우분투 리눅스

시스템 & 네트워크

Chapter 09. 소프트웨어 관리하기

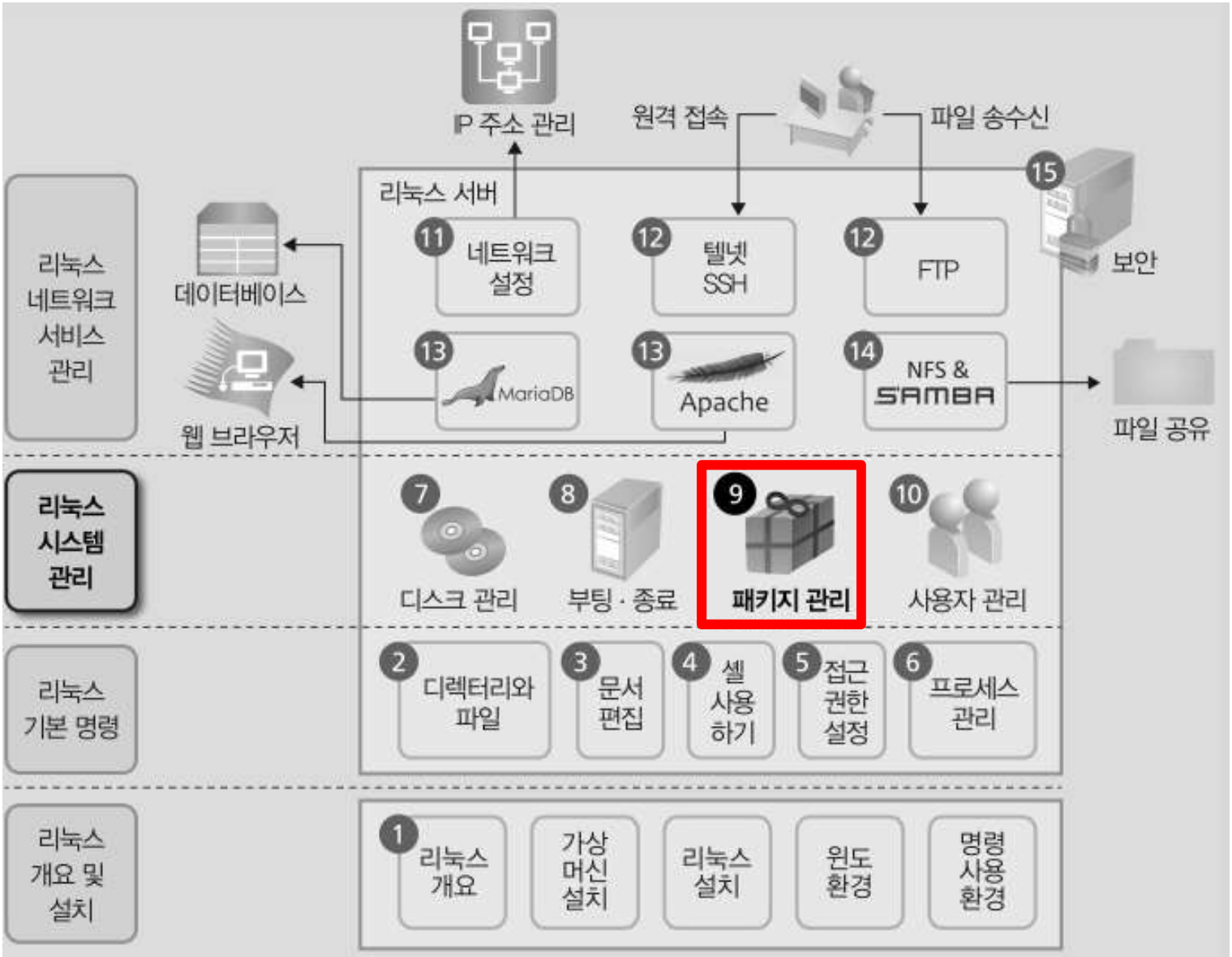
한빛아카데미 Hanbit Academy, Inc.

