

SW Management

Package

❖ 우분투 패키지

✓ 개요

- 리눅스에서 소프트웨어는 소스 코드 형식 또는 바로 설치하여 사용할 수 있는 패키지 형태로 배포
- 소스 코드로 배포하는 경우 대부분 하나의 아카이브 파일로 묶은 후 압축하여 배포
- 바이너리 패키지로 배포하는 경우 리눅스에서 주로 사용하는 패키지 형식은 RPM과 deb 두 가지인데 우분투는 기본적으로 deb 형식의 패키지를 사용하고 RPM 형식은 redhat 계열 리눅스에서 주로 사용
- 우분투에서도 RPM 형식의 패키지를 설치할 수는 있으나 별도의 명령을 설치해야 하며 특별한 경우가 아니면 사용하지 않음
- 우분투는 16.04 부터 새로운 패키지 형식으로 스냅을 도입
- 스냅 패키지는 기존 패키지 형식의 의존성 문제를 해결한 것으로 deb 패키지와 호환성이 있음

Package

❖ 우분투 패키지

✓ 우분투 패키지의 특징

- 바이너리 파일로 구성되어 있어 컴파일이 필요 없음
- 패키지 내의 파일이 관련 디렉토리에 바로 설치됨
- 패키지를 삭제할 때 관련된 파일을 일괄적으로 삭제할 수 있음
- 기존에 설치한 패키지를 삭제하지 않고 바로 업그레이드 할 수 있음
- 패키지의 설치 상태를 검증할 수 있음
- 패키지에 대한 정보를 제공
- 해당 패키지가 의존하는 패키지가 무엇인지 알려주기 의존성이 있는 패키지를 미리 설치할 수도 있고 apt-get 명령을 사용하면 의존성이 있는 패키지가 자동으로 설치됨

✓ 우분투 패키지의 카테고리

- main: 우분투에 의해 공식적으로 지원되며 자유롭게 배포할 수 있음
- restricted: 우분투에 의해 지원되나 완전한 자유 라이선스 소프트웨어는 아님
- universe: 리눅스에서 사용할 수 있는 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

Package

❖ 우분투 패키지

✓ 우분투 패키지의 이름

패키지명_버전_데비안 리비전ubuntu우분투 리비전_아키텍처.deb

①

②

③

④

⑤

⑥

- ① 패키지의 이름
- ② 오리지널 패키지의 버전
- ③ 오리지널 패키지 버전을 데비안에 적용한 리비전 번호
- ④ 오리지널 또는 데비안 패키지를 우분투에 적용한 리비전 번호
- ⑤ 사용하는 시스템 아키텍처로 all은 시스템의 종류와 상관없이 사용할 수 있는 것이고 amd64는 64비트 운영체제임을 의미
- ⑥ 확장자로 .deb를 사용

Package

❖ 우분투 패키지 관리

✓ 패키지 저장소

- 우분투는 패키지와 패키지에 대한 정보를 저장하고 있는 서버인 패키지 저장소 라는 개념을 사용
- 패키지 저장소에서는 패키지의 기능 추가나 보안 패치 등 지속적인 업그레이드를 집중적으로 관리하고 사용자는 저장소에 접속하여 최신 패키지를 내려받아 설치
- 패키지 저장소를 이용하려면 먼저 패키지 저장소를 설정해야 함
- 패키지 저장소에 대한 정보는 `/etc/apt/sources.list.d/ubuntu.sources` 파일에 저장되어 있는데 이 파일을 수정하면 저장소를 추가하거나 삭제할 수 있음

Package

❖ 우분투 패키지 관리

✓ apt-cache

- 기능: APT 캐시에 질의하여 여러 가지 정보를 검색
- 형식: apt-cache [옵션] 서브 명령
- 옵션
 - ◆ -f: 검색 결과로 패키지에 대한 전체 기록을 출력
 - ◆ -h: 간단한 도움말을 출력
- 서브 명령
 - ◆ stats: 캐시에 대한 통계 정보를 출력
 - ◆ dump: 현재 설치된 패키지를 업그레이드
 - ◆ search 키워드: 캐시에서 키워드를 검색
 - ◆ showpkg 패키지명: 패키지에 대한 의존성 정보와 역의존성 정보를 검색하여 출력
 - ◆ show 패키지명: 패키지에 대한 간단한 정보를 출력
 - ◆ pkgnames: 사용 가능한 모든 패키지의 이름을 출력

Package

❖ 우분투 패키지 관리

✓ apt-cache

● 예시

- ◆ apt-cache stats
- ◆ apt-cache show vsftpd
- ◆ apt-cache search vsftpd

Package

❖ 우분투 패키지 관리

✓ apt-cache

- APT 캐시 정보 통계 정보 확인 - adam@adam:~\$ **apt-cache stats**

Total package names: 116677 (3,267 k)

Total package structures: 104437 (4,595 k)

Normal packages: 69626

Pure virtual packages: 1835

Single virtual packages: 22251

Mixed virtual packages: 449

Missing: 10276

Total distinct versions: 77542 (6,824 k)

Total distinct descriptions: 147704 (3,545 k)

Total dependencies: 467110/119317 (11.3 M)

Total ver/file relations: 22921 (550 k)

Total Desc/File relations: 16632 (399 k)

Total Provides mappings: 35064 (842 k)

Total globbed strings: 219026 (5,202 k)

