

파일내용을 정렬하는 명령어는 **sort**의 명령이 있다. <그림 I -50>

① sort

- 파일내용을 정렬 조건에 따라 정렬

- 정렬기준인 필드를 선택하여 오름차순이나 내림차순으로 전체 줄을 순서대로 정렬

- 필드 기준은 빈칸을 기준으로 수행함

- sort abc.txt** : abc.txt파일을 오름차순으로 정렬함

- sort -r abc.txt** : abc.txt파일을 내림차순으로 정렬함

- sort -k2 abc.txt >result.txt** : 2번째 필드를 기준으로 정렬

● 3. 이해하기

```
kopoctc@kopoctc:~/mydir$ cat aa
0 h      six
3 a      one
4 g      nine
2 i      eight
5 d      three
kopoctc@kopoctc:~/mydir$ sort -kl aa
0 h      six
2 i      eight
3 a      one
4 g      nine
5 d      three
kopoctc@kopoctc:~/mydir$ sort -r aa
5 d      three
4 g      nine
3 a      one
2 i      eight
0 h      six
kopoctc@kopoctc:~/mydir$
```

<그림 1-50> sort 명령어

● 3. 이해하기

3) 파일 검색

파일을 검색하는 명령어는 **find** 명령이 있다. <그림 I -51>

① **find**

·원하는 조건의 파일의 위치를 찾아줌

1)사용법 : **find dirname -option1 ...-optionN**

·**dirname**은 주어진 디렉토리 및 이하 하위 디렉토리를 검색함

·**option**을 지정하여 다양한 조건과 실행법을 지정할 수 있음

2)주요 조건

·**-name** : 파일이름이나 패턴을 지정하여 찾을

Tip

예1) **find . -name abc.txt**: 현재 디렉토리부터 하위 디렉토리 까지 abc.txt파일을 찾을

Tip

예2) **find . -name *.txt** : 현재 디렉토리부터 하위 디렉토리까지 맨끝이 .txt로 끝나는 모든 파일을 찾을

·**-perm** : 파일의 퍼미션(권한)이 일치하는 파일을 찾을

● 3. 이해하기

예 쓰기 실행
4 2 1

Tip

예1) ^{나, 즉, 권한} find . -perm 775 ^{찾음} (: 현재 디렉토리부터 추후 하위 디렉토리까지 파일권한이 775인 파일을 찾아 권한을 다시 배움)

· **-type** : 파일의 파일유형이 일치하는 파일을 찾을 때

하위옵션) **-type b** : 문자로만 구성된 파일 찾기, **-type d** : 디렉토리 찾기, **-type f** : 일반적인 파일 찾기, **-type l** : 심볼릭 링크 찾기(심볼릭 링크는 추후 배움)

Tip

예) find . -type d -name abc : 현재 디렉토리부터 하위 디렉토리까지 이름이 abc인 디렉토리를 찾을 때

· **-user** : 파일의 사용자(user)가 일치하는 파일을 찾을 때

Tip

예) find / -user kopoc : 루트(/) 디렉토리부터 하위 디렉토리까지 파일의 사용자(만든 자 user)가 kopoc인 파일을 찾을 때

· **-group** : 파일의 그룹(group)이 일치하는 파일을 찾을 때

Tip

예) find / -group kopoc : 루트(/) 디렉토리부터 하위 디렉토리까지 파일의 그룹이 kopoc인 파일을 찾을 때

● 3. 이해하기

- `-atime +n/-n/n` : 최근 n 일 이전에 액세스된 파일을 찾아줌(`accessed time`)
 $+n$ 은 n 일 또는 그보다 더 오래 전의 파일을, $-n$ 은 오늘 부터 n 일 전까지의 파일을, n 은 정확히 n 일 전에 액세스된 파일을 찾아줌.

Tip

예) `find / -atime 2`: 루트(/) 디렉토리부터 하위 디렉토리까지 파일중 2일전에 접근한 (`access`)한 파일을 찾아줌

- `--print` : 찾은 파일명을 출력. 일반적으로 옵션을 주지 않아도 기본적으로 수행
- `--ls` : 찾은 파일에 대한 `ls`명령을 수행