목차

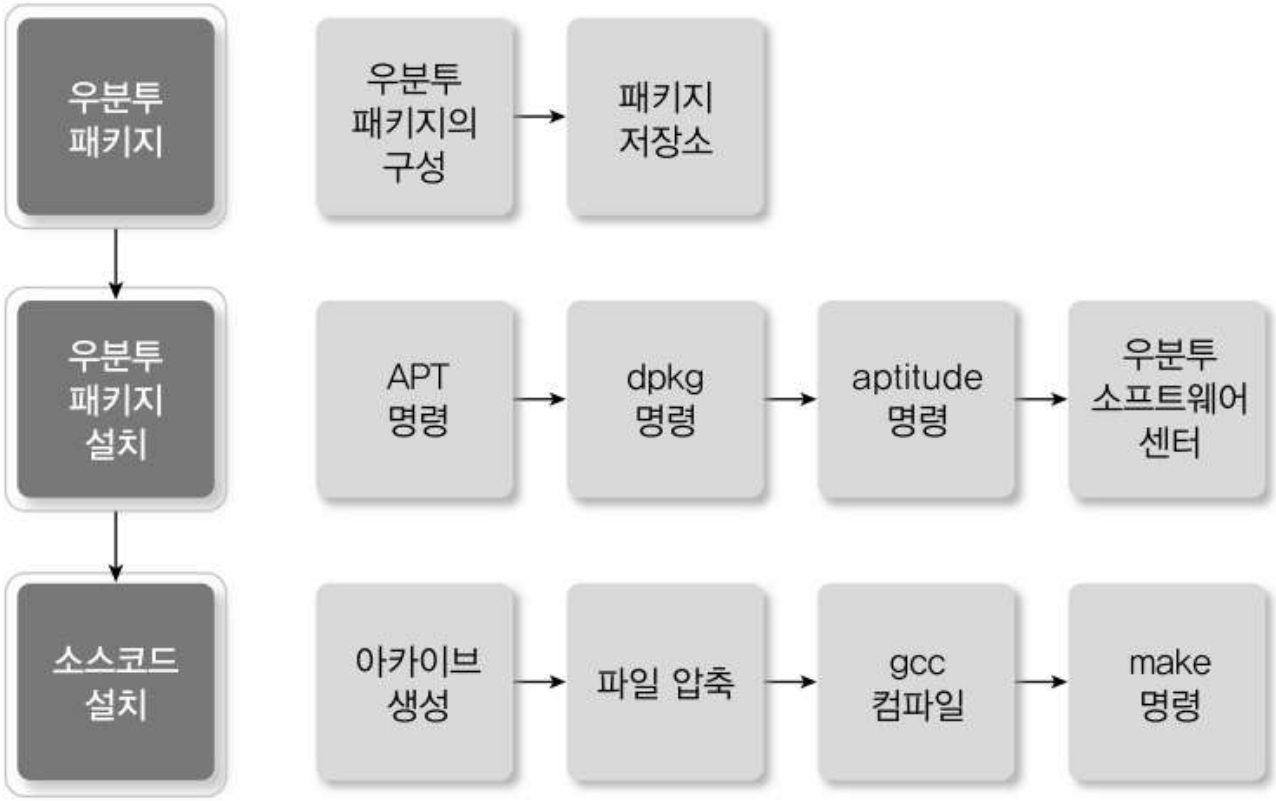
학습목표

- 우분투 패키지를 설치하고 업그레이드할 수 있다.
- APT 명령으로 패키지를 검색하고 상세 정보를 확인할 수 있다.
- dpkg 명령으로 패키지를 설치하고 업데이트하고 삭제할 수 있다.
- aptitude 명령으로 패키지를 관리할 수 있다.
- 우분투 소프트웨어 센터에서 프로그램을 확인하고 설치할 수 있다.
- tar 명령으로 아카이브를 생성하고, 내용을 확인하고 풀 수 있다.
- 파일을 압축하고 압축을 풀 수 있다.
- gcc로 C 파일을 컴파일할 수 있다.
- makefile을 작성하여 make 명령으로 실행 파일을 만들 수 있다.