Total slack space: 92.3 k

Total space accounted for: 38.1 M

Total buckets in PkgHashTable: 196613

Total buckets in GrpHashTable: 196613

Package

❖ 우분투 패키지 관리

✓ apt-cache

● APT 캐시 정보 통계 정보 확인 - adam@adam:~\$ **apt-cache stats**

- ◆ 전체 패키지 이름: 캐시에 있는 패키지 이름의 전체 개수를 의미
- ◆ 일반 패키지: 일반적으로 사용하는 패키지의 개수를 의미
- ◆ 순수 가상 패키지(pure virtual package): 가상 패키지는 패키지의 이름만 제공하며 그 이름을 가진 별도의 패키지가 실제로 있는 것은 아님 - mail-transport-agent 패키지는 순수 가상 패키지로 여러 패키지가 연계하여 이 기능을 제공하지만 이런 이름을 가진 패키지는 없음
- ◆ 단일 가상 패키지(Single virtual package): 한 패키지가 특정한 가상 패키지의 기능을 제공하는 경우로 Xll-text-viewer는 xless 패키지가 단독으로 제공하는 가상 패키지
- ◆ 혼합 가상 패키지(mixed virtual package): 특정한 가상 패키지를 제공하거나 가상 패키지의 이름을 패키지 이름으로 사용하는 경우로 debconf 패키지는 실제 패키지이면서 debconf-tiny 패키지에 의해 제공
- ◆ 빠짐(missing): 의존성은 있지만 어떠한 패키지도 제공하지 않는 패키지
- ◆ 개별 버전 전체(total distinct version): 캐시에 있는 패키지 버전의 개수로 이 숫자는 전체 패키지 이름의 개수보다 작거나 그 개수와 같음
- ◆ 전체 의존성(total dependencies): 캐시에 있는 모든 패키지의 의존성 관계를 의미

Package

❖ 우분투 패키지 관리

✓ apt-cache

- 사용 가능한 패키지 이름 확인: adam@adam:~\$ **apt-cache pkgnames**
- 패키지 이름 검색: adam@adam:~\$ **apt-cache search vsftpd**
 - vsftpd - lightweight, efficient FTP server written for security
 - vsftpd-dbgsym - lightweight, efficient FTP server written for security (debug)
 - ccze - robust, modular log coloriser
 - ftpd - File Transfer Protocol (FTP) server
 - resource-agents-extra - Cluster Resource Agents
 - yasat - simple stupid audit tool
- 패키지 정보 검색: adam@adam:~\$ **apt-cache show vsftpd**
 - Package: vsftpd
 - Architecture: arm64
 - Version: 3.0.5-0ubuntu1
 - Priority: extra
 - Section: net
 - Origin: Ubuntu

Package

❖ 우분투 패키지 관리

✓ apt-cache

- 패키지 의존성 검색: adam@adam:~\$ **apt-cache showpkg vsftpd**

Package: vsftpd

Versions:

3.0.5-0ubuntu1 (/var/lib/apt/lists/ports.ubuntu.com_ubuntu-ports_dists_jammy_main_binary-arm64_Packages)

Description Language:

File: /var/lib/apt/lists/ports.ubuntu.com_ubuntu-ports_dists_jammy_main_binary-arm64_Packages

MD5: 81386f72ac91a5ea48f8db0b023f3f9b

Description Language: en

File: /var/lib/apt/lists/ports.ubuntu.com_ubuntu-ports_dists_jammy_main_i18n_Translation-en

MD5: 81386f72ac91a5ea48f8db0b023f3f9b

Reverse Depends:

vsftpd-dbg,vsftpd 3.0.5-0ubuntu1

ubumirror,vsftpd

Dependencies:

3.0.5-0ubuntu1 - debconf (18 0.5) debconf-2.0 (0 (null)) libc6 (2 2.34) libcap2 (2 1:2.10) libpam0g (2 0.99.7.1) libssl3 (2 3.0.0~~alpha1) libwrap0 (2 7.6-4~) adduser (0 (null)) libpam-modules (0 (null)) lsb-base (2 3.0-6) netbase (0 (null)) procps (0 (null)) sysvinit-utils (2 2.96) ftp-server (0 (null)) logrotate (0 (null)) ssl-cert (0 (null)) ftp-server (0 (null))

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

● 명령 형식: apt-get [옵션] 서브 명령

◆ 옵션

- -d: 패키지를 내려받기만 수행
- -f: 의존성이 깨진 패키지를 수정하려고 시도
- -h: 간단한 도움말을 출력
- -y: 설치나 업그레이드를 할 때 사용하면 설치 여부를 묻는 부분을 생략

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

● 명령 형식: apt-get [옵션] 서브 명령

◆ 서브 명령

- update: 패키지 저장소에서 새로운 패키지 정보를 가져옴
- upgrade: 현재 설치된 패키지를 업그레이드
- install 패키지명: 패키지를 설치
- remove 패키지명: 패키지를 삭제
- purge 패키지명: 패키지를 삭제
- download 패키지명: 패키지를 현재 디렉토리로 내려받기
- autoclean: 불완전하게 내려받았거나 오래된 패키지를 삭제
- clean: /var/cache/apt/archives에 캐시되어 있는 모든 패키지를 삭제하여 디스크 공간을 확보
- check: 의존성이 깨진 패키지를 확인

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

● update

◆ update 서버 명령은 명시한 저장소로부터 패키지 정보를 다시 읽어 동기화

◆ 새로운 패키지 정보를 가져와서 APT 캐시를 수정하는 것

◆ 저장소를 추가한 경우 이 명령을 반드시 실행

◆ adam@adam:~\$ **sudo apt-get update**

[sudo] password for adam:

