

1. 파일 필터

1) 파이프(|)와 grep 명령

“ 파이프 필터(|)와 grep 명령을 이용하여
파일에 지정된 패턴을 찾아냄 ”

✓ grep(Globally find Regular-Expression and Print) 명령의 사용형식

- grep pattern filename : 파일명 또는 확장파일명으로 검색
- | grep pattern : 필터를 이용하여 검색

1) 파이프(|)와 grep 명령

구분	설명
- W	전체 단어가 일치되는 경우 출력
- n	라인넘버 출력
- v	단어가 일치하지 않는 경우 출력
- l	해당되는 파일명을 출력

예

```
cat xinetd.conf|grep -nw telnetd
```

1) 파이프(|)와 grep 명령

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

〈출처 : 교수자 저작물〉

2) Redirection

명령1 | 명령2

(파이프) 어떤 명령의 결과를 받아 다른 명령을 실행

명령 > filename

어떤 명령의 결과를 파일을 새로 생성하여 기록

명령 >> filename

어떤 명령의 결과를 파일 뒤로 계속 첨부기록

예

- cat xinetd.conf > a.file
- cat xinetd.conf >> a.file
- cat xinetd.conf|grep telnet > a.file

2) Redirection(계속)

“ 리다이렉션의 방향에 따라 입력과 출력의 역할을 함 ”

구분	설명
0	표준 입력
1	표준 출력
2	표준 에러

2) Redirection(계속)

✓ 표준 입출력, 예러는 서버에 지정된 콘솔 모니터, 키보드를 의미함

✓ 실질적으로 VM에서는 실습이 어려움



명령 > file or 숫자

명령의 결과출력을
file(또는 표준 입출력)에 기록



명령 < file or 숫자

명령의 입력을 받아
file(또는 표준 입출력)에 기록

2) Redirection(계속)

```
sjcu@sjcu:~/mydir$ cat aa
aaa
sasa
sjcu@sjcu:~/mydir$ cat > aa
aa
aa
^C
sjcu@sjcu:~/mydir$ cat aa
aa
aa
sjcu@sjcu:~/mydir$ cat >> aa
sdasa
sasa
^C
sjcu@sjcu:~/mydir$ cat aa
aa
aa
sdasa
sasa
```

〈출처 : 교수자 저작물〉

2. 파일 비교 · 정렬 · 탐색

1) 파일 비교

cmp

두 파일을 **비교함**

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

diff

- 두 파일의 **차이를 보여줌**
- 프로그램 소스 등을 비교할 때 편리함

1) 파일 비교

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

〈출처 : 교수자 저작물〉

2) 파일 정렬

sort

파일 내용을 정렬 조건에 따라 정렬

- `sort abc.txt` , `sort -r abc.txt` : 내림차순
- `sort -k2 abc.txt >result.txt` : 2번째 필드를 기준으로 정렬

2) 파일 정렬

```
sjcu@sjcu:~/mydir$ cat aa
0 h six
3 a one
4 g nine
2 i eight
5 d three
sjcu@sjcu:~/mydir$ sort -k1 aa
0 h six
2 i eight
3 a one
4 g nine
5 d three
sjcu@sjcu:~/mydir$ sort -r aa
5 d three
4 g nine
3 a one
2 i eight
0 h six
sjcu@sjcu:~/mydir$
```

〈출처 : 교수자 저작물〉

3) 파일 탐색

find 디렉토리 - 조건 - 실행

특정 조건에 맞는 파일을 찾아서 원하는 동작 실행함

1 조건

- -name : 파일이름 패턴
- -perm : 퍼미션
- -type : 파일유형
- -user : 유저
- -group : 그룹
- -mtime : 최근에 수정
- -atime : 최근에 접근

3) 파일 탐색

find 디렉토리 - 조건 - 실행

특정 조건에 맞는 파일을 찾아서 원하는 동작 실행함

2 실행

- -print : 파일명 출력
- -ls : ls수행
- -exec : 명령어 수행
- 많이 사용하는 방법 : find / |grep myfi

3) 파일 탐색

예

- `find . -name '*.c' -print`
- `find . -mtime 14 -print`
- `find . \W(-name '*.c' -o -name '*.txt' \W) - print`
- `find / |grep abc`
- `find / -name abc`

```
root@sjcu:~# 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 2014 /etc/mysql/conf.d/mysql
d_safe_syslog.cnf
146131 4 -rw-r--r-- 1 root root 3505 Jan 22 2014 /etc/mysql/my.cnf
root@sjcu:~#
```

〈출처 : 교수자 저작물〉

3. 파일 보관 · 압축

1) 파일 묶어서 보관(tar)

✓ 파일을 묶고, 푸는 기능(단, 압축 기능은 없음)

✓ 고전적으로 가장 많이 사용하는 방법

✓ 파일을 묶어서 테이프에 저장 등에 사용

1) 파일 묶어서 보관(tar)

➤ 파일을 묶을 때 (tar -cvf tarFileName fileList)

