



LINUX 개발환경(3)

- LINUX 패키지 관리
- 파일 아카이브와 압축



LINUX 패키지 관리

□ 패키지 관리(Package Management)

- 새로운 소프트웨어를 설치, 업데이트, 삭제하는 일
- 패키지 배포 형태
 - 소스코드: 하나의 아카이브 파일(tar)로 묶은 후 압축하여 배포
 - 바이너리: 응용프로그램, 라이브러리 파일, 버전 정보, 의존성 등의 메타정보 파일을 포함하여 배포
- 하나의 패키지가 다른 패키지나 공유라이브러리 간의 의존성(dependency) 문제를 해결하기 위한 것이 리눅스 패키지 관리의 핵심 역할

□ 리눅스에서 주로 사용하는 패키지

- Debian 계열(Debian, Ubuntu 등): **.deb** 파일
- RedHat 계열(RedHat, Fedora, CentOS): **.rpm** 파일
- openSUSE 계열: openSUSE용 **.rpm** 파일



LINUX 패키지 관리

□ 우분투 패키지의 특징

- 바이너리 파일로 배포되어 컴파일이 필요 없음
- 패키지의 파일이 관련 디렉터리에 바로 설치됨
- 패키지를 삭제할 때 관련된 파일을 일괄 삭제 가능함
- 기존 설치된 패키지를 삭제하지 않고 바로 업그레이드 가능
- 패키지의 설치 상태를 검증 가능함
- 패키지에 대한 정보를 제공하고 있음
- 패키지 의존성 정보를 확인할 수 있기 때문에
의존성이 있는 패키지를 미리 설치하거나 일괄 설치 가능함



LINUX 패키지 관리

□ 우분투 패키지의 카테고리

- **main**: 우분투에 의해 공식적으로 지원되며 자유롭게 배포 가능
- **restricted**: 우분투에 의해 지원되나 완전한 Free SW는 아님
- **universe**: 기술적 지원을 보장할 수 없으며,
Free SW 일 수도 있고 아닐 수도 있는 리눅스에서 사용 가능한
거의 대부분의 SW 패키지
- **multiverse**: Free SW가 아닌 소프트웨어가 포함되어 있으며,
개인이 직접 라이선스를 확인해야 함

□ 우분투 패키지의 이름 구성

- **/var/cache/apt/archives** 디렉터리에 .deb 파일들 보관(=**APT cache**)
- <파일명>_<버전>-<리비전>_<아키텍처>.deb



LINUX 패키지 관리

□ 우분투 패키지 저장소(Repository)

- Repository: 패키지와 패키지에 대한 정보를 저장하고 있는 서버
- 패키지의 기능 추가나 보안 패치 등의 업그레이드 사항을 집중 관리
- Repository 정보는 [/etc/apt/sources.list](#) 파일에 저장
 - 패키지 유형: **deb**(바이너리 패키지 저장소), **deb-src**(소스 저장소)
 - 저장소 주소: **http** 프로토콜을 사용하는 URL 주소
 - 우분투 버전 정보: 저장소에서 관리하는 패키지의 우분투 버전을 표시
 - 카테고리: 저장소가 가지고 있는 소프트웨어 카테고리 표시

```
linuxer@linuxer-PC:~$ cat /etc/apt/sources.list | more
deb cdrom:[Ubuntu 20.04.1 LTS _Focal Fossa_ - Release amd64 (20200731)]/ focal 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/ focal main restricted
# deb-src http://kr.archive.ubuntu.com/ubuntu/ focal main restricted
...
```



LINUX 패키지 관리

□ LINUX 패키지 관리 도구

- 저수준 도구(low-level tools): 실제 패키지의 설치, 업데이트, 삭제 수행
- 고수준 도구(high-level tools): 의존성 해결, 패키지 검색 등 기능 제공
- 리눅스 배포판 별 패키지 관리 도구

배포판 \ 관리도구	저수준 도구	고수준 도구
Debian 계열 (Ubuntu)	dpkg	•apt-get/apt-cache/apt •aptitude
Redhat 계열 (Fedora, CentOS)	rpm	yum
Slackware 계열 (openSUSE)	rpm	zypper