리눅스 실습 스터디 맵


IP 주소 관리
원격 접속
파일 송수신
리눅스 서버
15
보안
리눅스 네트워크 서비스 관리
데이터베이스
11 네트워크 설정
12 텔넷 SSH
12 FTP
웹 브라우저
13 MariaDB
13 Apache
14 NFS & SAMBA
파일 공유
리눅스 시스템 관리
7 디스크 관리
8 부팅 · 종료
9 패키지 관리
10 사용자 관리
리눅스 기본 명령
2 디렉터리와 파일
3 문서 편집
4 셸 사용하기
5 접근 권한 설정
6 프로세스 관리
리눅스 개요 및 설치
1 리눅스 개요
가상 머신 설치
리눅스 설치
윈도 환경
명령 사용 환경

00 개요



[그림 9-1] 9장의 내용 구성

01 우분투 패키지의 개요

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 리눅스에서 주로 사용하는 패키지

- .deb: 데비안, 우분투 계열에서 사용하는 패키지
- RPM(Redhat Package Manager): 레드햇에서 만든 패키지 관리 도구

■ 우분투 패키지의 특징

- **바이너리 파일로 구성되어 있어 컴파일이 필요 없다.**
- 패키지의 파일들이 **관련 디렉터리로 바로 설치된다.**
- 한 번에 **설치된 패키지의 파일을 일괄적으로 삭제할 수 있다.**
- 기존에 설치된 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.

01 우분투 패키지의 개요

■ 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 표시
- 패키지 버전 : 두 번째 항목은 패키지의 버전을 의미
- 패키지 리비전 : 리비전은 원래 소스의 버전이 업그레이드되지는 않았지만 패키지의 보안 문제나 의존성 변화, 스크립트의 변화 등이 있음을 의미
- 아키텍처 : 사용하는 시스템 아키텍처로 i386은 인텔을, all은 시스템과 상관없는 문서나 스크립트 등을 의미
- 확장자 : 우분투 패키지의 확장자는 .deb를 사용

01 우분투 패키지의 개요

■ 우분투 패키지 저장소

- 우분투는 **패키지와 패키지에 대한 정보를 저장하고 있는 패키지 저장소**라는 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 : deb는 바이너리 패키지의 저장소를, deb-src는 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 13.10 _Saucy Salamander_ - Release i386 (20131016.1)]/
saucy 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/ saucy main restricted
deb-src http://kr.archive.ubuntu.com/ubuntu/ saucy main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://kr.archive.ubuntu.com/ubuntu/ saucy-updates main restricted
deb-src http://kr.archive.ubuntu.com/ubuntu/ saucy-updates 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 : 캐시의 통계 정보를 출력한다. dump : 현재 설치되어 있는 패키지를 업그레이드한다. search 키워드 : 캐시에서 키워드를 검색한다. showpkg 패키지명 : 패키지의 의존성 정보와 역의존성 정보를 검색하여 출력한다. show 패키지명 : 패키지의 간단한 정보를 출력한다. pkgnames : 사용 가능한 모든 패키지의 이름을 출력한다.
사용 예	apt-cache stats　　apt-cache show vsftpd　　apt-cache search vsftpd

02 우분투 패키지 설치

원리를 알면 IT가 맛있다
IT COOKBOOK

apt-cache 명령

- **APT 캐시 통계 정보 보기 : stats**
  - 전체 패키지 이름 : 패키지 이름의 전체 개수
  - 일반 패키지 : 일반적으로 사용하는 패키지의 개수
  - 순수 가상 패키지(pure virtual package)
    - 가상 패키지는 패키지의 이름만 제공하며 그 이름을 가진 별도의 패키지가 실제로 있는 것은 아님
  - 단일 가상 패키지(single virtual package)
    - 한 패키지가 특정 가상 패키지의 기능을 제공
  - 혼합 가상 패키지(mixed virtual package)
    - 특정 가상 패키지를 제공하거나 가상 패키지의 이름을 패키지 이름으로 사용하는 경우
  - 빠짐(missing) : 의존성은 있지만 어떠한 패키지도 제공하지 않는 패키지
  - 개별 버전 전체(total distinct version) : 캐시에 있는 패키지 버전의 개수를 의미

```
user1@myubuntu:~$ apt-cache stats
전체 패키지 이름 : 54542 (1,091 k)
전체 패키지 구조: 54542 (2,618 k)
  일반 패키지: 42089
  순수 가상 패키지: 550
  단일 가상 패키지: 4201
  혼합 가상 패키지: 862
  빠짐: 6840
개별 버전 전체: 45364 (2,903 k)
개별 설명 전체: 87878 (2,109 k)
전체 의존성: 267404 (7,487 k)
전체 버전/파일 관계: 47856 (766 k)
전체 설명/파일 관계: 87878 (1,406 k)
전체 제공 매핑: 8100 (162 k)
전체 패턴 문자열: 240 (2,731 )
전체 의존성 버전 용량: 1,244 k
전체 빈 용량: 85.2 k
차지하는 전체 용량: 14.0 M
user1@myubuntu:~$
```

