



제 3장 우분투 관리


✓ 3-1 파일 시스템

✓ 3-3 사용자관리

✓ 3-5 네트워크 관리

✓ 3-2 패키지 관리

✓ 3-4 반복 작업 자동화하기

 한국기술교육대학교

 컴퓨터공학부 박진우 책임기술연구원 2405호



패키지 관리

● 우분투 소프트웨어 패키지 사용하기

- ◆ GUI 툴을 사용하여 소프트웨어를 설치하려면 데스크탑 환경에서는 시냅틱 패키지 관리자를 사용하면 된다. 텍스트 기반으로는 aptitude 유틸리티를 사용한다.
- ◆ 데비안 계열 시스템은 dpkg 유틸리티를 사용한다.
- ◆ 데비안 패키지(.deb 확장자)에는 소프트웨어 파일만 저장되어 있는 것이 아니라 패키지에 대한 많은 정보가 포함되어 있다. 이러한 메타데이터는 소프트웨어의 의존성, 컴퓨터 아키텍처, 벤더, 크기, 라이선스, 설명문 등 다양한 정보를 가지고 있다.



패키지 관리

- aptitude 툴은 소프트웨어를 설치하거나 삭제할 때 아주 유용하다.
 - ◆ APT - 온라인 리포지토리에서 패키지를 다운로드 받고 설치하는 데 사용한다. APT 명령어(apt-get, apt-cache등)들은 로컬에서 패키지를 설치하는 데 사용할 수도 있지만, 정상적인 사용을 위해서는 온라인 환경에서 사용해야 한다.
 - ◆ dpkg - CD-ROM이나 다른 디스크 저장장치에 있는 .deb 파일을 제어하는 경우에 사용한다. dpkg 명령어는 시스템 소프트웨어에 대한 설정, 설치, 그리고 정보 획득을 위한 옵션들을 가지고 있다.
 - ◆ aptitude - aptitude는 커맨드라인 명령어로 온라인 리포지토리와의 작업을 위해 사용된다. aptitude 도구는 APT나 dpkg를 사용할 경우 사용자가 직접 해주어야 하는 번거로운 작업들을 자동화해주기 때문에 많이 사용되고 있다.



패키지 관리

- apt-cache stats 명령으로 사용 가능한 전체 패키지 수를 확인할 수 있다.

```
root@study: ~  
root@study:~# apt-cache stats  
전체 패키지 이름 : 71075 (1,422 k)  
전체 패키지 구조 : 109987 (4,839 k)  
일반 패키지 : 80894  
순수 가상 패키지 : 1824  
단일 가상 패키지 : 9570  
혼합 가상 패키지 : 2804  
빠짐 : 14895  
개별 버전 전체 : 91633 (7,331 k)  
개별 설명 전체 : 148312 (3,559 k)  
전체 의존성 : 674937/218963 (17.0 M)  
전체 버전/파일 관계 : 57992 (1,392 k)  
전체 설명/파일 관계 : 17240 (414 k)  
전체 제공 매핑 : 22254 (534 k)  
전체 패턴 문자열 : 156628 (3,462 k)  
전체 빈 용량 : 26.4 k  
차지하는 전체 용량 : 40.4 M  
Total buckets in PkgHashTable: 50503  
Unused: 12309  
Used: 38194  
Utilization: 75.6272%  
Average entries: 2.87969  
Longest: 15  
Shortest: 1  
Total buckets in GrpHashTable: 50503  
Unused: 12309  
Used: 38194  
Utilization: 75.6272%  
Average entries: 1.86089  
Longest: 8  
Shortest: 1
```




패키지 관리

● 소프트웨어 패키지 사용하기

- ◆ dpkg는 단일 deb 패키지 설치를 위한 매우 강력한 명령어지만, 패키지가 설치될 때 필요한 소프트웨어 의존성 문제에 대해서는 다루지는 못한다. 그에 반해 APT의 경우 패키지를 설치할 때 설정된 리포지토리를 참조하여 의존성 문제를 해결할 수 있다.

● APT 리포지토리 추가하기

- ◆ 데비안 기반의 배포판(린스 파이어, 잔드로스는, 우분투, 그리고 Mepis)에 외부 리포지토리를 추가하고 싶다면 /etc/apt/sources.list 파일에 해당 정보를 추가해주면 된다. 이 설정파일을 편집하기 위해서는 루트 퍼미션을 가지고 있어야 한다. APT는 어떤 패키지를 설치하거나 업데이트할 때 다른 프로그램이 의존하고 있는 패키지를 실수로 삭제하는 것을 방지하여 의존성을 보장 받을 수 있게 한다.