LINUX 패키지 관리

□ 우분투 패키지 관리 도구

▪ dpkg

- Debian 기반 리눅스에서 사용되는 저수준 패키지 관리자
- .deb 패키지의 설치와 삭제 담당
- 자동으로 패키지를 다운로드하거나 의존성 문제를 해결해 주지 않음

▪ APT명령(`apt-get` / `apt-cache` / `apt`)

- Debian 기반 리눅스에서 사용되는 고수준 패키지 관리자
- 패키지 검색, 다운로드, 설치, 의존성 문제 해결

▪ aptitude

- Debian 기반 리눅스의 또 다른 고수준 패키지 관리자
- `apt-get` 보다 좀 더 개선된 기능 제공



LINUX 패키지 관리

□ 우분투 패키지 정보 확인 명령(1)

▪ APT cache에서 정보 검색: **apt-cache** [옵션] 서버명령

- APT cache = 패키지 데이터베이스

- 서버명령들

- stats**: 캐시에 대한 통계 정보 출력

- pkgnames**: 사용 가능한 모든 패키지의 이름 출력

- search**: 캐시에서 키워드로 검색

- show**: 패키지에 대한 간단한 정보를 출력

- showpkg**: 패키지에 대한 의존성 정보와 역의존성 정보를 검색

```
linuxer@linuxer-PC:~$ apt-cache state
```

```
linuxer@linuxer-PC:~$ apt-cache show vsftpd
```

```
linuxer@linuxer-PC:~$ apt-cache search vsftpd
```




LINUX 패키지 관리

□ 우분투 패키지 정보 확인 명령(2)

▪ APT cache 통계 정보 보기: **apt-cache stats**

- 전체 패키지 이름: 패키지 이름의 전체 개수
- 일반 패키지: 자주 사용되는 패키지의 개수
- 순수 가상 패키지(pure virtual package)
: 패키지 이름만 제공하며 실제로는 있지 않음
- 단일 가상 패키지(single virtual package)
: 한 패키지가 특정 가상 패키지의 기능을 제공
- 혼합 가상 패키지(mixed virtual package)
: 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 수

```
linuxer@linuxer-PC:~$ apt-cache state
전체 패키지 이름 : 99981 (2,799 k)
전체 패키지 구조: 96920 (4,264 k)
일반 패키지: 64081
순수 가상 패키지: 1994
단일 가상 패키지: 16905
혼합 가상 패키지: 2273
빠짐: 11667
개별 버전 전체: 73320 (6,452 k)
개별 설명 전체: 143185 (3,436 k)
전체 의존성: 487188/138645 (12.0 M)
전체 버전/파일 관계: 49058 (1,177 k)
전체 설명/파일 관계: 12113 (291 k)
전체 제공 매핑: 30576 (734 k)
전체 패턴 문자열: 190071 (4,383 k)
전체 빈 용량: 93.0 k
차지하는 전체 용량: 36.0 M
linuxer@linuxer-PC:~$ apt-cache state
```



LINUX 패키지 관리

□ 우분투 패키지 정보 확인 명령(3)

- 사용 가능한 패키지 이름 확인: **apt-cache pkgnames [prefix]**
- 시스템에 설치된 모든 패키지 나열

```
linuxer@linuxer-PC:~$ apt-cache pkgnames | more
```

- prefix 지정으로 특정 패키지 설치 여부 확인 가능

```
linuxer@linuxer-PC:~$ apt-cache pkgnames python | more
linuxer@linuxer-PC:~$ apt-cache pkgnames ftp | more
```



LINUX 패키지 관리

□ 우분투 패키지 정보 확인 명령(4)

- 패키지 이름 상세 검색: **apt-cache search regex...**
 - 주어진 keyword에 따라 대소문자 구별 없이 DB를 검색

```
linuxer@linuxer-PC:~$ 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
```

- 정규식(regex)으로 이름 필터링 가능

```
linuxer@linuxer-PC:~$ apt-cache search colo?r | head
linuxer@linuxer-PC:~$ apt-cache search php7.? | head
linuxer@linuxer-PC:~$ apt-cache search python[2-4] | head
linuxer@linuxer-PC:~$ apt-cache search '^python' | head
linuxer@linuxer-PC:~$ apt-cache search php[0-9].[0-9] | head
```



