

## 6.6 파일 보관 및 압축

---

### ■ 파일 압축 유틸리티 - gzip, bzip2, xz

- compress/uncompress – 유닉스 압축 유틸리티 (.Z 형식의 압축파일)
  - 리눅스에서는 특허권 문제로 제공되지 않음
- gzip, gunzip – gz 형식의 압축파일 생성/해제
- bzip2, bunzip2 – bz2 형식의 압축파일 생성/해제
- xz, unxz – xz 형식의 압축파일 생성/해제

\$ gzip file

\$ gunzip file[.gz]

\$ bzip2 file

\$ bunzip2 file[.bz2]

\$ xz file

\$ unxz file[.xz]

\$ znew file.Z                   ... .Z 압축파일을 .gz 압축파일로 변환

# 파일 압축

---

## ■ 압축 파일 내용 읽기

\$ zcat file

\$ bzip file

\$ xzcat file

## ■ 압축 보관 – zip

- zip/unzip – 윈도우의 pkzip과 호환되는 압축 유틸리티

- 디렉토리 구조의 파일들을 하나의 압축파일로 묶음

\$ zip file.zip file1 file2 file3 ... 압축보관파일 file.zip 생성

\$ zip -r backup.zip datadir ... 디렉토리 재귀적 압축

\$ unzip file[.zip]

# 파일 보관(archive)

## ■ tar (tape archive)

- 여러 개의 파일/디렉토리를 묶어서 tape로 백업하여 보관 (원래)
- tape대신 archive 파일로 지정하여 저장할 수 있음 (주로 사용)

### ■ **c**reate

- \$ tar cvf file.tar d\* ... 지정된 파일들을 file.tar에 보관, v: verbose
- \$ tar cf file2.tar dir ... dir 디렉토리 이하 파일을 file2.tar에 보관
- \$ tar cv . ... 현재 디렉토리를 /dev/rmt8 (tape)에 보관

### ■ **t**able (목록)

\$ tar tvf file.tar

### ■ **e**xtract

- \$ tar xvf file.tar ... file.tar → 현재 디렉토리에 추출
- \$ tar xvf file.tar main.c ... file.tar(main.c) → ./main.c

### ■ **a**ppend

- \$ tar uvf file.tar newfile ... newfile을 file.tar에 추가 (최신버전만)
- \$ tar rvf file.tar oldfile ... oldfile을 file.tar에 추가 (무조건)

# tar 파일 압축 보관

## ■ tar 파일 압축 보관

```
$ tar cvf - file .. // -는 archive파일을 표준출력으로 지정  
$ tar cvf - file .. | gzip > my.tar.gz // tar 파일 압축 저장  
$ tar cvf my.tar file .. ; gzip my.tar // tar 파일 저장 후 압축
```

## ■ tar의 압축/압축해제 옵션

z (gzip), j (bzip2), J (xz), Z (compress)

```
$ tar cvfz my.tar.gz file .. // 실제로 확장자 사용에 제약은 없음  
$ tar cvfj my.tar.bz2 file ..  
$ tar cvfJ my.tar.xz  
$ tar tvfz my.tar.gz 또는 tar tvf my.tar.gz  
$ tar xvfz my.tar.gz 또는 tar xvf my.tar.gz
```

- t와 x옵션에서 압축옵션을 지정하지 않으면 압축형식을 판단하여 사용

## 6.7 파일 탐색

- `find` : 특정 조건을 만족하는 파일을 찾음. (찾으면 `true` 반환)
  - 형식: `find` 탐색경로 탐색조건 [동작]
  - 조건: `-name` (파일이름 패턴), `-perm` (허가권:8진수)  
`-type` (파일유형), `-user` (소유자), `-group`(그룹)  
`-mtime` (최근에 수정), `-atime` (최근에 접근)
  - 동작: `-print` (파일명 출력,기본), `-ls` (`ls -l`수행), `-exec` (명령어수행)

```
$ find /etc -name 'y*.conf' 2> /dev/null    ... 파일이름 탐색. stderr제거
$ find /bin -perm 4755                      ... 허가권 탐색(4755, rwsr-xr-x)
$ find /bin -name 'p*' -perm 4755          ... 파일이름 및 허가권 탐색

$ find . -mtime 6                          ... 6일전 수정된 파일
$ find . -mtime +6                        ... 6일보다 오래전에 수정된 파일
$ find . -mtime -6                        ... 6일보다 최근에 수정된 파일
```

옵션	동작
-print	파일 이름을 출력 (다른 동작 옵션이 지정되지 않을 때 기본 설정)
-ls	파일의 자세한 속성을 출력
-exec <i>cmd</i> {} \;	파일 이름을 인수로 사용하여 명령어 <i>cmd</i> 를 실행. {}이 파일이름을 나타냄
-ok <i>cmd</i> {} \;	사용자 확인을 받아 명령어를 실행하는 것을 제외하면 -exec와 같음