02 우분투 패키지 설치

원리를 알면 IT가 맛있다
IT COOKBOOK

apt-cache 명령

- 사용 가능한 패키지 이름 보기 : pkgnames

```
user1@myubuntu:~$ apt-cache pkgnames
crm114
e3
filelight
fonts-moe-standard-kai
gstreamer1.0-fluendo-mp3
icewm-themes
kde-config-tablet
language-pack-fil-base
libauthen-simple-http-perl
libbio-ruby
libbiojava1.7-java
(생략)
```

02 우분투 패키지 설치

apt-cache 명령

- 패키지 이름 검색하기 : search

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

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

```
user1@myubuntu:~$ apt-cache show vsftpd
Package: vsftpd
Priority: extra
Section: net
Installed-Size: 363
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Daniel Baumann <daniel.baumann@progress-technologies.net>
Architecture: i386
Version: 3.0.2-1ubuntu2
(생략)
```

02 우분투 패키지 설치

apt-cache 명령

- 패키지 의존성 검색하기 : showpkg

```
user1@myubuntu:~$ apt-cache showpkg vsftpd
Package: vsftpd
Versions:
3.0.2-1ubuntu2 (/var/lib/apt/lists/kr.archive.ubuntu.com_ubuntu_dists_saucy_main
_binary-i386_Packages)
(생략)
Reverse Depends:
  ubumirror,vsftpd
  harden-servers,vsftpd
Dependencies:
3.0.2-1ubuntu2 - debconf (18 0.5) debconf-2.0 (0 (null)) upstart-job (0 (null))
libc6 (2 2.15) libcap2 (2 2.10) libpam0g (2 0.99.7.1) libssl1.0.0 (2 1.0.0)
libwrap0 (2 7.6-4~) adduser (0 (null)) libpam-modules (0 (null)) netbase (0 (null))
logrotate (0 (null)) ftp-server (0 (null)) ftp-server (0 (null))
Provides:
3.0.2-1ubuntu2 - ftp-server
Reverse Provides:
user1@myubuntu:~$
```

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 디렉터리에 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

02 우분투 패키지 설치

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
[sudo] password for user1:
무시http://kr.archive.ubuntu.com saucy InRelease
무시http://kr.archive.ubuntu.com saucy-updates InRelease
무시http://kr.archive.ubuntu.com saucy-backports InRelease
기존 http://kr.archive.ubuntu.com saucy Release.gpg
받기:1 http://kr.archive.ubuntu.com saucy-updates Release.gpg [933 B]
받기:2 http://kr.archive.ubuntu.com saucy-backports Release.gpg [933 B]
기존 http://kr.archive.ubuntu.com saucy Release
받기:3 http://kr.archive.ubuntu.com saucy-updates Release [49.6 kB]
(생략)
```

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
다음 패키지를 과거 버전으로 유지합니다:
  linux-generic linux-headers-generic linux-image-generic python3-distupgrade
  ubuntu-release-upgrader-core ubuntu-release-upgrader-gtk
다음 패키지를 업그레이드할 것입니다:
  account-plugin-aim account-plugin-jabber account-plugin-salut
  account-plugin-yahoo apparmor apport apport-gtk aptdaemon aptdaemon-data
  at-spi2-core avahi-autoipd avahi-daemon avahi-utils bind9-host cpp-4.8 cups
  cups-browsed cups-bsd cups-client cups-common cups-daemon cups-filters
(생략)
285개 업그레이드, 0개 새로 설치, 0개 제거 및 6개 업그레이드 안 함.
127 M바이트/211 M바이트 아카이브를 받아야 합니다.
이 작업 후 4,821 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]?
```

02 우분투 패키지 설치

원리를 알면 IT가 맛있다
IT COOKBOOK

apt-get 명령

- **특정 패키지 설치 또는 업그레이드하기 : install**
  - **하나 이상의 패키지를 설치하거나 업그레이드**할 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
다음 패키지를 더 설치할 것입니다:
  netcat-traditional
다음 새 패키지를 설치할 것입니다:
  netcat netcat-traditional
0개 업그레이드, 2개 새로 설치, 0개 제거 및 291개 업그레이드 안 함.
67.1 k바이트 아카이브를 받아야 합니다.
이 작업 후 186 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? y
(생략)
Selecting previously unselected package netcat.
netcat 패키지를 푸는 중입니다 (.../netcat_1.10-40_all.deb에서) ...
man-db에 대한 트리거를 처리하는 중입니다 ...
netcat-traditional (1.10-40) 설정하는 중입니다 ...
netcat (1.10-40) 설정하는 중입니다 ...
user1@myubuntu:~$
```

02 우분투 패키지 설치

apt-get 명령

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

