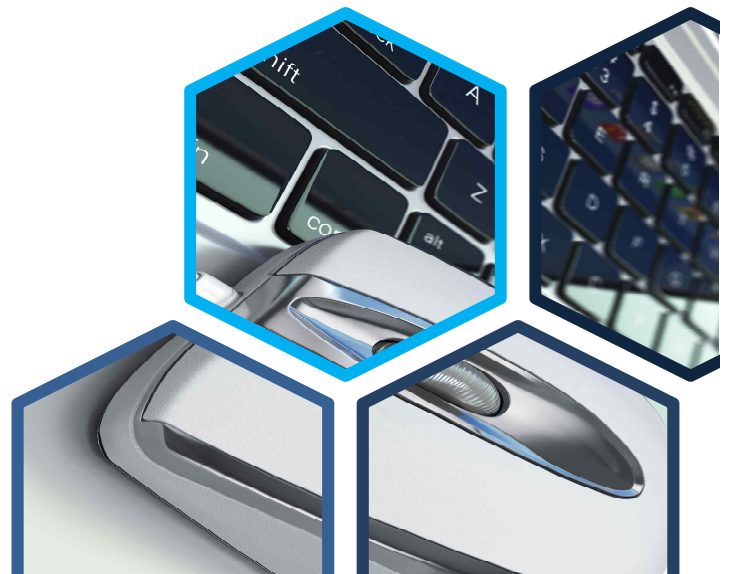


# 컴퓨터시스템

## 소프트웨어 관리





### 학습목표

- 우분투 패키지 설치방법을 설명할 수 있다.
- 스냅 패키지 설치방법을 설명할 수 있다.
- 파일 아카이브와 압축에 대해 설명할 수 있다.



### 학습내용

- 우분투 패키지 설치
- 스냅 패키지 설치
- 파일 아카이브와 압축



# 우분투 패키지 설치

## 1 패키지 개요



### 리눅스에서 주로 사용하는 패키지

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



### 우분투 패키지의 특징

- 바이너리 파일로 구성되어 있어 **컴파일이 필요 없음**
- 패키지의 파일이 관련 디렉터리에 바로 설치됨
- 패키지를 삭제할 때 관련된 파일을 **일괄적으로 삭제** 가능
- 기존에 설치한 패키지를 삭제하지 않고 **바로 업그레이드** 가능
- 패키지의 설치 상태 **검증** 가능
- 패키지에 대한 정보를 제공
- 해당 패키지와 **의존성을 가지고 있는 패키지**가 무엇인지 알려줌
  - ① 의존성이 있는 패키지를 미리 설치할 수 있음
  - ① **apt-get 명령**을 사용하면 의존성이 있는 패키지가 자동으로 설치됨



# 우분투 패키지 설치

## 1 패키지 개요



### 우분투 패키지의 카테고리

- 공식적으로 데비안 배포판에 포함된 모든 패키지는 **데비안 자유 소프트웨어 지침**에 따라 자유롭게 사용하고 배포할 수 있음
- 우분투도 네 개의 카테고리로 나누어 소프트웨어를 제공