LINUX 패키지 관리

□ 우분투 패키지 정보 확인 명령(5)

- 패키지 정보 검색: **apt-cache show 패키지이름**
 - 버전, 패키지 크기, 카테고리, 체크섬, 의존성 등의 패키지 기본 정보 출력

```
linuxer@linuxer-PC:~$ apt-cache show vsftpd
```

```
Package: vsftpd
```

```
Architecture: amd64
```

```
Version: 3.0.3-12
```

```
...
```

```
Depends: debconf (>= 0.5) | debconf-2.0, libc6 (>= 2.28), libcap2 (>= 1:2.10), libpam0g (>= 0.99.7.1), libssl1.1 (>= 1.1.0), libwrap0 (>= 7.6-4~), adduser, libpam-modules, lsb-base (>= 3.0-6), netbase, procps
```

```
Filename: pool/main/v/vsftpd/vsftpd_3.0.3-12_amd64.deb
```

```
Size: 115168
```

```
MD5sum: 56bcd8733e098bf40097b54ca087ea17
```

```
SHA1: 037aac061034dc0652f23dd1b41a255902331b55
```

```
SHA256: 33f03b0ef44c8f180c7a0e2dde316287e160c1f1c2f5878017f1a2427b91c4ec
```

```
Homepage: http://vsftpd.beasts.org/
```



LINUX 패키지 관리

□ 우분투 패키지 정보 확인 명령(6)

- 패키지 의존성(dependency) 검색: **apt-cache showpkg 패키지이름**
 - 해당 패키지의 의존성과 역의존성 전체 정보를 출력

```
linuxer@linuxer-PC:~$ apt-cache showpkg vsftpd
```

```
Package: vsftpd
```

```
...
```

Reverse Depends:

```
vsftpd-dbgsym,vsftpd 3.0.3-12
```

```
ubumirror,vsftpd
```

Dependencies:

```
3.0.3-12 - debconf (18 0.5) debconf-2.0 (0 (null)) libc6 (2 2.28) libcap2 (2 1:2.10) libpam0g (2 0.99.7.1) libssl1.1 (2 1.1.0) libwrap0 (2 7.6-4~)
```

```
adduser (0 (null)) libpam-modules (0 (null)) lsb-base (2 3.0-6) netbase (0 (null)) procps (0 (null)) ftp-server (0 (null)) logrotate (0 (null)) ssl-cert (0 (null)) ftp-server (0 (null)) ftp-server:i386 (0 (null)) ftp-server:i386 (0 (null))
```

```
...
```



LINUX 패키지 관리

□ 우분투 패키지 관리 명령(1)

▪ 패키지 관리 명령: **apt-get** [옵션] 서브명령

• 서브명령

update: 패키지 저장소에서 새로운 패키지 정보를 획득

upgrade: 현재 설치되어 있는 패키지를 업그레이드

install: 새로운 패키지를 설치

remove: 설치된 패키지를 삭제

download: 패키지를 현재 디렉터리로 내려받음

autoclean: 불완전하게 내려받았거나 오해된 패키지를 삭제

clean: /var/cache/apt/archives에 캐시되어 있는 모든 패키지를
삭제하여 디스크 공간 확보

check: 의존성이 깨진 패키지를 확인

```
linuxer@linuxer-PC:~$ apt-get update
```

```
linuxer@linuxer-PC:~$ apt-get install vsftpd
```



LINUX 패키지 관리

□ 우분투 패키지 관리 명령(2)

- 패키지 정보 수정: **apt-get update**

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

```
linuxer@linuxer-PC:~$ sudo apt-get update
```

```
[sudo] linuxer의 암호:
```

```
받기:1 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
```

```
기존:2 http://kr.archive.ubuntu.com/ubuntu focal InRelease
```

```
받기:3 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
```

```
받기:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
```

```
...
```

```
내려받기 1,022 k바이트, 소요시간 4초 (229 k바이트/초)
```

```
패키지 목록을 읽는 중입니다... 완료
```