● 3. 이해하기

Tip

예) find / -name my*.cnf -ls : 이 경우 my*.cnf의 형태의 파일을 찾아서 해당 파일을 ls명령을 실행함 <그림 1-xx>

```
root@kopoctc:~# find / -name my*.cnf -ls
262426      4 -rw-r--r--    1 root      root           2850 Oct 22
22:06
/usr/share/doc/mysql-server-5.5/examples/my-small.cnf
146136      4 -rw-r--r--    1 root      root           21 Jan 22
2021
/etc/mysql/conf.d/mysqld_safe_syslog.cnf
146131      4 -rw-r--r--    1 root      root          3505 Jan 22
```

<그림 1-51> ls옵션을 실행하여 find명령을 실행

● 4. 실습하기(3)

2)파일비교 정렬

- ① `cmp abc.txt abc2.txt`
- ② `diff abc.txt abc2.txt`
- ③ `diff -Dflag abc.c abc2.c`
- ④ `sort abc.txt , sort -r abc.txt` : 내림차순
- ⑤ `sort -k2 abc.txt >result.txt` : 2번째 필드를 기준으로 정렬

3)파일 찾기

- ① `find . -name '*.c' -print`
- ② `find . -mtime 14 -print`
- ③ `find . W(-name '*.c' -o -name '*.txt' W) -print : -o: OR`
- ④ `find / |grep abc` 이름에 abc가 들어있는 것은 모두
- ⑤ `find / -name abc` 이름이 abc인 것.

● 3. 이해하기

(3) 파일 보관, 압축

저장공간의 백업이나 자료를 옮기기 위하여 여러 파일을 묶어서 보관하거나 이를 압축하여 보관하고, 또한 파일의 압축을 풀 경우가 있는데, 이러한 파일의 보관과 압축을 위한 명령어를 다룬다.

1) tar (현재는 zip을 많이 사용하나 필요할 때가 있음)

tar은 묶고 압축

파일을 보관하거나 푸는 명령어는 tar의 명령이 있다. <그림 I-52>

① 일반적 기능

- 파일을 묶고 푸는 기능 (압축 기능 없음)
- 고전적으로 가장 많이 사용
- 파일을 묶어서 테이프에 저장 등에 사용

② 파일을 묶을 때 (-cvf)

· 사용법 : tar -cvf tarFileName fileList

- -c : fileList에 대한 tar형식의 백업 파일을 생성 (묶을 때) cf는 묶을 때 필수
- -v : 진행되는 상황을 설명 (진행상황보)
- -f : tar형식의 백업 파일 이름을 지정 (default: /dev/rmt/0 ...tape drive)

Tip

예) tar -cvf abc mydir : 만일 mydir이 디렉토리 일 경우 mydir디렉토리 이하 모든 파일과 디렉토리를 abc라는 파일명으로 묶음

● 3. 이해하기

Tip

예) `tar -cvf abc def ghi`: 이 경우 `def`와 `ghi`파일을 묶어서 `abc`라는 파일로 저장함. 첫 번째 지정된 명칭이 묶이는 파일명이 되는 것에 유의

③ 묶은 파일을 풀 때 (-xvf)

· 사용법 : `tar -txru tarFileName fileList`

- **-t** : tar형식의 백업 파일 안에 어떤 것들이 들어 있는지 목차만보임 (안써도 된다). 안에 뭐가 들어있는지 보임
- **-x** : 백업 파일로부터 파일을 추출복귀(extract) - 묶은 것은 푸는 기능
- **-r** : `fileList`를 기존의 백업 파일 뒤에 무조건 덧붙임(rear)
- **-u** : 기존의 백업 파일에 이미 포함되어 있는 `fileList` 중 수정된 파일들만을 백업 파일의 뒤에 덧붙임.

디렉토리가 있어도 recursive하게 적용 (업데이트)

Tip

예) `tar -xvf abc` : `abc`라는 파일이 tar명령어로 묶음파일일 경우 현재 디렉토리에 묶은 파일을 풀음

z cvf (압축하면서 풀는다.)

z xvf (압축은 하면서 풀다.)

Tip

예) `tar -xvf abc mydir` : `abc`라는 파일이 tar명령어로 묶음파일일 경우 `mydir` 디렉토리에 묶은 파일을 풀음, 만일 `mydir`이라는 디렉토리가 없다면 디렉토리를 만들고 풀음