Hit:1 <http://ports.ubuntu.com/ubuntu-ports> jammy InRelease

Get:2 <http://ports.ubuntu.com/ubuntu-ports> jammy-updates InRelease [119 kB]

Hit:3 <http://ports.ubuntu.com/ubuntu-ports> jammy-backports InRelease

Get:4 <http://ports.ubuntu.com/ubuntu-ports> jammy-security InRelease [110 kB]

Hit:5 <https://ppa.launchpadcontent.net/ubuntu-mozilla-security/ppa/ubuntu> jammy InRelease

Get:6 <http://ports.ubuntu.com/ubuntu-ports> jammy-updates/main arm64

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

● upgrade

◆ 현재 설치된 모든 패키지 중에서 새로운 버전이 나온 패키지를 업그레이드하려면 upgrade 서브 명령을 사용

◆ adam@adam:~\$ **sudo apt-get upgrade**

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

Calculating upgrade... Done

0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

● install

◆ 하나 이상의 패키지를 설치하거나 업그레이드할 때 사용

◆ adam@adam:~\$ **sudo apt-get install xterm**

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following additional packages will be installed:

xbitmaps

Suggested packages:

xfonts-cyrillic

The following NEW packages will be installed:

xbitmaps xterm

0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

● remove, purge

◆ 패키지를 삭제하는 명령

◆ remove 서브 명령은 패키지를 삭제할 때 해당 패키지의 설정 파일을 남겨두는데 이는 나중에 같은 패키지를 다시 설치할 때 재활용하기 위함으로 만약 설정 파일을 포함하여 패키지를 삭제하려면 purge 서브 명령을 사용

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

● remove, purge

◆ adam@adam:~\$ **sudo apt-get remove xterm**

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following packages will be REMOVED:

xterm

0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.

After this operation, 2,412 kB disk space will be freed.

Do you want to continue? [Y/n]

(Reading database ... 114563 files and directories currently installed.)

Removing xterm (372-1ubuntu1) ...

Processing triggers for hicolor-icon-theme (0.17-2) ...

Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...

Processing triggers for man-db (2.10.2-1) ...

Processing triggers for mailcap (3.70+nmu1ubuntu1) ...

Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...

Package

❖ 우분투 패키지 관리

✓ apt-get 또는 apt

- 패키지 자동 정리 및 삭제하기: `sudo apt-get autoremove`
- 디스크 공간 정리: `sudo apt-get clean`
- 패키지 다운로드: `sudo apt-get download xterm`
- 패키지의 소스 관련 서브 명령: source 명령
 - ◆ 특정 패키지의 소스 코드를 내려받기만 하는 경우: `sudo apt-get --download-only source 패키지명`
 - ◆ 특정 패키지의 소스 코드를 내려받고 압축을 푸는 경우: `sudo apt-get source 패키지명`
 - ◆ 특정 패키지의 소스 코드를 내려받아 압축을 풀고 컴파일하는 경우: `sudo apt-get --compile source 패키지명`

Package

❖ 우분투 패키지 관리

✓ 연습문제

- python3 패키지가 존재하는지 검색하고 존재하면 업그레이드를 수행하고 존재하지 않는다면 설치하십시오.



Package

❖ 우분투 패키지 관리

✓ 외부 레포지토리 활용

- 어떤 애플리케이션은 우분투 공식 레포지토리가 아닌 곳에서 다운로드 받아서 설치해야 하는 경우가 있는데 이런 경우에는 키 와 저장소를 등록하고 저장소의 패키지를 업데이트해서 설치하는 경우가 있음
- 도커 설치
 - ◆ 우분투 시스템 패키지 업데이트: `sudo apt-get update`
 - ◆ 필요한 패키지 설치: `sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common`
 - ◆ Docker의 공식 GPG키를 추가: `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
 - ◆ Docker의 공식 apt 저장소를 추가: `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`
 - ◆ 시스템 패키지 업데이트: `sudo apt-get update`
 - ◆ Docker 설치: `sudo apt-get install docker-ce docker-ce-cli containerd.io`
 - ◆ 도커 실행상태 확인: `sudo systemctl status docker`
 - ◆ 도커 실행: `sudo docker run hello-world`

Package

❖ 우분투 패키지 관리

✓ dpkg 명령으로 패키지 관리

● 개요

- ◆ APT 명령은 fedora의 yum이나 dnf 명령과 비슷하게 인터넷이 연결된 환경에서 패키지를 자동으로 설치하는데 fedora의 rpm과 같은 명령이 dpkg
- ◆ 데비안 계열의 리눅스에서 패키지를 관리하는 데 사용
- ◆ APT 명령도 내부적으로는 dpkg 명령을 사용
- ◆ 일반적으로 패키지를 설치할 때 APT 명령을 사용하면 되지만 시스템의 특정 파일이 어느 패키지에 속했는지를 확인하는 등 보다 세부적인 기능을 사용하려면 dpkg 명령이 필요

Package

❖ 우분투 패키지 관리

✓ dpkg 명령으로 패키지 관리

● 명령

◆ 형식: dpkg [옵션] 파일명 또는 패키지명

◆ 옵션

- - l: 설치된 패키지의 목록을 출력
- - I 패키지명: 패키지의 설치 상태를 출력
- - s 패키지명: 패키지의 상세 정보를 출력
- - L 패키지명: 패키지에서 설치된 파일의 목록을 출력
- - c .deb 파일: 지정한 .deb 파일의 내용을 출력
- - i .deb 파일: 해당 파일을 설치 (sudo)
- - r 패키지명: 해당 패키지를 삭제 (sudo)
- - P 패키지명: 해당 패키지와 설정 정보를 모두 삭제(sudo)
- - x .deb 파일 디렉토리: 해당 파일을 지정한 디렉토리에 풀어놓기