LINUX 패키지 관리

□ 우분투 패키지 관리 명령(3)

▪ 패키지 업그레이드: `apt-get upgrade`

- 현재 시스템에 설치된 모든 패키지의 최신 버전을 설치
- 항상 최신 버전으로 유지할 필요는 없으며,
업그레이드 후 동작되지 않는 프로그램이 있을 수 있으니 주의 요망

```
linuxer@linuxer-PC:~$ sudo apt-get upgrade
```

```
패키지 목록을 읽는 중입니다... 완료
```

```
...
```

```
다음 패키지를 업그레이드할 것입니다:
```

```
  apport apport-gtk cheese cheese-common libcheese-gtk25 libcheese8
```

```
  libcryptsetup12 libnetplan0 netplan.io python3-apport ...
```

```
14개 업그레이드, 0개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
```

```
1,253 k바이트 아카이브를 받아야 합니다.
```

```
이 작업 후 76.8 k바이트의 디스크 공간을 더 사용하게 됩니다.
```

```
계속 하시겠습니까? [Y/n]
```




LINUX 패키지 관리

□ 우분투 패키지 관리 명령(4)

- 특정 패키지 설치 또는 업그레이드: **apt-get install 패키지이름 [옵션]**
 - 하나 이상의 패키지를 설치하거나 업그레이드할 때 사용

<패키지 하나를 설치할 때>

```
linuxer@linuxer-PC:~$ sudo apt-get install netcat
```

<여러 패키지를 한 번에 설치할 때>

```
linuxer@linuxer-PC:~$ sudo apt-get install netcat goaccess
```

<설치할 때 업그레이드를 하지 않으려면>

```
linuxer@linuxer-PC:~$ sudo apt-get install netcat --no-upgrade
```

<새로운 패키지를 설치하지 않고 업그레이드만 하려면>

```
linuxer@linuxer-PC:~$ sudo apt-get install netcat --only-upgrade
```



LINUX 패키지 관리

□ 우분투 패키지 관리 명령(5)

- 패키지 삭제: **apt-get remove [옵션] 패키지이름**
 - 사용이 끝난 패키지는 주기적으로 삭제해서 용량 줄이는 습관이 필요

<패키지 하나를 삭제할 때>

```
linuxer@linuxer-PC:~$ sudo apt-get remove netcat
```

<여러 패키지를 한 번에 삭제할 때>

```
linuxer@linuxer-PC:~$ sudo apt-get remove netcat goaccess
```

<삭제할 때 설정 파일도 함께 삭제하려면>

```
linuxer@linuxer-PC:~$ sudo apt-get remove --purge netcat
```

또는

```
linuxer@linuxer-PC:~$ sudo apt-get purge netcat
```



LINUX 패키지 관리

□ 우분투 패키지 관리 명령(6)

▪ 패키지 자동 정리 및 삭제: **apt-get autoremove**

- 다른 패키지에 대한 종속성을 충족시키기 위해
자동으로 설치되어 더 이상 필요하지 않은 패키지를 제거하는 데 사용

```
linuxer@linuxer-PC:~$ sudo apt-get autoremove
```

▪ 디스크 공간 정리: **apt-get clean**

- 검색된 패키지 파일의 로컬 저장소를 삭제
- **/var/cache/apt/archives/** 및 **/var/cache/apt/archives/partial/**에
서 잠금 파일을 제외한 모든 것을 제거

```
linuxer@linuxer-PC:~$ sudo apt-get clean
```



LINUX 패키지 관리

□ 우분투 패키지 관리 명령(7)

- 패키지 내려받기: **apt-get download 패키지이름**

- 패키지를 설치하지 않고 단순히 내려받기만 할 때 사용

```
linuxer@linuxer-PC:~$ sudo apt-get download netcat
```

- 패키지 소스 코드 내려받기: **apt-get source 패키지이름**

- /etc/apt/sources.list 파일에 소스가 있는 URI가 정의되어 있어야 함

<소스 코드를 내려받기만 할 경우>

```
linuxer@linuxer-PC:~$ sudo apt-get --download-only source netcat
```

<소스 코드를 내려받고 압축을 푸는 경우>

```
linuxer@linuxer-PC:~$ sudo apt-get source netcat
```