#### main

우분투에 의해 공식적으로 지원되며 자유롭게 배포 가능

#### restricted

우분투에 의해 지원되나 완전한 자유 라이선스 소프트웨어는 아님

#### universe

리눅스에서 사용가능 한 대부분의 소프트웨어로 자유 소프트웨어일 수도 있고 아닐 수도 있으며, 기술적 지원을 보장하지 않음

#### multiverse

자유 소프트웨어가 아닌 소프트웨어가 포함되어 있으며, 개인이 직접 라이선스를 확인해야 함



# 우분투 패키지 설치

## 1 패키지 개요



### 우분투 패키지의 이름 구성

파일명\_버전-리비전\_아키텍처.deb

패키지의 성격을  
나타내는 파일명

확장자

패키지의 버전

원래 소스의 버전이 업그레이드 되지는  
않았지만 패키지의 보안 문제나 의존성 변화,  
스크립트의 변화 등이 있음을 의미함

사용하는 시스템  
아키텍처

- i386: 인텔
- all: 시스템과  
상관없는 문서나  
스크립트 등



# 우분투 패키지 설치

## 1 패키지 개요



### 우분투 패키지 저장소

- 우분투는 패키지와 패키지에 대한 정보를 저장하고 있는 서버인 **패키지 저장소**라는 개념을 사용함
- 패키지 저장소에서는 **패키지의 기능 추가나 보안 패치** 등 지속적인 업그레이드를 집중적으로 관리함
- 사용자는 저장소에 접속하여 **최신 패키지**를 내려 받아 설치할 수 있음
- 패키지 저장소에 대한 정보는 **/etc/apt/sources.list** 파일에 저장

패키지 유형	deb는 바이너리 패키지의 저장소를, deb-src는 패키지의 소스 저장소를 의미하며, 보통 한 저장소에 바이너리와 소스를 함께 저장함
저장소 주소	http 프로토콜을 사용하는 URL 주소를 사용
우분투 버전 정보	저장소에서 관리하는 패키지에 해당하는 우분투의 버전을 표시하는데, 버전은 번호가 아니라 버전의 이름을 사용함
카테고리	저장소가 가지고 있는 소프트웨어 카테고리(main, restricted 등)를 표시함



## 우분투 패키지 설치

## 2 우분투 패키지 설치



## apt-cache 명령

- 기능: APT 캐시(패키지 데이터베이스)에 질의하여 여러 가지 정보를 검색하여 출력함
- 형식: apt-cache [옵션] 서브 명령
- 옵션

-f	검색 결과로 패키지에 대한 전체 기록을 출력함
-h	간단한 도움말을 출력함

- 서브명령

stats	캐시에 대한 통계정보를 출력함
dump	현재 설치되어 있는 패키지를 업그레이드함
search 키워드	캐시에서 키워드를 검색함
showpkg 패키지명	패키지에 대한 의존성 정보와 역의존성 정보를 검색하여 출력함
show 패키지명	패키지에 대한 간단한 정보를 출력함
pkgnames	사용 가능한 모든 패키지의 이름을 출력함

- 사용 예

```
apt-cache stats
apt-cache show vsftpd
apt-cache search vsftpd
```





## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### apt-cache 명령

APT 캐시 통계 정보 보기: **stats**

전체 패키지	일반 패키지
패키지 이름의 전체 개수	일반적으로 사용하는 패키지의 개수

순수 가상 패키지 (Pure virtual package)	단일 가상 패키지 (Single virtual package)	혼합 가상 패키지 (Mixed virtual package)
가상 패키지는 패키지의 이름만 제공하며 그 이름을 가진 별도의 패키지가 실제로 있는 것은 아님	한 패키지가 특정 가상 패키지의 기능을 제공	특정 가상 패키지를 제공하거나 가상 패키지의 이름을 패키지 이름으로 사용하는 경우

- ❶ **빠짐(missing)**: 의존성은 있지만 어떠한 패키지도 제공하지 않는 패키지
- ❷ **개별 버전 전체(total distinct version)**: 캐시에 있는 패키지 버전의 개수를 의미





## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### apt-cache 명령

- ☁ 사용 가능한 패키지 이름 보기: `pkgnames`
- ☁ 패키지 이름 검색하기: `search`
- ☁ 패키지 정보 검색하기: `show`
  - 🕒 버전, 패키지 크기, 카테고리, 체크섬 등 패키지에 관한 정보를 확인하려면 `show` 서브 명령을 사용
- ☁ 패키지 의존성 검색하기: `showpkg`



## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### apt-get 명령

- 기능: 패키지를 관리함
- 형식: apt-get [옵션] 서브 명령
- 옵션

-d	패키지를 내려 받기만 함
-f	의존성이 깨진 패키지를 수정하려고 시도함
-h	간단한 도움말을 출력함

#### 서브명령

update	패키지 저장소에서 새로운 패키지 정보를 가져옴
upgrade	현재 설치되어 있는 패키지를 업그레이드함
install 패키지명	패키지를 설치함
remove 패키지명	패키지를 삭제함
download 패키지명	패키지를 현재 디렉터리로 내려 받음
autoclean	불완전하게 내려 받았거나 오래된 패키지를 삭제함
clean	/var/cache/apt/archives에 캐시되어 있는 모든 패키지를 삭제하여 디스크 공간을 확보함
check	의존성이 깨진 패키지를 확인함



## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### apt-get 명령

##### ☁ 사용 예

```
apt-get update  
apt-get install vsftpd  
apt-get clean
```

##### ☁ 패키지 정보 업데이트하기: **update**

- ① /etc/apt/sources.list에 명시한 저장소에서 패키지 정보를 읽어 **동기화**
- ① 새로운 패키지 정보를 가져와서 APT 캐시를 **수정**

##### ☁ 패키지 업그레이드하기: **upgrade**

- ① 현재 설치되어 있는 모든 패키지 중에서 새로운 버전이 있는 패키지를 모두 **업그레이드**

##### ☁ 특정 패키지 설치 또는 업그레이드하기: **install**

- ① 하나 이상의 패키지를 설치하거나 업그레이드할 때는 **install 서브 명령**을 사용
- ① 여러 패키지를 한 번에 설치하려면 다음과 같이 **패키지 이름을 나열**

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

- ① 패키지를 설치할 때 업그레이드를 하지 않으려면 '**--no-upgrade**' 옵션을 사용

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

- ① 새로운 패키지를 설치하지 않고 업그레이드만 할 때는 '**--only-upgrade**' 옵션 사용

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



## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### apt-get 명령

- 패키지 삭제하기: **remove**

- 설정 파일을 포함하여 패키지를 삭제하려면 **purge** 서브 명령을 함께 사용함

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

- purge** 서브 명령과 **--purge** 옵션을 사용함

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

- 패키지 자동 정리 및 삭제하기: **autoremove**

- 자동으로 설치되었으나 필요 없는 패키지는 **autoremove** 서브 명령으로 정리

- 디스크 공간 정리하기: **clean**

- 검색했거나 내려 받은 패키지 파일들을 삭제하고 디스크 공간을 정리

```
sudo apt-get clean
```

- 패키지 내려 받기: **download**

- 패키지를 설치하지 않고 내려 받기만 하려면 **download** 서브 명령을 사용

```
user@mybuntu:~$ sudo apt-get download netcat
```



## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### apt-get 명령

- ☁ 패키지의 소스 관련 서브 명령: **source**

- ① 특정 패키지의 소스코드를 내려 받기만 하는 경우

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

- ① 특정 패키지의 소스코드를 내려 받고 압축을 푸는 경우

```
sudo apt-get source 패키지명
```

- ① 특정 패키지의 소스코드를 내려 받아 압축을 풀고 컴파일하는 경우

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



## 우분투 패키지 설치

## 2 우분투 패키지 설치



## dpkg 명령

- 기능: 데비안의 패키지 관리 명령임
- 형식: dpkg [옵션] 파일명 또는 패키지명
- 옵션

-l	설치된 패키지의 목록을 출력함
-l 패키지명	패키지의 설치 상태를 출력함
-s 패키지명	패키지의 상세 정보를 출력함
-S 경로명	경로명이 포함된 패키지를 검색함
-L 패키지명	패키지에서 설치된 파일의 목록을 출력함
-c .deb 파일	지정한 deb 파일의 내용을 출력함
-i .deb 파일	해당 파일을 설치함(sudo)
-r 패키지명	해당 패키지를 삭제함(sudo)
-P 패키지명	해당 패키지와 설정 정보를 모두 삭제함(sudo)

- 사용 예

```
dpkg -l
dpkg -s netcat
dpkg -S /bin/lis
sudo dpkg -i netcat_1.10-40_all.deb
```



## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### dpkg 명령

☁ 패키지 목록 보기: -i 옵션

- ❶ 출력 결과에서 첫 글자는 **상단의 희망 상태**를 나타내고 두 번째 글자는 **상태**를 표시

```
user1@myubuntu:~$ dpkg -i  
희망상태=알수없음(U)/설치(I)/지우기(R)/깨끗이(P)/고정(H)  
|  
상태=아님(N)/설치(I)/설정(C)/풀림(U)/절반설정(F)/일부설치(H)/트리거대기(W)
```

- ❷ -i 다음에 특정 패키지의 이름을 지정하면 **해당 패키지에 관한 정보만 출력**





## 우분투 패키지 설치

### 2 우분투 패키지 설치



#### dpkg 명령

- 패키지 상세 정보 보기: **-s 옵션**

```
user1@myubuntu:~$ dpkg -s zip
```

- 특정 파일이 포함된 패키지 검색하기: **-S 옵션**
- 패키지가 설치한 파일 목록 검색하기: **-L 옵션**
- 패키지의 파일 목록 검색하기: **-c 옵션**
- 패키지 설치하기: **-i 옵션**
  - APT 명령과 달리 dpkg 명령은 의존성이 있는 패키지를 자동으로 설치하지 않으므로 **사용자가 일일이 설치**해야 함
    - netcat-traditional 패키지를 내려 받아 먼저 설치한 다음 netcat 패키지를 설치해야 함
- 패키지 삭제하기: **-r, -P 옵션**

-r	설치된 패키지만 삭제
-P	패키지와 설정 정보를 모두 삭제



### 우분투 패키지 설치

## 2 우분투 패키지 설치



### aptitude 명령

- aptitude는 APT 명령처럼 패키지 관리를 **자동화**하여 쉽게 작업할 수 있도록 해줌
- 옵션이나 서브 명령 없이 실행할 경우 **curses**를 이용한 **비주얼 모드**로 동작
- 기능: 우분투에서 패키지를 관리함
- 형식: aptitude [서브 명령]
- 서브 명령 단독 실행: curses 프로그램이 나타남

search 키워드	키워드를 검색하여 일치하는 패키지 목록을 출력함
update	패키지 저장소를 업데이트함
upgrade	모든 패키지를 최신 버전으로 업그레이드함
show 패키지명	패키지에 대한 자세한 정보를 보여줌
download 패키지명	패키지를 내려 받음
clean	패키지 캐시 디렉터리에서 모든 패키지 파일을 삭제함
install	패키지를 설치함
remove	패키지를 삭제함
purge	패키지와 설정 파일을 모두 삭제함



우분투 패키지 설치 실습 영상은  
학습 콘텐츠에서 확인하실 수 있습니다.



# 스냅 패키지 설치

## 1 패키지 개요



### 스냅의 개념

- ☁️ 우분투가 새로 도입한 패키지 형식으로 **샌드박스 형태의 패키지**
- ☁️ 패키지를 만들 때 프로그램이 사용하는 모든 라이브러리를 패키지 안에 포함

#### 장점

- 🔒 개발자가 다른 패키지나 라이브러리와 의존성을 신경 쓰지 않아도 됨
- 🔒 기존 시스템과 격리되어 실행하는 샌드박스 형식이므로 보안이 강화됨

#### 단점

- 🔒 패키지의 용량이 커짐

- ☁️ 스냅은 이제 시작 단계로 어떻게 발전할지 관심을 가지고 지켜볼 필요가 있음



### 스냅 사용하기

- ☁️ **apt-get 명령**으로 snap을 설치하고 시작

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



## 스냅 패키지 설치

### 2 스냅 명령



#### snap 명령의 기본 형식

- 기능: 스냅 패키지를 설치하고 설정하고 삭제함
- 형식: snap [옵션] 명령
- 옵션

-h	도움말을 출력함
----	----------

- 명령

disable	스냅 서비스와 실행 파일의 사용을 중지함
download 스냅명	지정한 스냅 패키지를 내려 받음
enable	스냅 서비스와 실행 파일의 사용을 시작함
find 스냅명	지정한 스냅을 검색함
info 스냅명	지정한 스냅의 상세 정보를 출력함
install 스냅명	지정한 스냅을 설치함
list	설치한 스냅의 목록을 출력함
remove 스냅명	지정한 스냅을 삭제함