Package

❖ 우분투 패키지 관리

✓ dpkg 명령으로 패키지 관리

● 사용

◆ 패키지 목록 보기: -l

- 출력 결과에서 첫 글자는 상단의 희망 상태를 나타내는 것이고 두 번째 글자는 상태이며 패키지 이름을 뒤에 기재하면 해당 패키지에 대한 정보만 출력
- ~\$ dpkg -l

Desired=Unknown/Install/Remove/Purge/Hold

| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend

|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)

||/ Name Version >

+++-----
----->

ii adduser 3.137ubuntu1 >

ii alsa-topology-conf 1.2.5.1-2 >

- 처음 두 글자 ii는 희망 상태가 i(설치) 상태도 i(설치) 임을 의미

Package

❖ 우분투 패키지 관리

✓ dpkg 명령으로 패키지 관리

● 사용

◆ 패키지 상세 정보 보기: -s

▪ ~\$ dpkg -s zip

Package: zip

Status: install ok installed

Priority: optional

Section: utils

Installed-Size: 532

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Architecture: arm64

Multi-Arch: foreign

Version: 3.0-13build1

Depends: libbz2-1.0, libc6 (>= 2.34)

Recommends: unzip

Description: Archiver for .zip files

This is InfoZIP's zip program. It produces files that are fully compatible with the popular PKZIP program; however, the command line options are not identical. In other words, the end result is the same, but the methods differ. :-)

- 패키지가 없으면 설치 후 수행

Package

❖ 우분투 패키지 관리

✓ aptitude 명령으로 패키지 관리

● 개요

- ◆ aptitude는 APT 명령처럼 패키지 관리를 자동화하여 쉽게 작업할 수 있는 명령
- ◆ 패키지를 관리하기 위한 서브 명령이 apt-get 명령과 거의 유사
- ◆ aptitude 명령의 또 다른 특징은 옵션이나 서브 명령 없이 실행할 경우 curses를 이용한 비주얼 모드로 동작한다는 것
- ◆ curses는 터미널에서 텍스트 그래픽 기능을 제공하므로 메뉴를 선택하면서 작업할 수 있음

Package

❖ 우분투 패키지 관리

✓ aptitude 명령으로 패키지 관리

● 명령의 사용 형식

◆ 형식: aptitude [서브 명령]

◆ 서브 명령

- 단독 실행: curses 프로그램이 나타남
- search 키워드: 키워드를 검색하여 일치하는 패키지 목록을 출력
- update: 패키지 저장소를 업데이트
- upgrade: 모든 패키지를 최신 버전으로 업그레이드
- show 패키지명: 패키지에 대한 자세한 정보를 출력
- download 패키지명: 패키지를 내려받기
- clean: 패키지 캐시 디렉토리에서 모든 패키지 파일을 삭제
- install: 패키지를 설치
- remove: 패키지를 삭제
- purge: 패키지와 설정 파일을 모두 삭제

