

■ 리눅스에서 주로 사용하는 패키지

- deb: 데비안, 우분투 계열에서 사용하는 패키지
- RPM(Redhat Package Manager): 레드햇 계열 리눅스에서 주로 사용

■ 우분투 패키지의 특징

- 바이너리 파일로 구성되어 있어 컴파일이 필요 없다.
- 패키지의 파일이 관련 디렉터리에 바로 설치된다.
- 패키지를 삭제할 때 관련된 파일을 일괄적으로 삭제할 수 있다.
- 기존에 설치한 패키지를 삭제하지 않고 바로 업그레이드할 수 있다.
- 패키지의 설치 상태를 검증할 수 있다.
- 패키지에 대한 정보를 제공한다.
- 해당 패키지와 의존성을 가지고 있는 패키지가 무엇인지 알려준다. 따라서 의존성이 있는 패키지를 미리 설치할 수도 있고, apt-get 명령을 사용하면 의존성이 있는 패키지가 자동으로 설치된다.

■ 우분투 패키지의 카테고리

- 공식적으로 데비안 배포판에 포함된 모든 패키지는 데비안 자유 소프트웨어 지침에 따라 자유롭게 사용하고 배포할 수 있음
- 우분투도 네 개의 카테고리로 나누어 소프트웨어를 제공
 - main: 우분투에 의해 공식적으로 지원되며 자유롭게 배포할 수 있다.
 - restricted: 우분투에 의해 지원되나 완전한 자유 라이선스 소프트웨어는 아니다.
 - universe: 리눅스에서 사용할 수 있는 거의 대부분의 소프트웨어로 자유 소프트웨어일 수도 있고 아닐 수도 있으며, 기술적 지원을 보장하지 않는다.
 - multiverse: 자유 소프트웨어가 아닌 소프트웨어가 포함되어 있으며, 개인이 직접 라이선스를 확인해야 한다.

■ 우분투 패키지의 이름 구성

파일명_버전-리비전_아키텍처.deb

그림 9-2 우분투 패키지의 이름 구성

- 파일명: 첫 번째 항목은 패키지의 성격을 나타내는 파일명이다.
- 패키지 버전: 두 번째 항목은 패키지의 버전을 의미한다.
- 패키지 리비전: 리비전은 원래 소스의 버전이 업그레이드되지는 않았지만 패키지의 보안 문제나 의존성 변화, 스크립트의 변화 등이 있음을 의미한다.
- 아키텍처: 사용하는 시스템 아키텍처로 i386은 인텔을, all은 시스템과 상관없는 문서나 스크립트 등을 뜻한다.
- 확장자: 확장자는 .deb를 사용한다.

■ 우분투 패키지 저장소

- 우분투는 패키지와 패키지에 대한 정보를 저장하고 있는 서버인 패키지 저장소라는 개념을 사용
- 패키지 저장소에서는 패키지의 기능 추가나 보안 패치 등 지속적인 업그레이드를 집중적으로 관리
- 사용자는 저장소에 접속하여 최신 패키지를 내려받아 설치 가능
- 패키지 저장소에 대한 정보는 `/etc/apt/sources.list` 파일에 저장
 - 패키지 유형 : `deb`는 바이너리 패키지의 저장소를, `deb-src`는 패키지의 소스 저장소를 의미한다. 보통 한 저장소에 바이너리와 소스를 함께 저장
 - 저장소 주소 : `http` 프로토콜을 사용하는 URL 주소를 사용
 - 우분투 버전 정보 : 저장소에서 관리하는 패키지에 해당하는 우분투의 버전을 표시한다. 버전은 번호가 아니라 버전의 이름을 사용
 - 카테고리 : 저장소가 가지고 있는 소프트웨어 카테고리(`main`, `restricted` 등)를 표시

```
user1@myubuntu:~$ cat /etc/apt/sources.list
#deb cdrom:[Ubuntu 17.10 _Artful Aardvark_ - Release amd64 (20171018)]/ artful
main restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://kr.archive.ubuntu.com/ubuntu/ artful main restricted
# deb-src http://kr.archive.ubuntu.com/ubuntu/ artful main restricted
(생략)
```