패키지 관리

\$ sudo vi /etc/apt/sources.list

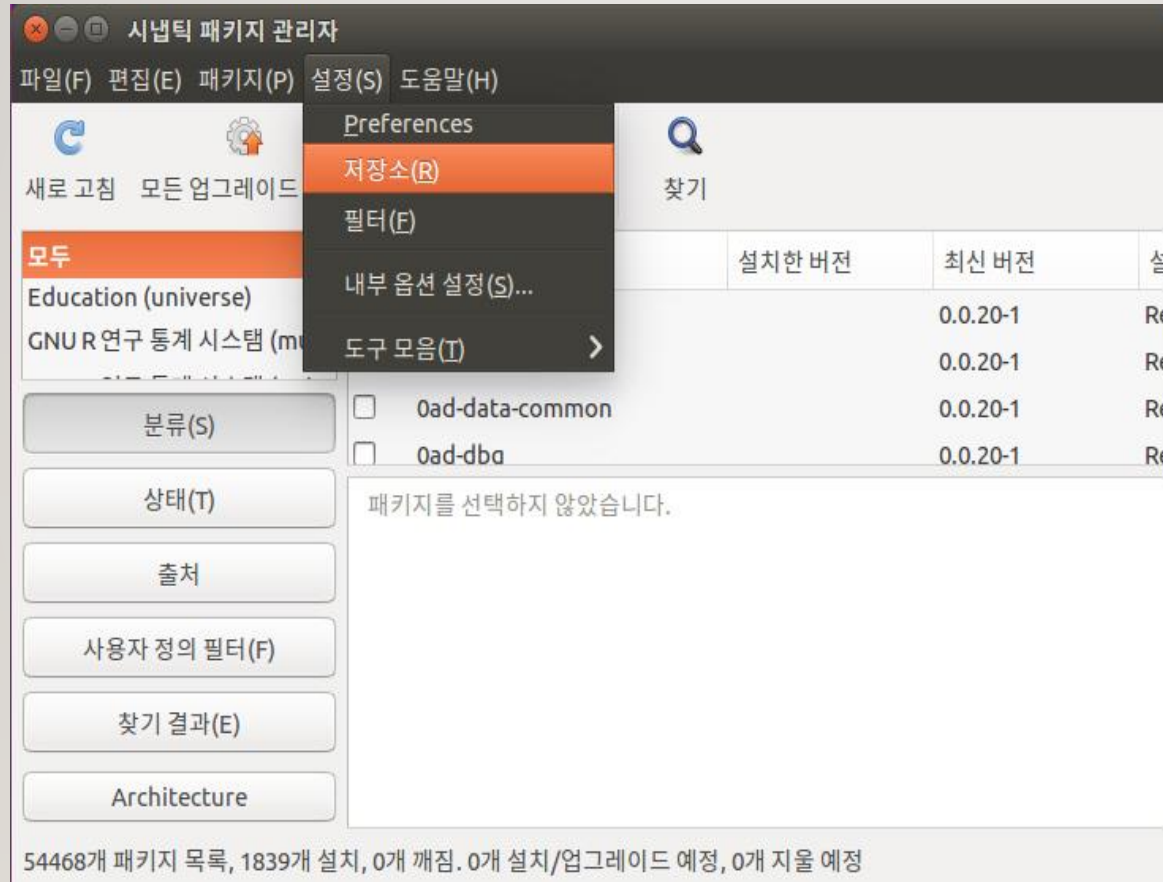
- sources.list 파일에는 패키지 유형, 웹 주소 (URL), distribution, 섹션(들) 항목을 입력한다.
- deb로 시작하는 라인(빌드된 패키지) 또는 deb-src로 시작되는 라인(소스 패키지)을 입력한 다음, 배포판에 대한 리포지토리의 URL 주소를 입력한다.

```
deb http://host/debian distribution section1 section2 section3
deb-src http://host/debian distribution section1 section2 section3
deb http://http.us.debian.org/debian stable main contrib non-free
deb-src http://http.us.debian.org/debian stable main
```