## ■ 동작을 포함한 find 사용

\$ find . -mtime +6 -ls ... 조건에 맞는 파일 상세정보 출력

\$ find . -name '\*.bak' -exec rm {} \; ... 조건에 맞는 파일 삭제

\$ find . -name '\*.bak' -ls -exec rm {} \; ... 삭제할 파일 정보 출력 포함

# which, whereis, locate

---

## ■ which – 명령어 위치 탐색

- alias 설정과 PATH 경로에서 명령어 정보를 찾음

```
$ which ls
```

```
$ which cp
```

```
$ which type
```

## ■ whereis

- 미리 정해진 경로에서 명령어 실행파일, 소스, 매뉴얼 위치 찾음

```
$ whereis ls
```

```
$ whereis -b ls // binary file only
```

```
$ whereis -m ls // manual only
```

## ■ locate – 정해진 위치의 파일 탐색

- updatedb 명령어에 의해서 지정된 DB에서 찾음

```
$ locate '*/y*.conf'
```

## 6.8 기타 파일 유틸리티

---

- `basename` – 파일 경로 이름에서 base이름 추출
  - \$ `basename /usr/include/stdio.h` ... 베이스 이름 추출
  - \$ `basename /usr/include/stdio.h .h` ... 접미어.h 제거(일치하면)
- `dirname` – 파일 경로 이름에서 경로이름 추출(마지막 이름 제외)
  - \$ `dirname /usr/include/stdio.h` ... 디렉토리 이름
  - \$ `dirname /usr/include`

# od – 이진 파일 내용 보기

---

## ■ od (octal dump)

- 텍스트가 아닌 파일 내용을 8진수 등 여러 형식으로 출력
- 기본동작 : 8진수로 파일 내용을 출력

\$ od /bin/ls ... 16비트 단위, 8진수 출력

\$ od -N 64 /bin/ls ... 첫 64바이트 출력

\$ od -N 64 -c /bin/ls ... 문자 출력 (ELF 형식 실행파일)

## ■ 옵션

byte: -a(ASCII), -b(byte), -c(char),

two-byte: -d(u-dec), -i(s-dec), -x(hex), -o(oct)

일반적: -t type ... type은 출력형식과 크기 지정 (교과서 참조)

-t d2    -t d4    -tx1

---

■ strings – 출력가능 문자열 보기

\$ strings /bin/cat

\$ strings -n 10 /bin/cat ... 길이 10이상

\$ strings -n 10 -t o /bin/cat ... 8진수로 오프셋 주소 출력 (-t o)

(cf) \$ od -S 10 /bin/cat ... 길이 10이상의 문자열 출력

■ split – 큰 파일을 여러 개의 작은 파일로 분리

- 한번에 보조기억장치에 저장하거나 전송하기 어려울 때 사용
- 분리된 파일이름 : x로 시작, 두 개의 알파벳 문자 순서대로 추가
- 분리 크기 지정
  - 바이트수(-b) : K, M, G, T, P, E, Z, Y, KB(1000), MB(10<sup>6</sup>)
  - 줄수 (-l)

\$ split -b 1M coreutils-8.13.tar.gz ... 1M 바이트 단위로 분리 (-b 1M)

\$ cat x\* > combined ... 분리된 파일들을 결합

## 6.9 기타 유용한 명령어

### ■ xargs – 표준입력의 인수 전환

- 형식: `xargs [옵션] command` ... `command`가 없으면 `echo` 실행

`$ cat xdata | xargs` ... 표준입력을 인수로 사용하여 `echo` 실행

`$ echo `cat xdata`` ... 명령어 대치 이용. 앞과 같은 동작

`$ cat xdata | xargs -n 4` ... 한 번에 최대 4개의 인수를 사용 (-n 4)

`$ cat xdata | xargs -n 3 -t` ... `xargs`에서 실행하는 명령어 표시

`$ cat xdata | xargs echo 'Hello! '`

`$ cat xdata | xargs -d'\n' -n 1 echo 'Hello! '` ... 인수 구분자 '\n'

- `find`와 결합하여 처리하는 데 많이 사용됨

`$ find . -name '*.bak' | xargs file`

`$ find . -name '*.bak' -exec file {} \;` ... 앞과 같은 동작 (비효율적)

# su, time

---

- su – 사용자 대치(substitute user)

\$ su username

\$ su – username ... 로그인 초기화 과정 수행

\$ su ... root로 변경

- time – 실행 시간 측정

\$ time command ... command 실행시간 출력

\$ time find / -name '\*abc' 2> /dev/null ... abc로 끝나는 파일이름 탐색

# date – 다양한 형식으로 날짜 출력

## ■ date

- 형식: date [옵션] [+형식]
- 옵션: 교과서 p180-181 참조

\$ date +%m ... 월 출력  
\$ date +%x %r ... 날짜, 시간(빈칸 포함)  
\$ date +"오늘은 %Y년 %m월 %d일 입니다." ... 문자열 포함

- -d : 현재가 아닌 날짜를 출력

\$ date -d 12/25/2016 ... 날짜 지정  
\$ date -d "3 months 10 day ago" ... 3개월 이후의 10일 이전  
\$ date -d "3 months ago 10 day ago" ... 3개월 10일 이전  
\$ date -d "next mon" ... 미래 (다음 월요일)  
\$ date -d "last month" ... 과거 (지난 달)