④ 기타 tar 이용

· 서버의 테이프 디바이스에 저장또는 테이프 디바이스 읽기

- `tar -cvf tarfile /dev/rmt0` , `tar -xvf /dev/rmt0`

- 일반적으로 `cd`나 테이프 등 외장 저장매체는 `/dev/cd0`, `/dev/rmt0` 등으로 생성됨, 가상의 파일형식으로 만들어 짐

● 3. 이해하기

```
root@kopoctc:/home# tar -cvf a.tar kopoctc
kopoctc/
kopoctc/hepl3.txt
kopoctc/help2.txt
kopoctc/.viminfo
kopoctc/mydir/
kopoctc/mydir/myfile
kopoctc/mydir/abc.txt
kopoctc/mydir/aa
kopoctc/mydir/bb
kopoctc/.bashrc
kopoctc/.bash_logout
kopoctc/.bash_history
kopoctc/.profile
kopoctc/.cache/
kopoctc/.cache/motd.legal-displayed
root@kopoctc:/home# ls
a.tar kopoctc
root@kopoctc:/home#
```

<그림 1-52> tar명령어

2) compress / uncompress (잘 사용 안함)

파일을 압축과 푸는 명령어는 `compress / uncompress`의 명령이 있다.

① 압축 및 푸는 기능

- 파일을 압축 (파일 크기를 줄임)하고 푸는 명령어
- 압축하는 경우 `compress abc` 로 명령하면 `abc`파일이 압축되어 `abc.z`의 해당 이름의 `.z`의 확장자가 붙은 압축파일이 생성됨
- 압축을 푸는 경우 `uncompress abc.z` 로 명령하면 `abc.z`파일이 압축이 풀려서 `abc`로 복원된 파일이 생성됨
- 파일을 묶어서 압축하는 기능은 `gzip, gunzip` 명령어를 많이 사용하는데 이는 개인용PC에서 많이 쓰는 ZIP압축방식을 사용.
- `compress/uncompress`는 유닉스, 리눅스 서버환경에서 전통적으로 많이 사용되는 유틸리티 임.
- `zip`유틸리티보다 일반적으로 압축효율은 떨어짐
- 우분투 리눅스에서 `compress/uncompress`의 명령은 `apt-get install ncompress`로 유틸리티 설치 후 사용가능

Tip

예1) `compress *.txt` : 모든 *.txt를 확장자로 가진 파일들이 파일 하나하나 별로 압축 이 수행되며 해당 파일명+.Z의 명칭을 가진 파일이 여러 개 생김

Tip

예2) `uncompress *.Z` : 모든 Z 확장자를 가진 각각의 파일에 대하여 `uncompress` 풀기를 실행

3) gzip / gunzip

파일을 압축과 푸는 명령어로 **gzip / gunzip**의 명령이 있다.

① 압축 및 푸는 기능

- 많이 사용되는 파일을 압축 (파일 크기를 줄임)하고 푸는 명령어
- 압축하는 경우 **gzip abc** 로 명령하면 abc파일이 압축되어 abc.gz의 해당 이름의 .gz의 확장자가 붙은 압축파일이 생성됨

- 압축을 푸는 경우 **gunzip abc.z** 로 명령하면 abc.gz 파일이 압축이 풀려서 abc로 복원된 파일이 생성됨
- **gunzip** 명령은 **gzip -d** 명령과 동일

② gzip명령어의 주요 옵션

- **-d** : 압축을 푼다
- **-l** : 현재 압축된 파일의 내용을 보여준다.
- **-r** : 현재 디렉토리부터 하위 디렉토리까지 전부를 압축한다.
- **-t** : 압축된 파일의 완전성을 검사한다.
- **-v** : 압축 진행상황을 보여준다.
- **-9** : 최대한 압축한다.
(최대)

Tip

gzip * / gunzip *로 전체 파일을 대상으로 압축과 풀기를 실행 <그림 | -53>

```
root@kopoctc:/home/kopoctc# ls
help2.txt  hepl3.txt  mydir
root@kopoctc:/home/kopoctc# gzip *
gzip: mydir is a directory -- ignored
root@kopoctc:/home/kopoctc# ls
help2.txt.gz  hepl3.txt.gz  mydir
root@kopoctc:/home/kopoctc# gunzip *
gzip: mydir is a directory -- ignored
root@kopoctc:/home/kopoctc# ls
help2.txt  hepl3.txt  mydir
root@kopoctc:/home/kopoctc#
```

<그림 1-53> gzip/gunzip 명령어

● 4. 실습하기(3)