Package

❖ 우분투 패키지 관리

✓ aptitude 명령으로 패키지 관리

● 명령의 사용

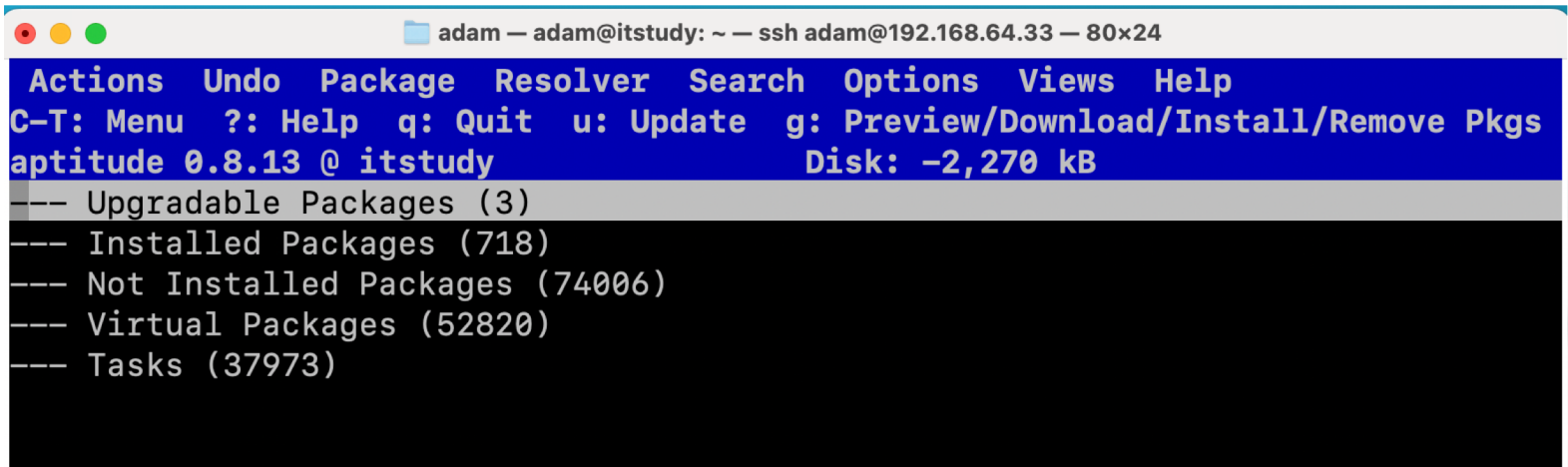
◆ ~\$ sudo aptitude

sudo: aptitude: command not found

◆ ~\$ sudo apt install aptitude

Reading package lists... Done

◆ ~\$ sudo aptitude



```
adam — adam@itstudy: ~ — ssh adam@192.168.64.33 — 80x24
Actions Undo Package Resolver Search Options Views Help
C-T: Menu ?: Help q: Quit u: Update g: Preview/Download/Install/Remove Pkgs
aptitude 0.8.13 @ itstudy Disk: -2,270 kB
--- Upgradable Packages (3)
--- Installed Packages (718)
--- Not Installed Packages (74006)
--- Virtual Packages (52820)
--- Tasks (37973)
```

Package

❖ 우분투 패키지 관리

✓ 스냅

● 개요

- ◆ 스냅은 우분투가 새로 도입한 패키지 형식으로 샌드박스 형태의 패키지
- ◆ 패키지를 만들 때 프로그램이 사용하는 모든 라이브러리를 패키지 안에 포함하는 방식으로 외부에서 받은 파일을 그냥 실행하는 것이 아니라 보호된 영역에서 실행해보는 것으로 외부 파일이 내부 시스템에 악영향을 주는 것을 방지하는 보안 기술을 위한 것(Sandbox)
- ◆ 안드로이드폰이나 아이폰같은 스마트폰의 운영체제에서 사용하고 있는 패키지 개념인 샌드박스 형식을 스냅이 사용함으로써 얻게 되는 장점
 - 개발자가 다른 패키지나 라이브러리와 의존성을 신경 쓰지 않아도 됨
 - 기존 시스템과 격리되어 실행하는 샌드박스 형식이므로 보안이 강화
- ◆ 단점은 패키지의 용량이 커진다는 것인데 이는 프로그램에 필요한 모든 라이브러리를 포함하기 때문
- ◆ 우분투는 스냅 패키지 형식을 통해 운영체제에 독립적인 패키지를 제공하여 모바일에서 클라우드에 이르기까지 크로스 플랫폼 기술을 지원하려 하며 스냅 스토어 기능을 제공하여 사용자들이 개발한 스냅을 유료나 무료로 공개할 수 있음

Package

❖ 우분투 패키지 관리

✓ 스냅

● 기본 형식

◆ snap [옵션] 명령

◆ 옵션

- -h: 도움말을 출력

◆ 명령

- disable: 스냅 서비스와 실행 파일의 사용을 중지
- download 스냅명: 지정한 스냅 패키지 내려받기
- enable: 스냅 서비스와 실행 파일의 사용을 시작
- find 스냅명: 지정한 스냅을 검색
- info 스냅명: 지정한 스냅의 상세 정보를 출력
- install 스냅명: 지정한 스냅을 설치
- list: 설치한 스냅의 목록을 출력
- remove 스냅명: 지정한 스냅을 삭제

Package

❖ 우분투 패키지 관리

✓ 스냅

● 사용

◆ 스냅 찾기: snap find

▪ ~\$ sudo snap find hello-world

Name	Version	Publisher	Notes	Summary
hello-world snaps	6.4	canonical✓	-	The 'hello-world' of snaps
hello-world2jerryattheedge stage snap example	0.1	jmackenb	-	super early

Package

❖ 우분투 패키지 관리

✓ 스냅

● 사용

◆ 스냅 설치: snap install

- ~\$ sudo snap install hello-world

◆ 스냅 목록 보기: snap list

- ~\$ sudo snap list

Name	Version	Rev	Tracking	Publisher	Notes
core	16-2.61.4-20240607	17201	latest/stable	canonical✓	core
hello-world	6.4	29	latest/stable	canonical✓	-

◆ 설치된 파일 확인

- ~\$ ls /snap

bin core hello-world README

Package

❖ 우분투 패키지 관리

✓ 스냅

● 사용

◆ 상세 정보 보기: snap info

- ~\$ sudo snap info hello-world

name: hello-world

summary: The 'hello-world' of snaps

publisher: Canonical✓

store-url: <https://snapcraft.io/hello-world>

contact: snaps@canonical.com

license: unset

description: |

This is a simple hello world example.

commands:

- hello-world.env

- hello-world.evill

- hello-world

- hello-world.sh

Package

❖ 우분투 패키지 관리

✓ 스냅

● 사용

◆ 삭제: snap remove

- ~\$ sudo snap remove hello-world

hello-world removed

- ~\$ sudo snap list

Name	Version	Rev	Tracking	Publisher	Notes
core	16-2.61.4-20240607	17201	latest/stable	canonical✓	core

Package

❖ 우분투 패키지 관리

✓ 스냅

● 실습

- ◆ calculator 를 검색
- ◆ kcalc 설치
- ◆ kcalc 상세 정보를 확인
- ◆ kcalc 삭제

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 개요

- ◆ 명령은 다른 시스템과 파일을 주고 받거나 백업을 하기 위해 여러 파일이나 디렉토리를 하나의 아카이브 파일로 생성하거나 기존 아카이브에서 파일을 추출하기 위해 사용

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 명령

◆ 형식. tar 기능[옵션] [아카이브 파일] [파일명]

◆ 기능

- c: 새로운 tar 파일을 생성
- t: tar 파일의 내용을 출력
- x: tar 파일에서 원본 파일을 추출
- r: 새로운 파일을 추가
- u: 수정된 파일을 업데이트