구분	설명
-c	fileList에 대한 tar 형식의 백업 파일을 생성
-v	진행되는 상황을 설명
-f	tar 형식의 백업 파일 이름을 지정 (default : /dev/rmt/0 ... tape drive)

1) 파일 묶어서 보관(tar)

➤ 파일을 풀 때 (tar -txru tarFileName fileList)

구분	설명
-t	tar 형식의 백업 파일 안에 어떤 것들이 들어 있는지 목차만 보임
-x	백업 파일로부터 파일을 추출 복귀(extract)
-r	fileList를 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

1) 파일 묶어서 보관(tar)



참고

- tar 명령은 cd(dvd)드라이브나 tape드라이브에 저장 및 복원명령어로 많이 사용됨
- ex) tar -cvf tarfile /dev/rmt0
- ex) tar -xvf /dev/rmt0

1) 파일 묶어서 보관(tar)

```
root@sjcu:/home# tar -cvf a.tar sjcu
sjcu/
sjcu/hep13.txt
sjcu/help2.txt
sjcu/.viminfo
sjcu/mydir/
sjcu/mydir/myfile
sjcu/mydir/abc.txt
sjcu/mydir/aa
sjcu/mydir/bb
sjcu/.bashrc
sjcu/.bash_logout
sjcu/.bash_history
sjcu/.profile
sjcu/.cache/
sjcu/.cache/motd.legal-displayed
root@sjcu:/home# ls
a.tar sjcu
root@sjcu:/home#
```

〈출처 : 교수자 저작물〉

2) 파일 압축 및 풀기

`compress abc`

abc.z의 압축파일 생성

`uncompress abc.z`

abc의 복원 파일 생성

✓ 파일을 묶어서 압축하는 기능은 `gzip, gunzip` 사용

- `gzip, gunzip`은 고전적으로 많이 사용되던 유틸리티

✓ 우분투 리눅스에서 `apt-get install ncompress`로 유틸리티 설치 후 사용 가능

2) 파일 압축 및 풀기

예

- `cp abc abc2`
- `compress abc // abc.Z`과 `abc2`와 사이즈비교
- `uncompress abc.z`
- `compress *.txt` : 수많은 Z 파일
- `uncompress *.Z` : 모든 Z 확장자 파일 `uncompress`

3) 파일 묶음과 압축시행

“ 파일을 묶고 압축하는 유틸리티 ”



gzip

파일을 압축하여 묶음



gunzip

파일을 복원하여 풀음
(gzip-d 과 동일)

3) 파일 묶음과 압축시행

“ 파일을 묶고 압축하는 유틸리티 ”

〈gzip 옵션〉

구분	설명
-d	압축을 풀
-l	현재 압축된 파일의 내용을 보여줌
-r	현재 디렉토리부터 하위 디렉토리까지 전부 압축함
-t	압축된 파일의 완전성을 검사함
-v	압축 진행 상황을 보여줌
-9	최대한 압축함

3) 파일 묶음과 압축시행

```
root@sjcu:/home/sjcu# ls
help2.txt hep13.txt mydir
root@sjcu:/home/sjcu# gzip *
gzip: mydir is a directory -- ignored
root@sjcu:/home/sjcu# ls
help2.txt.gz hep13.txt.gz mydir
root@sjcu:/home/sjcu# gunzip *
gzip: mydir is a directory -- ignored
root@sjcu:/home/sjcu# ls
help2.txt hep13.txt mydir
root@sjcu:/home/sjcu#
```

〈출처 : 교수자 제작물〉

4. 파일 필터 비교 정렬 찾기 실습

1) 실습하기

실습내용

- (1) 파일필터
- (2) 파일 비교 정렬
- (3) 파일 찾기

5. 파일 보관 압축 실습

1) 실습하기

실습내용

- (1) 파일 묶어서 보관
- (2) 파일 압축 및 풀기
- (3) 파일 묶음과 압축시행



학습활동

* 일시정지 버튼을 클릭하고 학습활동에 참여해 보세요.

Q

파일을 다루는 명령을 여러 가지 방법으로 응용하여 사용할 수 있습니다.
인터넷 검색을 통해 다양하게 사용하는 방법을 실습해 보세요.

※ 학습활동에 대한 해설

Q 파일을 다루는 명령을 여러 가지 방법으로 응용하여 사용할 수 있습니다. 인터넷 검색을 통해 다양하게 사용하는 방법을 실습해 보세요.

- A**
- 오늘 배운 리눅스 명령어는 앞으로 셸 프로그래밍을 배울 때, 유용하게 활용될 것입니다.
 - 리눅스, 유닉스에서는 “파일 비교 정렬 탐색”을 셸 명령어 상태에서 사용하였으나, 윈도우에서는 메모장이나 워드 프로세스에서 지원하는 다양한 기능을 이용하여 적용할 수 있습니다.
 - “파일 보관 압축”은 파일을 보관하거나 백업하는 경우, 압축하여 파일 크기를 줄여서 유용하게 사용할 수 있습니다.
 - 명령어 중 zip 명령어는 현재 PC에서도 동일하게 사용되고 있습니다.