02 우분투 패키지 설치

- APT 명령으로 패키지 관리하기
- apt-cache 명령 : APT 캐시(패키지 데이터베이스)에서 정보를 검색하여 출력

apt-cache

- **기능** APT 캐시에 질의하여 여러 가지 정보를 검색한다.
- **형식** apt-cache [옵션] 서브 명령
- **옵션** -f: 검색 결과로 패키지에 대한 전체 기록을 출력한다.
-h: 간단한 도움말을 출력한다.
- **서브 명령** stats: 캐시에 대한 통계 정보를 출력한다.
dump: 현재 설치되어 있는 패키지를 업그레이드한다.
search 키워드: 캐시에서 키워드를 검색한다.
showpkg 패키지명: 패키지에 대한 의존성 정보와 역의존성 정보를 검색하여 출력한다.
show 패키지명: 패키지에 대한 간단한 정보를 출력한다.
pkgnames: 사용 가능한 모든 패키지의 이름을 출력한다.
- **사용 예** apt-cache stats
apt-cache show vsftpd
apt-cache search vsftpd

■ apt-cache 명령

- 패키지 이름 검색하기 : search

```
user1@myubuntu:~$ apt-cache search vsftpd
resource-agents - Cluster Resource Agents
vsftpd - lightweight, efficient FTP server written for security
vsftpd-dbg - lightweight, efficient FTP server written for security (debug)
ccze - robust, modular log coloriser
ftpd - File Transfer Protocol (FTP) server
yasat - simple stupid audit tool
user1@myubuntu:~$
```

■ apt-cache 명령

- 패키지 정보 검색하기 : show
 - 버전, 패키지 크기, 카테고리, 체크섬 등 패키지에 관한 정보를 확인하려면 show 서브 명령을 사용

```
uuser1@myubuntu:~$ apt-cache show vsftpd
Package: vsftpd
Architecture: amd64
Version: 3.0.3-9
Priority: extra
Section: net
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Keng-Yu Lin <kengyu@lexical.tw>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 326
Provides: ftp-server
(생략)
```

02 우분투 패키지 설치

■ apt-get 명령

apt-get

- **기능** 패키지를 관리한다.
- **형식** apt-get [옵션] 서브 명령
- **옵션**
 - d: 패키지를 내려받기만 한다.
 - f: 의존성이 깨진 패키지를 수정하려고 시도한다.
 - h: 간단한 도움말을 출력한다.
- **서브 명령**
 - update: 패키지 저장소에서 새로운 패키지 정보를 가져온다.
 - upgrade: 현재 설치되어 있는 패키지를 업그레이드한다.
 - install 패키지명: 패키지를 설치한다.
 - remove 패키지명: 패키지를 삭제한다.
 - download 패키지명: 패키지를 현재 디렉터리로 내려받는다.
 - autoclean: 불완전하게 내려받았거나 오래된 패키지를 삭제한다.
 - clean: /var/cache/apt/archives에 캐시되어 있는 모든 패키지를 삭제하여 디스크 공간을 확보한다.
 - check: 의존성이 깨진 패키지를 확인한다.
- **사용 예** apt-get update apt-get install vsftpd apt-get clean

■ apt-get 명령

- 패키지 정보 업데이트하기 : update
 - /etc/apt/sources.list에 명시한 저장소에서 패키지 정보를 읽어 동기화
 - 새로운 패키지 정보를 가져와서 APT 캐시를 수정

```
user1@myubuntu:~$ sudo apt-get update
기준:1 http://kr.archive.ubuntu.com/ubuntu artful InRelease
기준:2 http://kr.archive.ubuntu.com/ubuntu artful-updates InRelease
기준:3 http://kr.archive.ubuntu.com/ubuntu artful-backports InRelease
받기:4 http://security.ubuntu.com/ubuntu artful-security InRelease [78.6 kB]
받기:5 http://security.ubuntu.com/ubuntu artful-security/main amd64 DEP-11
Metadata [204 B]
받기:6 http://security.ubuntu.com/ubuntu artful-security/universe amd64 DEP-11
Metadata [2,184 B]
내려받기 81.0 k바이트, 소요시간 2초 (38.2 k바이트/초)
패키지 목록을 읽는 중입니다... 완료
user1@myubuntu:~$
```