◆ 옵션

- f: 아카이브 파일이나 테이프 장치를 지정하는데 파일명을 - 으로 지정하면 tar 파일 대신 표준 입력에서 읽어들이м
- v: 처리하고 있는 파일의 정보를 출력
- h: 심볼릭 링크의 원본 파일을 포함
- p: 파일 복구 시 원래의 접근 권한을 유지
- j: bzip2로 압축하거나 해제
- z: gzip으로 압축하거나 해제

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 자주 사용하는 옵션

- ◆ 아카이브 생성하기: cvf 옵션
- ◆ 아카이브의 압축 해제: xvf 옵션
- ◆ 아카이브의 내용 확인하기: tvf 옵션
- ◆ 아카이브 업데이트하기: uvf 옵션
- ◆ 아카이브에 파일 추가하기: rvf 옵션

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 실습

◆ ~\$ mkdir ex_archive

◆ ~\$ cd ex_archive

◆ ~/ex_archive\$ mkdir sample1

◆ ~/ex_archive\$ mkdir sample2

◆ ~/ex_archive\$ mkdir sample3

◆ 아카이브: ~/ex_archive\$ tar cvf sample1.tar sample1

sample1/

◆ 아카이브 파일 내용 확인: ~/ex_archive\$ tar tvf sample1.tar

drwxrwxr-x adam/adam

0 2024-11-03 00:51 sample1/

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 실습

◆ 디렉토리 삭제:~/ex_archive\$ rm -r sample1

◆ 디렉토리 확인:~/ex_archive\$ ls

sample1.tar sample2 sample3

◆ 아카이브 해제:~/ex_archive\$ tar xvf sample1.tar

sample1/

◆ 디렉토리 확인:~/ex_archive\$ ls

sample1 sample1.tar sample2 sample3

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 실습

◆ 파일 생성:~/ex_archive\$ touch sample1/data

◆ 아카이브 업데이트:~/ex_archive\$ tar uvf sample1.tar sample1
sample1/data

◆ 아카이브 파일 확인:~/ex_archive\$ tar tvf sample1.tar

drwxrwxr-x adam/adam 0 2024-11-03 00:51 sample1/

-rw-rw-r-- adam/adam 0 2024-11-03 00:59 sample1/data

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 실습

◆ 파일 복사:~/ex_archive\$ cp /etc/services .

◆ 아카이브에 파일 추가:~/ex_archive\$ tar rvf sample1.tar services

◆ 확인:~/ex_archive\$ tar tvf sample1.tar

drwxrwxr-x adam/adam 0 2024-11-03 00:51 sample1/

-rw-rw-r-- adam/adam 0 2024-11-03 00:59 sample1/data

-rw-r--r-- adam/adam 12813 2024-11-03 01:01 services

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 아카이브 생성하고 압축

- ◆ 아카이브를 생성하면서 동시에 압축 가능
- ◆ 압축을 하면 아카이브 파일의 크기가 작아지므로 디스크 공간이 부족할 때 이용
- ◆ 보통 tar로 아카이브를 만든 다음 gzip 같은 압축 명령으로 압축했으나 생성과 압축을 동시에 처리할 수 있도록 옵션이 추가
- ◆ gzip으로 압축할 경우 z 옵션을 사용
- ◆ 아카이브명에 .gz를 붙여서 gzip으로 압축된 것임을 표시하는 것을 권장

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 아카이브 생성하고 압축

◆ 디렉토리 확인: ~/ex_archive\$ ls -l

total 48

drwxrwxr-x 2 adam adam 4096 Nov 3 00:59 sample1

-rw-rw-r-- 1 adam adam 20480 Nov 3 01:02 sample1.tar

drwxrwxr-x 2 adam adam 4096 Nov 3 00:51 sample2

drwxrwxr-x 2 adam adam 4096 Nov 3 00:51 sample3

-rw-r--r-- 1 adam adam 12813 Nov 3 01:01 services

◆ 압축:~/ex_archive\$ tar czvf sample1.tar.gz sample1

sample1/

sample1/data

◆ 디렉토리 확인:~/ex_archive\$ ls -l

total 52

drwxrwxr-x 2 adam adam 4096 Nov 3 00:59 sample1

-rw-rw-r-- 1 adam adam 20480 Nov 3 01:02 sample1.tar

-rw-rw-r-- 1 adam adam 142 Nov 3 01:11 sample1.tar.gz

파일 압축

❖ 파일 아카이브

✓ tar(tape archive)

● 아카이브 생성하고 압축

◆ 압축:~/ex_archive\$ tar czvf sample1.tar.bz2 sample1

sample1/

sample1/data

◆ 디렉토리 확인:~/ex_archive\$ ls -l

total 56

drwxrwxr-x 2 adam adam 4096 Nov 3 00:59 sample1

-rw-rw-r-- 1 adam adam 20480 Nov 3 01:02 sample1.tar

-rw-rw-r-- 1 adam adam 142 Nov 3 01:13 sample1.tar.bz2

-rw-rw-r-- 1 adam adam 142 Nov 3 01:11 sample1.tar.gz

파일 압축

❖ 파일 아카이브

✓ gzip/gunzip

- 압축률이 좋은 압축 유틸
- gzip 명령은 지정한 파일을 압축하여 파일명.gz 파일을 생성
- 형식
 - ◆ 형식: gzip [옵션] [파일명]
 - ◆ 옵션
 - -d: 파일 압축을 해제
 - -l: 압축된 파일의 정보를 출력
 - -r: 하위 디렉터리를 탐색하여 압축
 - -t: 압축 파일을 검사
 - -v: 압축 정보를 화면에 출력
 - -9: 최대한 압축