```
user1@myubuntu:~$ sudo apt-get install nethogs goaccess
```

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

```
user1@myubuntu:~$ sudo apt-get install netcat --no-upgrade
```

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

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

02 우분투 패키지 설치

원리를 알면 IT가 맛있다
IT COOKBOOK

apt-get 명령

- **패키지 삭제하기 : remove**

```
user1@myubuntu:~$ sudo apt-get remove netcat
[sudo] password for user1:
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 'apt-get autoremove' to remove it.
다음 패키지를 지울 것입니다:
  netcat
0개 업그레이드, 0개 새로 설치, 1개 제거 및 291개 업그레이드 안 함.
이 작업 후 30.7 k바이트의 디스크 공간이 비워집니다.
계속 하시겠습니까 [Y/n]?y
(데이터베이스 읽는중 ...현재 168073개의 파일과 디렉터리가 설치되어 있습니다.)
netcat 패키지를 지우는 중입니다 ...
user1@myubuntu:~$
```

- **설정 파일을 포함하여 패키지를 삭제하려면 purge** 서브 명령을 사용

```
user1@myubuntu:~$ sudo apt-get purge netcat
```

```
user1@myubuntu:~$ 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 288개 업그레이드 안 함.
이 작업 후 156 k바이트의 디스크 공간이 비워집니다.
계속 하시겠습니까 [Y/n]? y
(데이터베이스 읽는중 ...현재 168070개의 파일과 디렉터리가 설치되어 있습니다.)
netcat-traditional 패키지를 지우는 중입니다 ...
man-db에 대한 트리거를 처리하는 중입니다 ...
user1@myubuntu:~$
```

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

```
user1@myubuntu:~$ 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 netcat 1.10-40 다운로드 중 [3,340 B]
내려받기 3,340 바이트, 소요시간 1초 (1,775 바이트/초)
user1@myubuntu:~$ ls net*
netcat_1.10-40_all.deb
user1@myubuntu:~$
```

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

```
user1@myubuntu:~$ sudo apt-get --download-only source 패키지명
```

  - 특정 패키지의 소스코드를 내려받고 압축을 푸는 경우

```
user1@myubuntu:~$ sudo apt-get source 패키지명
```

  - 특정 패키지의 소스코드를 내려받아 압축을 풀고 컴파일하는 경우

```
user1@myubuntu:~$ sudo apt-get --compile source 패키지명
```

03 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 명령[옵션] [아카이브 파일] 파일 이름
명령	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. 파일 이름을 -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 : gzip로 압축하거나 해제한다.
사용 예	tar cvf unix.tar Unix tar xvf unix.tar

03 파일 아카이브와 압축

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 아카이브 생성하기 : cvf

```
user1@myubuntu:~/linux_ex$ tar cvf ch2.tar ch2
ch2/
ch2/one/
ch2/one/tmp/
ch2/one/tmp/test/
ch2/data1.ln
ch2/temp/
ch2/temp/hosts
ch2/temp/services
ch2/temp/data1.cp
ch2/temp/text2
ch2/data
ch2/test
ch2/data1.sl
user1@myubuntu:~/linux_ex$ ls
ch2  ch2.tar  ch3  ch4  ch5  ch6
user1@myubuntu:~/linux_ex$
```

- tar 명령으로 파일을 묶어서 아카이브 파일을 만들어도 원본 파일은 그대로 있음

03 파일 아카이브와 압축

■ 아카이브 내용 확인하기 : tvf