02 우분투 패키지 설치

■ apt-get 명령

- 패키지 업그레이드하기 : upgrade
 - 현재 설치되어 있는 모든 패키지 중에서 새로운 버전이 있는 패키지를 모두 업그레이드

```
user1@myubuntu:~$ sudo apt-get upgrade
```

패키지 목록을 읽는 중입니다... 완료

의존성 트리를 만드는 중입니다

상태 정보를 읽는 중입니다... 완료

업그레이드를 계산하는 중입니다... 완료

다음 패키지를 과거 버전으로 유지합니다:

ubuntu-desktop

다음 패키지를 업그레이드할 것입니다:

accountsservice apt apt-utils distro-info-data gdm3 gir1.2-accountsservice-1.0

gir1.2-gdm-1.0 gir1.2-gtk-3.0 gir1.2-mutter-1 gnome-control-center

gnome-control-center-data gnome-control-center-faces gnome-session-bin

(생략)

44개 업그레이드, 0개 새로 설치, 0개 제거 및 1개 업그레이드 안 함.

0 바이트/13.4 M바이트 아카이브를 받아야 합니다.

이 작업 후 37.9 k바이트의 디스크 공간이 비워집니다.

계속 하시겠습니까? [Y/n] n

중단.

```
user1@myubuntu:~$
```

■ apt-get 명령

- 특정 패키지 설치 또는 업그레이드하기 : install
 - 하나 이상의 패키지를 설치하거나 업그레이드할 때는 install 서브 명령을 사용

```
user1@myubuntu:~$ sudo apt-get install netcat
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다 :
  netcat-traditional
다음 새 패키지를 설치할 것입니다:
  netcat netcat-traditional
0개 업그레이드, 2개 새로 설치, 0개 제거 및 45개 업그레이드 안 함.
65.1 k바이트 아카이브를 받아야 합니다.
이 작업 후 157 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n]
(생략)
netcat-traditional (1.10-41.1) 설정하는 중입니다 ...
netcat (1.10-41.1) 설정하는 중입니다 ...
Processing triggers for man-db (2.7.6.1-2) ...
user1@myubuntu:~$
```

■ apt-get 명령

- 특정 패키지 설치 또는 업그레이드하기 : install
 - 여러 패키지를 한 번에 설치하려면 다음과 같이 패키지 이름을 나열

```
sudo apt-get install nethogs goaccess
```

- 패키지를 설치할 때 업그레이드를 하지 않으려면 '--no-upgrade' 옵션을 사용

```
sudo apt-get install netcat --no-upgrade
```

- 새로운 패키지를 설치하지 않고 업그레이드만 할 때는 '--only-upgrade' 옵션을 사용

```
sudo apt-get install netcat --only-upgrade
```

02 우분투 패키지 설치

■ apt-get 명령

- 패키지 삭제하기 : remove

```
user1@myubuntu:~$ sudo apt-get remove netcat
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
  netcat-traditional
Use 'sudo apt autoremove' to remove it.
다음 패키지를 지울 것입니다:
  netcat
0개 업그레이드, 0개 새로 설치, 1개 제거 및 45개 업그레이드 안 함.
이 작업 후 13.3 k바이트의 디스크 공간이 비워집니다.
계속 하시겠습니까? [Y/n]
(데이터베이스 읽는 중 ...현재 130505개의 파일과 디렉터리가 설치되어 있습니다.)
Removing netcat (1.10-41.1) ...
user1@myubuntu:~$
```

- 설정 파일을 포함하여 패키지를 삭제하려면 purge 서브 명령을 사용

```
sudo apt-get purge netcat
```

```
sudo apt-get remove --purge netcat
```

02 우분투 패키지 설치

■ apt-get 명령

- 패키지 자동 정리 및 삭제하기 : autoremove
 - 자동으로 설치되었으나 필요 없는 패키지는 autoremove 서브 명령으로 정리

```
user1@myubuntu:~$ sudo apt-get autoremove
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지를 지울 것입니다:
  netcat-traditional
0개 업그레이드, 0개 새로 설치, 1개 제거 및 45개 업그레이드 안 함.
이 작업 후 143 k바이트의 디스크 공간이 비워집니다.
계속 하시겠습니까? [Y/n]
(데이터베이스 읽는 중 ...현재 130502개의 파일과 디렉터리가 설치되어 있습니다.)
Removing netcat-traditional (1.10-41.1) ...
Processing triggers for man-db (2.7.6.1-2) ...
user1@myubuntu:~$
```