4) tar명령어

- ① `tar -cvf tarfile .` - 파일 추가
- ② `tar -tvf tarfile` - 파일 보기
- ③ `tar -rvf tarfile reverse.c` - 뒤에 파일 덧붙임
- ④ `tar -uvf tarfile reverse.c`
- ⑤ `mkdir ./tmp`
- ⑥ `cd tmp`
- ⑦ `tar -xvf ../tarfile`
- ⑧ `tar -xvf ../tarfile esc 키 1 번 tar -tf ../tarfile | grep 'lady*'`
- ⑨ `tar -xvf ../tarfile ./reverse.c`
- ⑩ `tar -cvf tarfile /dev/rmt0 , tar -xvf /dev/rmt0` - 작동 안함

● 4. 실습하기(3)

y 72고 복제할 줄 숫자 72고 아시는데도 P3 복사

5) compress, uncompress 명령어

- ① cp abc abc2
- ② compress abc // abc.Z과 abc2와 사이즈비교..
- ③ uncompress abc.z
- ④ compress *.txt : 수많은 Z파일
- ⑤ uncompress *.Z : 모든 Z 확장자 파일 uncompress

6) gzip, gunzip 명령어

- ① gzip -v9 tarfile
 - ② ls tarfile.gz
 - ③ gzip -l tarfile.gz
 - ④ gzip -vd tarfile.gz
 - ⑤ gunzip tarfile.gz
 - ⑥ gzip -vd tarfile.gz > tar.log &
- 여기다가 환죽 풀기
백그라운드로 돌리기
다른

● 4. 실습하기 (4 – Jump Up)

- 1) 금일 배운 명령어를 man으로 찾아보고 각종 옵션에 대하여 조사 후 실습
- 2) 금일 배운 명령어를 조합하여 복수의 명령어를 한번에 실행하는 방법을 실습
- ; , while : do done

● 5. 퀴즈

문제	보기	정답 및 해설
1) 다음 중 aa파일의 내용 중 tom이라는 단어가 일치한 부분을 찾아서 표시하라는 명령은	가) find tom aa 나) search tom aa 다) grep aa tom 라) grep tom aa	라) [grep 찾을 문자 대상파일]으로 명령어를 사용합니다.
2) 다음 중 파일 비교나 정렬 명령이 아닌 것은?	가) find 나) cmp 다) diff 라) sort	가) find는 원하는 파일을 찾는 명령어입니다.
3) aa이라는 파일이 text file일 때, cat aa > bb 와 결과가 동일한 명령은?	가) mv aa bb 나) cp aa bb 다) cp aa > bb 라) mv aa > bb	나) aa의 내용을 입력 받아 bb라는 파일로 기록하는 명령으로 결국 aa의 내용과 bb의 내용이 동일한 파일이 생기게 됩니다.
4) 파일을 묶음과 압축을 한번에 실행할 수 있는 명령은?	가) inittab 나) gzip 다) compress 라) tar	나) gzip은 파일을 묶음과 압축을 한번에 실행합니다.

● 6. 정리하기

✓ 다음 제시된 내용을 자필로 작성하여 제출 하시오 (상단 학번, 이름 기입)

1. 파일 필터

- 1) 파이프와 grep 명령에 대하여 설명하십시오
- 2) 리다이렉션 명령에 대하여 설명하십시오
- 3) 위 세개의 명령어를 복합한 명령어 사례를 몇 개 들고 설명하십시오

2. 파일 비교, 정렬, 탐색

- 1) 파일 비교 명령 및 주요옵션에 대하여 설명하십시오
- 2) 파일 정렬 명령 및 주요옵션에 대하여 설명하십시오
- 3) 파일 검색 명령 및 주요옵션에 대하여 설명하십시오

3. 파일 보관 압축

- 1) tar 명령 및 주요옵션에 대하여 설명하십시오
- 2) compress / uncompress 명령 및 주요옵션에 대하여 설명하십시오
- 3) gzip / gunzip 명령 및 주요옵션에 대하여 설명하십시오

● 7. 차시 예고

✓ 차시 학습내용

- 사용자,그룹, 권한관리에 대하여 이해할 수 있다.
- 그룹을 관리 할 수 있다.
- 파일이나 디렉토리의 소유자의 개념을 이해할 수 있다.
- 권한의 개념을 이해하고 변경 할 수 있다.
- 링크파일의 개념을 이해할 수 있다.

✓ 차시 준비

- 윈도우에서 일반 사용자를 만들어 관리자(Administrator)와 다른 권한을 주는 방법을 알아봅니다.
- 윈도우의 탐색기를 통하여 파일 속성([읽기전용] [숨김] 등)을 바꿔봅니다.
- 바탕화면의 아이콘과 실행파일(*.exe)의 차이를 알아봅니다.