```
user1@myubuntu:~/linux_ex$ tar tvf ch2.tar
drwxrwxr-x   user1/user1  0  2014-02-22 15:10 ch2/
drwxrwxr-x   user1/user1  0  2014-02-22 12:11 ch2/one/
drwxrwxr-x   user1/user1   0  2014-02-22 12:11 ch2/one/tmp/
drwxrwxr-x   user1/user1  0  2014-02-22 12:11 ch2/one/tmp/test/
-rw-r--r--   user1/user1      223  2014-02-22 12:36 ch2/data1.ln
drwxrwxr-x   user1/user1  0  2014-02-22 15:10 ch2/temp/
-rw-r--r--   user1/user1      223  2014-02-22 12:42 ch2/temp/hosts
-rw-r--r--   user1/user1   19436  2014-02-22 12:42 ch2/temp/services
-rw-r--r--   user1/user1      223  2014-02-22 14:54 ch2/temp/data1.cp
-rw-r--r--   user1/user1      223  2014-02-22 12:40 ch2/temp/text2
-rw-r--r--   user1/user1    19436  2014-02-22 14:56 ch2/data
-rw-rw-r--   user1/user1        0  2014-01-01 12:00 ch2/test
lrwxrwxrwx   user1/user1   0  2014-02-22 14:40 ch2/data1.sl -> data1
user1@myubuntu:~/linux_ex$
```

03 파일 아카이브와 압축

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 아카이브 풀기 : xvf

```
user1@myubuntu:~/linux_ex$ mkdir ch9
user1@myubuntu:~/linux_ex$ mv ch2.tar ch9
user1@myubuntu:~/linux_ex$ cd ch9
user1@myubuntu:~/linux_ex/ch9$ tar xvf ch2.tar
ch2/
ch2/one/
ch2/one/tmp/
ch2/one/tmp/test/
ch2/data1.ln
ch2/temp/
ch2/temp/hosts
ch2/temp/services
ch2/temp/data1.cp
ch2/temp/text2
ch2/data
ch2/test
ch2/data1.sl
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar
user1@myubuntu:~/linux_ex/ch9$
```

03 파일 아카이브와 압축

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 아카이브 업데이트하기 : uvf

- u 기능은 지정한 파일이 아카이브에 없는 파일이거나, 아카이브에 있는 파일이지만 수정된 파일일 경우 아카이브의 마지막에 추가 -> ch2/data 파일의 수정시간을 touch 명령으로 수정후 아카이브 업데이트

```
user1@myubuntu:~/linux_ex/ch9$ tar uvf ch2.tar ch2
user1@myubuntu:~/linux_ex/ch9$ touch ch2/data
user1@myubuntu:~/linux_ex/ch9$ tar uvf ch2.tar ch2
ch2/data
user1@myubuntu:~/linux_ex/ch9$ tar tvf ch2.tar
drwxrwxr-x   user1/user1         0  2014-02-22 15:10 ch2/
drwxrwxr-x   user1/user1         0  2014-02-22 12:11 ch2/one/
drwxrwxr-x   user1/user1         0  2014-02-22 12:11 ch2/one/tmp/
drwxrwxr-x   user1/user1         0  2014-02-22 12:11 ch2/one/tmp/test/
-rw-r--r--   user1/user1       223  2014-02-22 12:36 ch2/data1.ln
drwxrwxr-x   user1/user1         0  2014-02-22 15:10 ch2/temp/
-rw-r--r--   user1/user1       223  2014-02-22 12:42 ch2/temp/hosts
-rw-r--r--   user1/user1     19436  2014-02-22 12:42 ch2/temp/services
-rw-r--r--   user1/user1       223  2014-02-22 14:54 ch2/temp/data1.cp
-rw-r--r--   user1/user1       223  2014-02-22 12:40 ch2/temp/text2
-rw-r--r--   user1/user1     19436  2014-02-22 14:56 ch2/data
-rw-rw-r--   user1/user1         0  2014-01-01 12:00 ch2/test
lrwxrwxrwx   user1/user1         0  2014-02-22 14:40 ch2/data1.sl -> data1
-rw-r--r--   user1/user1     19436  2014-03-20 08:05 ch2/data
user1@myubuntu:~/linux_ex/ch9$
```

03 파일 아카이브와 압축

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 아카이브에 파일 추가하기 : rvf

- r 기능은 지정한 파일을 무조건 아카이브의 마지막에 추가