- 디스크 공간 정리하기 : clean
 - 검색했거나 내려받은 패키지 파일들을 삭제하고 디스크 공간을 정리

```
sudo apt-get clean
```

02 우분투 패키지 설치

■ apt-get 명령

- 패키지 내려받기 : download
 - 패키지를 설치하지 않고 내려받기만 하려면 download 서브 명령을 사용

```
user1@myubuntu:~$ sudo apt-get download netcat
받기:1 http://kr.archive.ubuntu.com/ubuntu artful/universe amd64 netcat all 1.10-41.1 [3,436 B]
내려받기 3,436 바이트, 소요시간 0초 (170 k바이트/초)
W: Download is performed unsandboxed as root as file '/home/user1/netcat_1.10-41.1_all.deb' couldn't be accessed by user '_apt'. - pkgAcquire::Run (13: 허가 거부)
user1@myubuntu:~$ ls net*
netcat_1.10-41.1_all.deb
user1@myubuntu:~$
```

- 패키지의 소스 관련 서브 명령 : source
 - 특정 패키지의 소스코드를 내려받기만 하는 경우

```
sudo apt-get --download-only source 패키지명
```

- 특정 패키지의 소스코드를 내려받고 압축을 푸는 경우

```
sudo apt-get source 패키지명
```

- 특정 패키지의 소스코드를 내려받아 압축을 풀고 컴파일하는 경우

```
sudo apt-get --compile source 패키지명
```

02 우분투 패키지 설치

■ dpkg 명령으로 패키지 관리하기

dpkg

centos rpm과 같은 기능

- **기능** 데비안의 패키지 관리 명령이다.
- **형식** dpkg [옵션] 파일명 또는 패키지명
- **옵션**
 - l: 설치된 패키지의 목록을 출력한다.
 - l 패키지명: 패키지의 설치 상태를 출력한다.
 - s 패키지명: 패키지의 상세 정보를 출력한다.
 - S 경로명: 경로명이 포함된 패키지를 검색한다.
 - l 패키지명: 패키지에서 설치된 파일의 목록을 출력한다.
 - c .deb 파일: 지정한 .deb 파일의 내용을 출력한다.
 - i .deb 파일: 해당 파일을 설치한다(sudo).
 - r 패키지명: 해당 패키지를 삭제한다(sudo).
 - P 패키지명: 해당 패키지와 설정 정보를 모두 삭제한다(sudo).
 - x .deb 파일 디렉터리: 해당 파일을 지정한 디렉터리에 풀어놓는다.
- **사용 예**
dpkg -l dpkg -s netcat dpkg -S /bin/ls
sudo dpkg -i netcat_1.10-40_all.deb

02 우분투 패키지 설치

■ dpkg 명령으로 패키지 관리하기

- 패키지 목록 보기 : -l 옵션
 - 출력 결과에서 첫 글자는 상단의 희망 상태를 나타내고, 두 번째 글자는 상태를 표시

```
user1@myubuntu:~$ dpkg -l
희망상태=알수없음(U)/설치(I)/지우기(R)/깨끗이(P)/고정(H)
! 상태=아님(N)/설치(I)/설정(C)/폴림(U)/절반설정(F)/일부설치(H)/트리거대기(W)/
! /      트리거밀림(T)
!/? 오류?=(없음)/다시설치필요(R) (상태, 오류가 대문자=불량)
!!/ 이름                버전                Architecture 설명
++=====
ii  accountsservic 0.6.42-0ubun amd64      query and manipulate user account
ii  acl             2.2.52-3buil amd64      Access control list utilities
ii  acpi-support   0.142        amd64      scripts for handling many ACPI eve
ii  acpid          1:2.0.28-1ub amd64      Advanced Configuration and Power I
ii  adduser        3.113+nmu3ub all        add and remove users and groups
(생략)
user1@myubuntu:~$
```


02 우분투 패키지 설치

■ dpkg 명령으로 패키지 관리하기

- 특정 파일이 포함된 패키지 검색하기 : -S 옵션

```
user1@myubuntu:~$ dpkg -S /bin/ls
coreutils: /bin/ls
user1@myubuntu:~$
```