- 사용 예

```

snap list
snap install hello-world

```



## 스냅 패키지 설치

### 2 스냅 명령



스냅 목록 출력하기: **list** 명령



스냅 찾기: **find** 명령

- ☁ 앞의 예에서 출력된 'hello-world' 스냅 찾기

```
user1@myubunty:~$ snap find hello-world
```



스냅 설치하기: **install** 명령

- ☁ **find**로 검색한 hello-world 스냅을 설치

```
user1@mybuntu:~$ sudo snap install hello-world
```

- ☁ hello-world 스냅이 설치한 명령을 실행해보면 'hello world!'가 출력

```
user1@myubuntu:~$ hello-world  
Hello world
```



## 스냅 패키지 설치

### 2 스냅 명령



#### 스냅의 상세 정보 확인하기: **info** 명령

- 설치된 스냅의 상세 정보를 확인하는 명령
- info 명령을 실행 시 다음 정보들이 출력됨
  - 스냅의 이름과 요약 정보
  - 제공자
  - 연락처
  - 간단한 설명
  - 관련 명령
  - 설치된 버전 정보



#### 스냅의 삭제하기: **remove** 명령

- hello-world 스냅을 삭제하기

```
user1@myubuntu:~$ sudo snap remove hello-world  
Hello-world removed
```



## 파일 아카이브와 압축

### 1 파일 아카이브 개요



#### 파일 아카이브: tar

- 파일을 묶어서 하나로 만든 것
- tar(tape archive) 명령은 원래 여러 파일이나 디렉터리를 묶어서 **마그네틱 테이프와 같은 이동식 저장 장치에 보관하기 위해 사용**
- 파일 아카이브의 사용
  - ① 다른 시스템과 **파일을 주고 받음**
  - ① 백업을 하기 위해 여러 파일이나 디렉터리를 **하나의 아카이브 파일로 생성**
  - ① 기존 아카이브에서 **파일 추출**





## 파일 아카이브와 압축

## 1 파일 아카이브 개요



## 파일 아카이브: tar

- 기능: 파일과 디렉터리를 묶어 하나의 아카이브 파일을 생성함
- 형식: tar 기능[옵션] [아카이브 파일] [파일명]
- 기능

c	새로운 tar 파일을 생성함
t	tar 파일의 내용을 출력함
x	tar 파일에서 원본 파일을 추출함
r	새로운 파일을 추가함
u	수정한 파일을 업데이트함

- 옵션

f	<ul style="list-style-type: none"> <li>아카이브 파일이나 테이프 장치를 지정함</li> <li>파일명을 '-'로 지정하면 tar 파일 대신 표준 입력에서 읽어 들임</li> </ul>
v	<ul style="list-style-type: none"> <li>처리하고 있는 파일의 정보를 출력함</li> </ul>
h	<ul style="list-style-type: none"> <li>심벌릭 링크의 원본 파일을 포함함</li> </ul>
p	<ul style="list-style-type: none"> <li>파일 복구 시 원래의 접근 권한을 유지함</li> </ul>
j	<ul style="list-style-type: none"> <li>bzip2로 압축하거나 해제함</li> </ul>
z	<ul style="list-style-type: none"> <li>gzip으로 압축하거나 해제함</li> </ul>