<소스 코드를 내려받아 압축을 풀고 컴파일하는 경우>

```
linuxer@linuxer-PC:~$ sudo apt-get --compile source netcat
```



LINUX 패키지 관리

□ 데비안 패키지 관리 명령(1)

▪ 패키지 관리 명령: **dpkg** [옵션]

• 주요 옵션들

- I 패키지명 : 설치된 패키지의 **목록** 출력
- s 패키지명 : 설치된 패키지의 **상세 정보** 출력
- S 경로명 : 경로명이 포함된 패키지를 검색
- L 패키지명 : 패키지에서 설치된 파일의 목록 출력
- c .deb파일 : 지정된 .deb 파일의 내용을 출력
- i .deb파일 : 지정된 .deb 파일 **설치** (sudo 접근)
- r 패키지명 : 해당 패키지를 **삭제** (sudo 접근)
- P 패키지명 : 해당 패키지와 설정 정보를 모두 삭제 (sudo 접근)
- x .deb파일 디렉터리 : 해당 파일을 지정한 디렉터리에 압축 해제



LINUX 패키지 관리

□ 데비안 패키지 관리 명령(2)

- APT 명령과 달리 dpkg 명령은 의존성이 있는 패키지를 자동으로 설치하지 않기 때문에 사용자가 일일이 설치해야 함
- 따라서, 데비안 패키지들은 의존성 패키지를 별도로 묶어서 제공하기도 함

```
linuxer@linuxer-PC:~$ dpkg -l
linuxer@linuxer-PC:~$ dpkg -l zip
linuxer@linuxer-PC:~$ dpkg -s zip
linuxer@linuxer-PC:~$ dpkg -S /bin/ls
linuxer@linuxer-PC:~$ dpkg -L zip
linuxer@linuxer-PC:~$ dpkg -c netcat_1.10-41.1_all.deb
linuxer@linuxer-PC:~$ sudo dpkg -i netcat_1.10-41.1_all.deb (의존성오류발생)
linuxer@linuxer-PC:~$ sudo dpkg -r netcat (설치된 것만 삭제)
linuxer@linuxer-PC:~$ sudo dpkg -P netcat-traditional (설정 정보 모두 삭제)
linuxer@linuxer-PC:~$ dpkg -x netcat_1.10-41.1_all.deb netcat
```



LINUX 패키지 관리

□ 데비안 패키지 관리 명령(3)

- 의존성 오류 발생 시

해당 의존성 패키지를 먼저 다운로드와 설치를 한 후 해당 패키지 설치

- 아래 순서를 지켜서 설치

```
linuxer@linuxer-PC:~$ apt-get download netcat-traditional
```

```
linuxer@linuxer-PC:~$ sudo dpkg -i netcat-traditional_1.10-41.1ubuntu1_amd64.deb
```

```
linuxer@linuxer-PC:~$ sudo dpkg -i netcat_1.10-41.1_all.deb
```



LINUX 패키지 관리

□ 비주얼 모드 패키지 관리 명령(1)

▪ 패키지 관리 명령: **aptitude** [옵션] [서브 명령]