- 패키지가 설치한 파일 목록 검색하기 : -L 옵션

```
user1@myubuntu:~$ dpkg -L zip
/.
/usr
/usr/bin
/usr/bin/zip
/usr/bin/zipcloak
/usr/bin/zipnote
/usr/bin/zipsplit
(생략)
user1@myubuntu:~$
```

■ dpkg 명령으로 패키지 관리하기

- 패키지의 파일 목록 검색하기 : -c 옵션

```
user1@myubuntu:~$ dpkg -c netcat_1.10-41.1_all.deb
drwxr-xr-x root/root      0  2017-04-14  03:31 ./
drwxr-xr-x root/root      0  2017-04-14  03:31 ./usr/
drwxr-xr-x root/root      0  2017-04-14  03:31 ./usr/share/
drwxr-xr-x root/root      0  2017-04-14  03:31 ./usr/share/doc/
drwxr-xr-x root/root      0  2017-04-14  03:31 ./usr/share/doc/netcat/
-rw-r--r-- root/root    1906  2017-04-14  03:31 ./usr/share/doc/netcat/changelog.
Debian.gz
-rw-r--r-- root/root      735  2011-02-11  14:21 ./usr/share/doc/netcat/copyright
user1@myubuntu:~$
```

■ dpkg 명령으로 패키지 관리하기

- 패키지 설치하기 : -i 옵션

- ① netcat_1.10-40_all.deb 패키지를 설치 시도

```
user1@myubuntu:~$ sudo dpkg -i netcat_1.10-41.1_all.deb
[sudo] user1의 암호:
Selecting previously unselected package netcat.
(데이터베이스 읽는 중 ...현재 130468개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack netcat_1.10-41.1_all.deb ...
Unpacking netcat (1.10-41.1) ...
dpkg: dependency problems prevent configuration of netcat:
 netcat 패키지는 다음 패키지에 의존: netcat-traditional (>= 1.10-39): 하지만:
 netcat-traditional 패키지는 설치하지 않았습니다.

dpkg: error processing package netcat (--install):
 의존성 문제 - 설정하지 않고 남겨둠
처리하는데 오류가 발생했습니다:
 netcat
user1@myubuntu:~$
```

- APT 명령과 달리 dpkg 명령은 의존성이 있는 패키지를 자동으로 설치하지 않으므로 사용자가 일일이 설치해야함
- 따라서 netcat-traditional 패키지를 내려받아 먼저 설치한 다음 netcat 패키지를 설치해야함

■ aptitude 명령으로 패키지 관리하기

- aptitude는 APT 명령처럼 패키지 관리를 자동화하여 쉽게 작업할 수 있도록 해줌
- 옵션이나 서브 명령 없이 실행할 경우 curses를 이용한 비주얼 모드로 동작

aptitude

- **기능** 우분투에서 패키지를 관리한다.
- **형식** aptitude [서브 명령]
- **서브 명령** 단독 실행: curses 프로그램이 나타난다.
 - search 키워드: 키워드를 검색하여 일치하는 패키지 목록을 출력한다.
 - update: 패키지 저장소를 업데이트한다.
 - upgrade: 모든 패키지를 최신 버전으로 업그레이드한다.
 - show 패키지명: 패키지에 대한 자세한 정보를 보여준다.
 - download 패키지명: 패키지를 내려받는다.
 - clean: 패키지 캐시 디렉터리에서 모든 패키지 파일을 삭제한다.
 - install: 패키지를 설치한다.
 - remove: 패키지를 삭제한다.
 - purge: 패키지와 설정 파일을 모두 삭제한다.

04 파일 아카이브와 압축

■ 파일 아카이브

- 파일을 묶어서 하나로 만든 것
- tar(tape archive) 명령은 원래 여러 파일이나 디렉터리를 묶어서 마그네틱 테이프와 같은 이동식 저장 장치에 보관하기 위해 사용하는 명령
- 현재는 다른 시스템과 파일을 주고받거나, 백업을 하기 위해 여러 파일이나 디렉터리를 하나의 아카이브 파일로 생성하거나, 기존 아카이브에서 파일을 추출하기 위해 사용