```
user1@myubuntu:~/linux_ex/ch9$ cp /etc/hosts .
user1@myubuntu:~/linux_ex/ch9$ tar rvf ch2.tar hosts
hosts
user1@myubuntu:~/linux_ex/ch9$ tar tvf ch2.tar
drwxrwxr-x  user1/user1     0  2014-02-22 15:10 ch2/
drwxrwxr-x  user1/user1     0  2014-02-22 12:11 ch2/one/
drwxrwxr-x  user1/user1     0  2014-02-22 12:11 ch2/one/tmp/
(생략)
-rw-r--r--   user1/user1  19436  2014-03-20 08:05 ch2/data
-rw-r--r--   user1/user1    223  2014-03-20 08:07 hosts
user1@myubuntu:~/linux_ex/ch9$
```

03 파일 아카이브와 압축

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 파일 압축과 아카이브

- 아카이브를 생성하면서 동시에 압축 수행
- 예: gzip으로 압축

```
user1@myubuntu:~/linux_ex/ch9$ tar cvzf ch2.tar.gz ch2
ch2/
ch2/one/
ch2/one/tmp/
ch2/one/tmp/test/
ch2/data1.ln
(생략)
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar  ch2.tar.gz  hosts
user1@myubuntu:~/linux_ex/ch9$
```

03 파일 아카이브와 압축

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 파일 압축과 아카이브

- 아카이브를 생성하면서 동시에 압축 실행
- 예: bzip2로 압축 실행: bzip2로 압축할 경우 j 옵션을 사용

```
user1@myubuntu:~/linux_ex/ch9$ tar cvjf ch2.tar.bz2 ch2
ch2/
ch2/one/
ch2/one/tmp/
ch2/one/tmp/test/
ch2/data1.ln
(생략)
user1@myubuntu:~/linux_ex/ch9$ ls
ch2  ch2.tar  ch2.tar.bz2  ch2.tar.gz  hosts
user1@myubuntu:~/linux_ex/ch9$
```

- 압축한 아카이브 파일의 내용은 tvf로 확인이 가능하며 xvf로 추출 가능

```
user1@myubuntu:~/linux_ex/ch9$ tar tvf ch2.tar.gz
drwxrwxr-x user1/user1       0 2014-02-22 15:10 ch2/
drwxrwxr-x user1/user1       0 2014-02-22 12:11 ch2/one/
drwxrwxr-x user1/user1       0 2014-02-22 12:11 ch2/one/tmp/
drwxrwxr-x user1/user1       0 2014-02-22 12:11 ch2/one/tmp/test/
-rw-r--r--   user1/user1     223 2014-02-22 12:36 ch2/data1.ln
(생략)
```

03 파일 아카이브와 압축

■ 파일 압축하기: gzip/gunzip - .gz 파일

gzip

기능 파일을 압축한다.

형식 gzip [옵션] 파일 이름

옵션
-d : 파일 압축을 해제한다.
-l : 압축된 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$ gzip -l ch2.tar.gz
         compressed        uncompressed  ratio uncompressed_name
               8648               71680     88.0%  ch2.tar
user1@myubuntu:~/linux_ex/ch9$
```

03 파일 아카이브와 압축

■ 압축 파일의 내용 보기 : zcat

zcat

기능 gz로 압축된 파일의 내용을 출력한다.

형식 zcat 파일 이름

사용 예 zcat abc.gz zcat abc

```
user1@myubuntu:~/linux_ex/ch9$ zcat ch2.tar.gz | more
ch2/
0000000
127.0.0.1       localhost
127.0.1.1       myubuntu
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
(생략)
```

03 파일 아카이브와 압축

원리를 알면 IT가 맛있다
IT COOKBOOK

■ 압축 풀기 : gunzip

gunzip

기능	gz로 압축된 파일의 압축을 푼다.
형식	gunzip 파일 이름
사용 예	gunzip abc.gz　　gunzip abc

```
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$
```

03 파일 아카이브와 압축

■ bzip2/bunzip2 : .bz2 파일

bzip2

기능	파일을 압축한다.
형식	bzip2 [옵션] 파일 이름
옵션	-d : 파일 압축을 해제한다. -l : 압축된 파일의 내용을 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$
```

03 파일 아카이브와 압축

■ 압축 파일의 내용 보기 : bzcat

bzcat

기능 압축된 파일의 내용을 출력한다.
형식 bzcat 파일 이름
사용 예 bzcat abc.bz2 bzcat abc

■ 압축 풀기 : bunzip2

bunzip2

기능 bzip2로 압축된 파일의 압축을 푼다.
형식 bunzip2 파일 이름
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$
```