- 옵션이나 서브 명령 없이 실행할 경우 **curses**를 이용한 비주얼 모드로 동작
- 서버 명령

search 키워드 : 키워드를 검색하여 일치하는 패키지 목록 출력

update : 패키지 저장소를 업데이트

upgrade : 모든 패키지를 최신 버전으로 업그레이드

show 패키지명 : 패키지에 대한 자세한 정보 출력

download 패키지명 : 지정된 패키지를 다운로드

clean : 패키지 캐시 디렉터리에서 모든 패키지 파일 삭제

install 패키지명 : 해당 패키지를 설치

remove 패키지명 : 설치된 패키지만 삭제

purge 패키지명 : 설치된 패키지와 설정 파일 모두 삭제



LINUX 패키지 관리

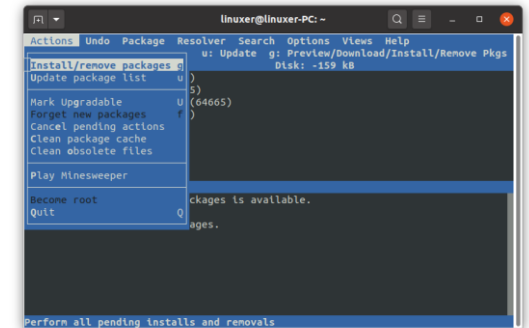
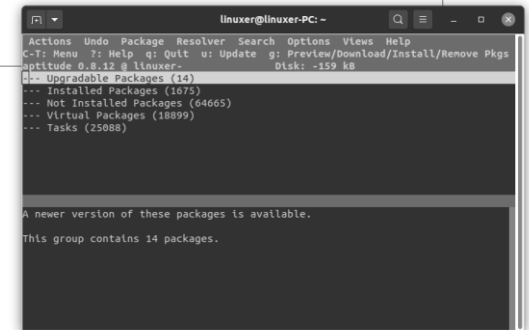
□ 비주얼 모드 패키지 관리 명령(2)

- 우분투에 aptitude 명령이 기본 설치되어 있지 않으니 먼저 설치 후 실행

```
linuxer@linuxer-PC:~$ sudo apt-get install aptitude
```

```
linuxer@linuxer-PC:~$ sudo aptitude
```

- 비주얼 모드에서는 마우스와 단축키 동작 함
 - ? 키 : 도움말 출력
 - q 키 : aptitude 종료
 - j, k, 화살표를 이용해서 메뉴 이동





LINUX 패키지 관리

□ 비주얼 모드 패키지 관리 명령(3)

▪ 서버명령을 이용한 aptitude 실행

<패키지 정보 업데이트>

```
linuxer@linuxer-PC:~$ sudo aptitude update
```

<패키지 검색>

```
linuxer@linuxer-PC:~$ aptitude search gnome
```

<패키지 정보 확인>

```
linuxer@linuxer-PC:~$ aptitude show gnome-clocks
```

<패키지 설치>

```
linuxer@linuxer-PC:~$ sudo aptitude install gnome-clocks
```

<패키지 삭제>


```
linuxer@linuxer-PC:~$ sudo aptitude remove gnome-clocks
```

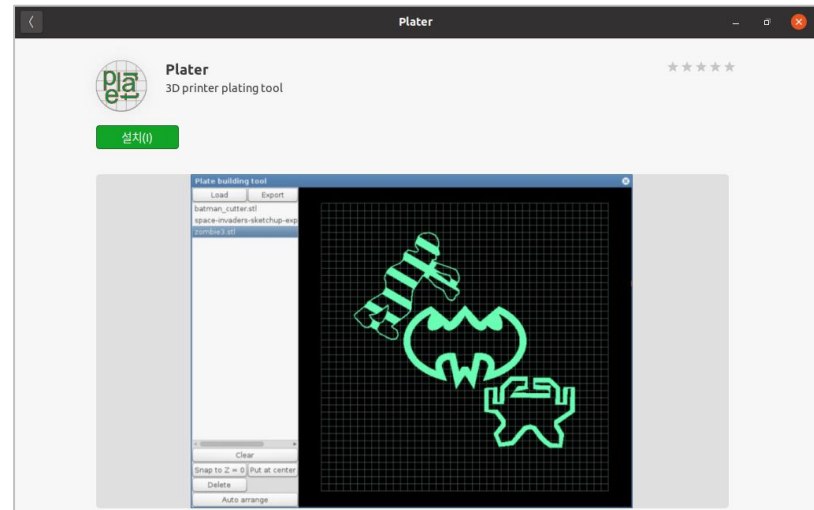
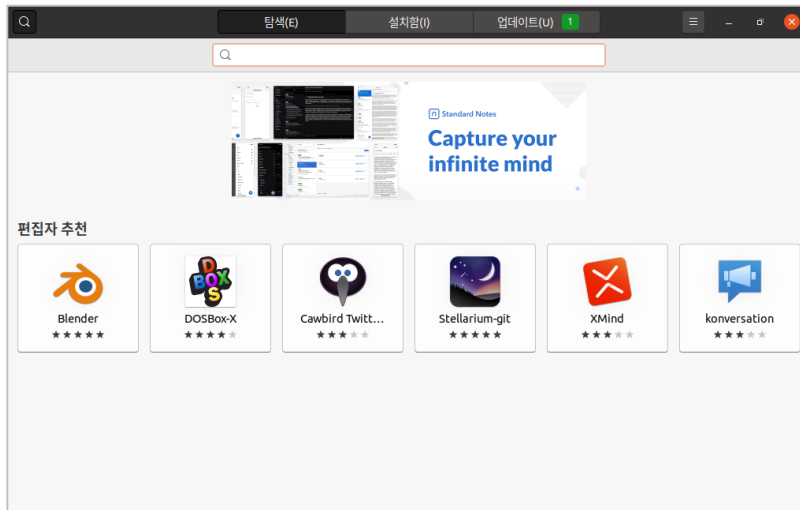