파일 압축

❖ 파일 아카이브

✓ gzip/gunzip

● 실습

◆ 압축된 파일의 내용 조회: ~/ex_archive\$ zcat sample1.tar.gz | more

sample1/0000775000175000017500000000000014711544432011015 5ustar
adamadam

sample1/data0000664000175000

01750000000000000014711545417011643 0ustar adamadam

services0000644000175000017500000003101514711545606011223 0

ustar adamadam# Network services, Internet style

#

Updated from [https://www.iana.org/assignments/service-names-port-](https://www.iana.org/assignments/service-names-port-numbers/servi)
numbers/servi

ce-names-port-numbers.xhtml .

#

New ports will be added on request if they have been officially assigned

by IANA and used in the real-world or are needed by a debian package.

If you need a huge list of used numbers please install the nmap package.

파일 압축

❖ 파일 아카이브

✓ gzip/gunzip

● 실습

◆ 압축 풀기:~/ex_archive\$ gunzip sample1.tar.gz

◆ 압축 풀기:~/ex_archive\$ ls

sample1 sample1.tar sample1.tar.bz2 sample2 sample3 services

파일 압축

❖ 파일 아카이브

✓ bzip2/bunzip2

- bzip2는 gzip에 비해 대체로 압축률은 좋지만 속도가 약간 느림
- bzip2 명령은 지정한 파일을 압축하여 파일명.bz2 파일을 생성
- 명령
 - ◆ 형식: bzip2 [옵션] [파일명]
 - ◆ 옵션
 - -d: 파일 압축을 해제
 - -l: 압축된 파일의 정보 출력
 - -t: 압축 파일을 검사
 - -v: 압축 정보를 화면에 출력
 - --best: 최대한 압축

파일 압축

❖ 파일 아카이브

✓ bzip2/bunzip2

● 실습

◆ 기존 파일 삭제:~/ex_archive\$ rm sample1.tar.bz2

◆ 압축:~/ex_archive\$ bzip2 sample1.tar

◆ 확인:~/ex_archive\$ ls

sample1 sample1.tar.bz2 sample2 sample3 services

◆ 압축된 파일 확인:~/ex_archive\$ bzcata sample1.tar.bz2

sample1/000077500017500001750000000000014711545417011021 5ustar
adamadam/sample1/data0000664000175000017500000000000147115454170116
43 0ustar adamadamadam@its

◆ 복원:~/ex_archive\$ bunzip2 sample1.tar.bz2

◆ 확인:~/ex_archive\$ ls

sample1 sample1.tar sample2 sample3 services

파일 압축

❖ 파일 아카이브

✓ 연습문제

● 실습

- ◆ sample2 디렉터리를 아카이브 파일로 묶기 - 파일명은 sample2.tar
- ◆ sample2.tar 파일의 내용을 확인
- ◆ sample2.tar 파일을 sample3 디렉터리로 이동
- ◆ sample2.tar 파일을 bzip2로 압축
- ◆ sample2.tar.bz2 파일의 압축을 해제
- ◆ sample2.tar 파일을 아카이브 해제

파일 압축

❖ 파일 아카이브

✓ xz

- 확장명.xz로 압축
- 비교적 최신 압축 명령으로 압축률이 뛰어남
- 명령
 - ◆ xz 파일이름: 파일이름을 압축 파일인 파일이름.xz로 만들고 기존 파일은 삭제
 - ◆ xz -d 파일이름.xz: 파일이름.xz의 압축을 풀어서 파일이름으로 만듦
 - ◆ xz -l 파일이름.xz: 압축 파일에 포함된 파일 목록과 압축률 등 출력
 - ◆ xz -k 파일이름: 압축 후 기존 파일을 삭제하지 않고 그대로 둠

파일 압축

❖ 파일 아카이브

✓ zip/unzip

- Windows용과 호환되는 확장명 zip으로 압축하거나 풀기
- 명령
 - ◆ zip 새로생성될파일이름.zip 압축할파일이름
 - 압축할 파일 이름을 새로생성될파일이름.zip으로 압축
 - 기존 파일은 그대로 남아 있음
 - ◆ unzip 압축파일이름.zip
 - 압축파일이름.zip의 압축을 풀기
 - 압축 파일은 그대로 남아 있음

파일 다운로드

❖ wget 명령어

✓ 개요

- Web Get의 약어로 웹 상의 파일을 다운로드 받을 때 사용하는 명령어로 wget은 비 상호작용 네트워크 다운로더로 네트워크 상에서 데이터를 다운로드하는 기능을 수행
- wget이 상호 작용을 필요로 하지 않기(non-interactive) 때문에 사용자가 로그인하지 않은 상태 동안에도 백그라운드 상태에서 동작할 수 있음
- HTTP, HTTPS, FTP 프로토콜을 지원하며 HTTP proxy에서 데이터를 가져올 수도 있음
- wget은 HTML과 XHTML, CSS 페이지를 다운로드 받아 웹 사이트의 로컬 버전을 만들 수 있고 본래 사이트의 디렉토리 구조를 만들 수 있으며 recursive downloading을 지원해서 사이트 전체를 쉽게 내려받을 수 있음
- wget은 느리거나 불안정한 네트워크 환경에서도 매우 잘 작동하는 견고한 프로그램으로 네트워크 환경이 불안해서 도중에 연결이 끊겼다면 연결이 끊긴 시점부터 다운로드 받는 기능도 가지고 있음
- NGS 데이터의 de novo assembler 중의 하나인 velvet은 웹(<https://www.ebi.ac.uk/~zerbino/velvet/>)상에서 프로그램을 다운로드 받을 수 있는데 웹 브라우저에서 소스 다운로드 링크를 클릭하면 다시 분석 서버로 해당 파일을 업로드 해야 하는 번거로움이 발생할 수 있는데 이런 경우 wget 명령어를 이용하여 리눅스 명령 행에서 바로 해당 파일을 다운로드 받을 수 있는데 wget 명령어와 다운로드 링크를 입력하기만 하면 자동으로 현재 디렉토리에 파일이 다운로드 받아짐