tar

- **기능** 파일과 디렉터리를 묶어 하나의 아카이브 파일을 생성한다.
- **형식** tar 기능[옵션] [아카이브 파일] 파일명
- **기능** c: 새로운 tar 파일을 생성한다.
t: tar 파일의 내용을 출력한다.
x: tar 파일에서 원본 파일을 추출한다.
r: 새로운 파일을 추가한다.
u: 수정된 파일을 업데이트한다.
- **옵션** f: 아카이브 파일이나 테이프 장치를 지정한다. 파일명을 '-'로 지정하면 tar 파일 대신 표준 입력에서 읽어들인다.
v: 처리하고 있는 파일의 정보를 출력한다.
h: 심벌릭 링크의 원본 파일을 포함한다.
p: 파일 복구 시 원래의 접근 권한을 유지한다.
j: bzip2로 압축하거나 해제한다.
z: gzip으로 압축하거나 해제한다.
- **사용 예** tar cvf unix.tar Unix tar xvf unix.tar

04 파일 아카이브와 압축

■ 파일 압축하기: gzip/gunzip - .gz 파일

gzip

- **기능** 파일을 압축한다.
- **형식** gzip [옵션] 파일명
- **옵션**
 - d: 파일 압축을 해제한다.
 - l: 압축 파일의 정보를 보여준다.
 - r: 하위 디렉터리를 이동하여 파일을 압축한다.
 - t: 압축 파일을 검사한다.
 - v: 압축 정보를 화면에 출력한다.
 - 9: 최대한 압축한다.
- **사용 예**
 - gzip a.txt
 - gzip -v b.txt c.txt

```
user1@myubuntu:~/linux_ex/ch9$ rm ch2.tar.gz
rm: 일반 파일 'ch2.tar.gz'를 제거할까요? y
user1@myubuntu:~/linux_ex/ch9$ gzip ch2.tar
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar.bz2  ch2.tar.gz  hosts
user1@myubuntu:~/linux_ex/ch9$
```

04 파일 아카이브와 압축

■ 압축 파일의 내용 보기 : zcat

zcat

- **기능** gzip으로 압축된 파일의 내용을 출력한다.
- **형식** zcat 파일명
- **사용 예** zcat abc.gz
zcat abc

```
user1@myubuntu:~/linux_ex/ch9$ zcat ch2.tar.gz | more
ch2/

0
r1
127.0.1.1      myubuntu

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
(생략)
```

■ 압축 풀기 : gunzip

gunzip

- **기능** gzip으로 압축된 파일의 압축을 푼다.
- **형식** gunzip 파일명
- **사용 예** gunzip abc.gz
gunzip abc

```
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar.bz2  ch2.tar.gz  hosts
user1@myubuntu:~/linux_ex/ch9$ gunzip ch2.tar.gz
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar  ch2.tar.bz2  hosts
user1@myubuntu:~/linux_ex/ch9$
```


■ bzip2/bunzip2 : .bz2 파일

bzip2

- **기능** 파일을 압축한다.
- **형식** bzip2 [옵션] 파일명
- **옵션**
 - d: 파일 압축을 해제한다.
 - l: 압축 파일의 정보를 보여준다.
 - t: 압축 파일을 검사한다.
 - v: 압축 정보를 화면에 출력한다.
 - best: 최대한 압축한다.
- **사용 예**
bzip2 abc.txt
bzip2 -v a.txt b.txt

```
user1@myubuntu:~/linux_ex/ch9$ rm ch2.tar.bz2
rm: 일반 파일 'ch2.tar.bz2'를 제거할까요? y
user1@myubuntu:~/linux_ex/ch9$ bzip2 ch2.tar
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar.bz2  hosts
user1@myubuntu:~/linux_ex/ch9$
```

04 파일 아카이브와 압축

■ 압축 파일의 내용 보기 : bzip2

bzcat

- **기능** bzip2로 압축된 파일의 내용을 출력한다.
- **형식** bzcat 파일명
- **사용 예** bzcat abc.bz2
bzcat abc

■ 압축 풀기 : bunzip2

bunzip2

- **기능** bzip2로 압축된 파일의 압축을 푼다.
- **형식** bunzip2 파일명
- **사용 예** bunzip2 1.c.bz2
bunzip2 1.c

```
user1@myubuntu:~/linux_ex/ch9$ bunzip2 ch2.tar.bz2
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar  hosts
user1@myubuntu:~/linux_ex/ch9$
```