LINUX 패키지 관리

□ GUI 기반 패키지 관리

▪ 우분투 소프트웨어

- 가장 편리한 GUI 기반 패키지 관리 도구
- GNOME 바탕화면에 있는 "Unubtu Software" 스냅샷 실행 
- 인터넷 연결되어 있으면 필요한 패키지를 검색해서 설치/제거 가능





파일 아카이브와 압축

□ 파일 아카이브(File Archives) (1)

▪ 파일을 묶어서 하나의 파일로 만든 것

- 압축하지 않고 여러 파일과 디렉터리의 단순한 묶는 개념이었으나 최근에는 압축 알고리즘과 연결하여 압축도 진행함

- 이동식 저장장치에 보관하기 위해 사용된 명령: tar (tape archive)

- 형식: **tar** 기능[옵션] [아카이브 파일] 파일명

• 기능

c : 새로운 tar 파일 생성

t : tar 파일의 내용을 출력

x : tar 파일에서 원본 파일 추출

r : 새로운 파일 추가

u : 수정된 파일 업데이트

• 옵션

f : 파일 지정

v : 처리중인 파일의 정보 출력

p : 복구 시 원래 접근 권한 유지

j : bzip2로 압축하거나 해제

z : gzip으로 압축하거나 해제



파일 아카이브와 압축

□ 파일 아카이브(File Archives) (2)

▪ 자주 사용하는 tar 명령

<아카이브 생성>

```
linuxer@linuxer-PC:~$ tar cvf text.tar *.txt
```

<아카이브 내용 확인>

```
linuxer@linuxer-PC:~$ tar tvf text.tar
```

<아카이브 풀기 (디렉터리 위치에 주의)>

```
linuxer@linuxer-PC:~$ tar xvf text.tar
```

<아카이브 업데이트 (목록에 없거나 수정된 파일이 대상임)>

```
linuxer@linuxer-PC:~$ tar uvf text.tar
```

<아카이브에 파일 추가 (아카이브의 마지막에 추가됨)>

```
linuxer@linuxer-PC:~/info$ tar rvf text.tar *.dat
```



파일 아카이브와 압축

□ 파일 압축(File Compression) (1)

- 압축 알고리즘을 이용하여 대상 파일을 압축
- 파일이 압축되면 원본 파일 이름이 변경됨
- 디렉터리를 압축하려면 tar를 이용하여 아카이브 생성 후 압축
- 압축 명령: **gzip** [옵션] 파일명 (해당 파일명 뒤에 **.gz** 붙음)

<압축파일 생성>

```
linuxer@linuxer-PC:~$ gzip *.txt
```

<압축파일 내용 확인>

```
linuxer@linuxer-PC:~$ zcat list.txt.gz
```

<압축 해제>

```
linuxer@linuxer-PC:~$ gunzip *.gz
```



파일 아카이브와 압축

□ 파일 압축(File Compression) (2)

- 압축 명령: **bzip2** [옵션] 파일명 (해당 파일명 뒤에 **.bz2** 붙음)

<압축파일 생성>

```
linuxer@linuxer-PC:~$ bzip2 *.txt
```

<압축파일 내용 확인>

```
linuxer@linuxer-PC:~$ bzipcat list.txt.bz2
```

<압축 해제>

```
linuxer@linuxer-PC:~$ bunzip2 *.bz2
```