04 소프트웨어 컴파일

C 컴파일러 설치하기

- C 언어로 작성한 프로그램을 컴파일하기 위해서는 C 컴파일러가 필요
- 리눅스에서 사용하는 C 컴파일러는 **GNU C 컴파일러로 패키지 이름이 gcc**
- gcc 설치

```
user1@myubuntu:~/linux_ex/ch9$ aptitude show gcc
Package: gcc
State: installed
Automatically installed: no
Version: 4:4.8.1-2ubuntu3
Priority: 옵션
Section: devel
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: i386
(생략)
```

```
$ apt-cache search gcc
$ apt-cache showpkg gcc

$ sudo apt-get install gcc
```

04 소프트웨어 컴파일

■ 간단한 C 프로그램 작성하기

```
user1@myubuntu:~/linux_ex/ch9$ nano hello.c
#include <stdio.h>
main() {
    printf("Hello, World.₩n");
}
```

■ C 프로그램 컴파일하기: 실행파일명은 a.out

```
user1@myubuntu:~/linux_ex/ch9$ gcc hello.c
user1@myubuntu:~/linux_ex/ch9$ ls
a.out  ch2  ch2.tar  hello.c  hosts
user1@myubuntu:~/linux_ex/ch9$
```

■ C 프로그램 실행하기: 경로지정 확인

```
user1@myubuntu:~/linux_ex/ch9$ a.out
a.out: command not found
user1@myubuntu:~/linux_ex/ch9$ ./a.out
Hello, World.
user1@myubuntu:~/linux_ex/ch9$
```

04 소프트웨어 컴파일

실행 파일명 변경하기

- gcc로 생성한 기본 실행 파일은 a.out
- 사용자가 원하는 이름으로 지정하려면 -o 옵션 사용

```
user1@myubuntu:~/linux_ex/ch9$ gcc -o hello hello.c
user1@myubuntu:~/linux_ex/ch9$ ./hello
Hello, World.
user1@myubuntu:~/linux_ex/ch9$
```

04 소프트웨어 컴파일

Creating an object file

- To create an object file from a source file, the compiler is invoked with the -c flag

```
user1@myubuntu:~/linux_ex/ch9$ gcc -c hello.c
```

- Then, to create an executable file from an object file

```
user1@myubuntu:~/linux_ex/ch9$ gcc -o hello hello.o
```

04 소프트웨어 컴파일

- 여러 소스 파일을 컴파일하고 링크하여 최종 실행 파일을 생성(과제2)

04 소프트웨어 컴파일

make 명령 사용하기

- make 명령은 makefile(또는 Makefile)에 설정된 정보를 읽어서 여러 소스 파일을 컴파일하고 링크하여 최종 실행 파일을 생성
- 소스파일 준비

```
user1@myubuntu:~/linux_ex/ch9$ nano one.c
#include <stdio.h>
extern int two();
main() {
     printf("Go to Module Two--₩n");
     two();
     printf("End of Module One.₩n");
}
```

- 컴파일 하면 오류 발생: two()가 무엇인지 모르겠다는 메시지

```
user1@myubuntu:~/linux_ex/ch9$ gcc one.c
/tmp/ccoaYv9w.o: In function 'main':
one.c:(.text+0x16): undefined reference to 'two'
collect2: error: ld returned 1 exit status
user1@myubuntu:~/linux_ex/ch9$
```

04 소프트웨어 컴파일

make 명령 사용하기

- 두 번째 파일 생성: two() 함수 정의

```
user1@myubuntu:~/linux_ex/ch9$ nano two.c
#include <stdio.h>
two() {
     printf("In Module Two--₩n");
     printf("--- This is a Moudule Two.₩n");
     printf("End of Module Two.₩n");
}
```

makefile 작성하기

```
user1@myubuntu:~/linux_ex/ch9$ nano makefile
TARGET=one
OBJECTS=one.o two.o
$TARGET : $OBJECTS
        gcc -o $TARGET $OBJECTS
one.o : one.c
        gcc -c one.c
two.o : two.c
        gcc -c two.c
```

04 소프트웨어 컴파일

make 파일 실행

```
user1@myubuntu:~/linux_ex/ch9$ make



user1@myubuntu:~/linux_ex/ch9$ ./one
Go to Module Two--
In Module Two--
--- This is a Moudule Two.
End of Module Two.
End of Module One.
user1@myubuntu:~/linux_ex/ch9$
```

우분투 리눅스

시스템 & 네트워크