파일 다운로드

❖ wget 명령어

✓ 사용

\$ wget [옵션]... [URL]...

● 옵션

- ◆ 기본 다운로드: wget <https://example.com/file.zip>
- ◆ 다른 이름으로 저장: wget -O my_filename.zip <https://example.com/file.zip>
- ◆ 다운로드 재개: wget -c <https://example.com/large-file.iso>
- ◆ 백그라운드에서 다운로드: wget -b <https://example.com/heavy-data.tar.gz>
- ◆ 속도 제한: wget --limit-rate=500k <https://example.com/file.zip>

● 웹 사이트 미러링

wget -m <https://example.com>

- ◆ -r (Recursive): 하위 디렉토리를 따라가며 다운로드
- ◆ -np (No Parent): 상위 디렉토리로 이동하지 않도록 제한
- ◆ -k (Convert Links): 로컬에서 오프라인으로 보기 편하도록 링크를 변환

파일 다운로드

❖ curl 명령어

✓ 개요

- curl (Client URL)은 터미널 환경에서 다양한 프로토콜(HTTP, HTTPS, FTP, SMTP 등)을 사용하여 데이터를 전송하거나 가져오는 매우 강력한 도구
- 서버와의 데이터 통신 및 API 테스트에 최적화되어 있어 개발자들이 필수적으로 사용하는 도구

✓ 기본 문법

- curl [옵션] [URL]

파일 다운로드

❖ curl 명령어

✓ 활용

- 웹 페이지 소스 보기: curl <https://www.google.com>
- 파일로 저장
 - ◆ -o (소문자): 저장될 파일 이름을 직접 지정: curl -o my_page.html <https://example.com>
 - ◆ -O (대문자): 원본 서버의 파일 이름을 그대로 사용: curl -O <https://example.com/image.png>
- HTTP 헤더 확인: curl -I <https://example.com>
- 리다이렉트 자동 추적: curl -L <http://google.com>
- 데이터 전송 및 API 호출
 - ◆ REST API를 테스트할 때 가장 많이 사용되는 옵션
 - -X: 사용할 HTTP 메서드(GET, POST, PUT, DELETE 등)를 지정
 - -d: 전송할 데이터(Body)를 입력
 - -H: 헤더 정보(Content-Type, 인증 토큰 등)를 추가
 - ◆ JSON 데이터를 POST 방식으로 전송하는 예시

```
curl -X POST https://api.example.com/user ₩
```

```
-H "Content-Type: application/json" ₩
```

```
-d '{"name": "Gemini", "email": "test@test.com"}'
```

파일 다운로드

❖ curl 명령어

✓ 활용

- 자주 사용하는 옵션: -fsSL

◆ 해석

- -f --fail: HTTP 오류 발생 시(404, 500 등) 아무것도 출력하지 않고 실패 처리(스크립트 실행 시 잘못된 HTML 파일이 실행되는 것을 방지)
- -s --silent: 진행 바(Progress bar)나 에러 메시지를 표시하지 않는 정숙 모드
- -S --show-error: s와 함께 사용되며, 진짜 에러가 났을 때만 에러 메시지를 출력
- -L --location: 페이지가 이동(Redirect)되었을 경우 자동으로 새 주소로 가서 데이터를 가져옴

◆ 이 명령은 파이프(|)를 통해 sh나 bash로 연결되어 설치 스크립트를 실행하는 데 사용

- `curl -fsSL https://get.docker.com -o get-docker.sh`
- `sudo sh get-docker.sh`
- 한 줄로 실행: `curl -fsSL https://install.direct/go.sh | sudo bash`

파일 다운로드

❖ curl 명령어

✓ 활용

- 자주 사용하는 옵션: -fsSL

◆ 사용하는 이유

- 깔끔한 출력: 설치 중에 불필요한 다운로드 진행 바가 화면을 가득 채우지 않게 함
- 안전성: 만약 URL이 잘못되어 404 에러 페이지(HTML)가 응답으로 오면, -f 옵션 덕분에 그 HTML 코드가 쉘(bash)로 넘어가 실행되는 사고를 막아줌
- 유연성: 서버 설정에 따라 다운로드 위치가 바뀌어도 -L 옵션이 알아서 추적해 줌

파일 다운로드

❖ wget 과 curl 차이점

특징	wget	curl
주요 목적	단순 파일/사이트 다운로드	데이터 전송 및 API 테스트
재시도	네트워크 오류 시 자동 재시도 강점	기본적으로 재시도하지 않음
재귀적 다운로드	지원 (웹사이트 전체 복제 가능)	지원하지 않음
복잡성	사용법이 단순하고 직관적	매우 다양한 프로토콜 및 옵션 제공