- 사용 예

```
tar cvf unix.tar unix
tar xvf unix.tar
```



## 파일 아카이브와 압축

### 2 파일 아카이브 생성



#### 아카이브 생성하기: **cvf**

```
[user1@localhost linux_ex]$ tar cvf ch2.tar ch2
```

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



#### 아카이브 내용 확인하기: **tvf**

```
user1@myubuntu:~/linux_ex$ tar tvf ch2.tar
```



#### 아카이브 풀기: **xvf**

```
user1@myubuntu:~/linux_ex/ch9$ tar xvf ch2.tar
```



#### 아카이브 업데이트하기: **uvf**

- u 기능은 지정한 파일이 아카이브에 없는 파일이거나 아카이브에 있는 파일이지만 **수정된 파일일 경우 아카이브의 마지막에 추가**



#### 아카이브에 파일 추가: **rvf**

- 지정한 파일을 무조건 아카이브의 마지막에 추가



## 파일 아카이브와 압축

### 2 파일 아카이브 생성



#### 아카이브 생성하고 압축하기

- 아카이브를 생성하면서 동시에 압축 수행
- 예: gzip로 압축할 경우 **z 옵션**을 사용

```
[user1@localhost ch9]$ tar czvf ch2.tar.gz ch2
```

- 예: bzip2로 압축 실행: bzip2로 압축할 경우 **j 옵션**을 사용

```
[user1@localhost ch9]$ tar cvjf ch2.tar.bz2 ch2
```



## 파일 아카이브와 압축

### 3 파일 압축과 아카이브



#### 파일 압축하기: **gzip/gunzip - .gz 파일**

- 기능: 파일을 압축함
- 형식: gzip [옵션] [파일명]
- 옵션

-d	파일 압축을 해제함
-l	압축 파일의 정보를 보여줌
-r	하위 디렉터리를 탐색하여 압축함
-t	압축 파일을 검사함
-v	압축 정보를 화면에 출력함
-g	최대한 압축함

- 사용 예

```
gzip a.txt  
gzip -v b.txt c.txt
```



## 파일 아카이브와 압축

### 3 파일 압축과 아카이브



#### 압축 파일의 내용 보기: **zcat**

- 기능: gzip으로 압축된 파일의 내용을 출력함
- 형식: `zcat [파일명]`
- 사용 예

```
zcat abc.gz
zcat abc
```



#### bzip2/bunzip2: **.bz2** 파일

- 기능: 파일을 압축함
- 형식: `bzip2 [옵션] [파일명]`
- 옵션

<code>-d</code>	파일 압축을 해제함
<code>-l</code>	압축 파일의 정보를 보여줌
<code>-t</code>	압축 파일을 검사함
<code>-v</code>	압축 정보를 화면에 출력함
<code>--best</code>	최대한 압축함

- 사용 예

```
bzip2 abc.txt
bzip2 -v a.txt b.txt
```



## 파일 아카이브와 압축

### 3 파일 압축과 아카이브



#### 압축 파일의 내용 보기: **bzcat**

- 기능: bzip2로 압축된 파일의 내용을 출력함
- 형식: bzcat [파일명]
- 사용 예

```
bzcat abc.bz2  
bzcat abc
```



#### 압축 풀기: **bunzip2**

- 기능: bzip2로 압축된 파일의 압축을 풀
- 형식: bunzip2 [파일명]
- 사용 예

```
bunzip2 1.c.bz2  
bunzip2 1.c
```






파일 아카이브와 압축 실습 영상은  
학습 콘텐츠에서 확인하실 수 있습니다.





### 핵심요약

#### 1 우분투 패키지 설치

-  우분투는 패키지와 패키지에 대한 정보를 저장하고 있는 서버인 패키지 저장소라는 개념을 사용함
-  패키지 저장소에서는 패키지의 기능 추가나 보안 패치 등 지속적인 업그레이드를 집중적으로 관리함
-  우분투 패키지 설치 명령: apt-cache 명령, apt-get 명령, dpkg 명령, aptitude 명령

#### 2 스냅 패키지 설치

-  우분투가 새로 도입한 패키지 형식으로 샌드박스 형태의 패키지임
-  패키지를 만들 때 프로그램이 사용하는 모든 라이브러리를 패키지 안에 포함함





### 핵심요약

#### 3 파일 아카이브와 압축

- 📁 파일을 묶어서 하나로 만든 것임
- 📁 tar(tape archive) 명령은 원래 여러 파일이나 디렉터리를 묶어서 마그네틱 테이프와 같은 이동식 저장 장치에 보관하기 위해 사용함