패키지 관리

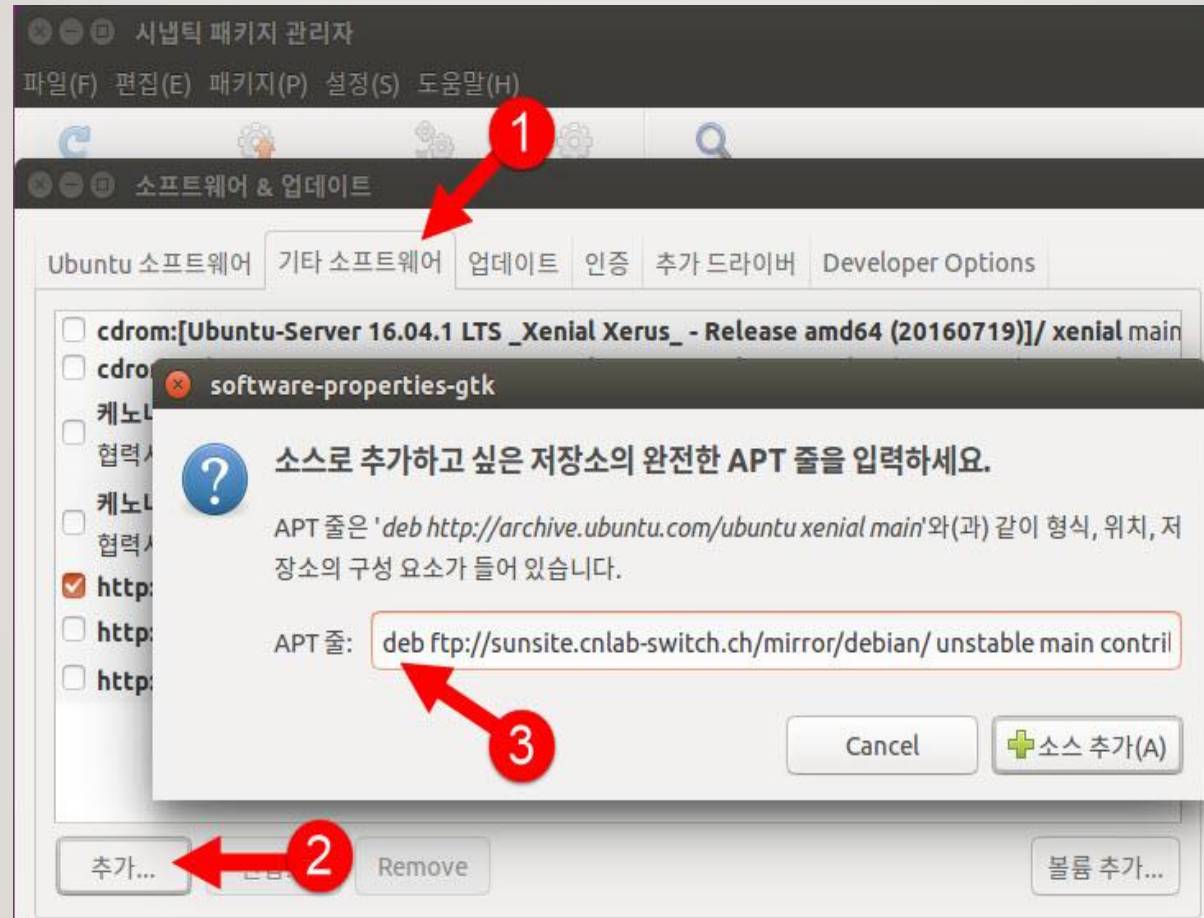
- GUI 방식으로는 시냅틱 꾸러미 관리자를 이용하는 방법
 - ◆ 먼저, 설정 -> 저장소를 선택한다.





패키지 관리

- ◆ 추가하려는 리포지토리를 입력한다.
- ◆ ※ sources.list 파일을 수정한후에는 꼭 aptitude update를 실행하십시오.





패키지 관리

● APT를 사용하여 소프트웨어 관리하기

- ◆ 실질적으로 APT는 dpkg와 함께 동작한다. 하지만, 필요한 소프트웨어의 검색, 다운로드, 설치, 업그레이드, 검사 등 대부분의 패키지 관리작업을 APT단독으로 가능하다.

APT 명령어	설 명
<code>sudo apt-get update</code>	etc/apt/sources.list를 참조해 사용할 수 있는 패키지 데이터베이스를 업데이트한다. <code>sources.list</code> 파일을 변경했다면 반드시 이 명령을 실행한다.
<code>apt-cache search <keyword></code>	패키지 데이터베이스에서 주어진 키워드를 대소문자 구분 없이 검색한다. 결과로 키워드를 포함하고 있는 패키지명과 설명이 출력된다.
<code>sudo apt-get install <package></code>	주어진 패키지명을 데이터베이스에서 찾을 수 있다면, 해당 패키지를 다운로드 받아 설치한다. APT 버전 0.6부터, 이 명령어는 자동적으로 gpg 키를 사용해 패키지의 신뢰성을 검증한다. 보다 자세한 내용은 다음을 참고한다. http://wiki.debian.org/SecureApt



패키지 관리

APT 명령어	설 명
<code>sudo apt-get -d install <package></code>	패키지를 설치하지 않고 /var/cache/apt/archives 디렉터리에 다운로드만 받는다.
<code>apt-cache show <package></code>	주어진 패키지명에 해당하는 소프트웨어에 대한 정보를 출력한다.
<code>sudo apt-get upgrade</code>	설치되어 있는 모든 패키지에 대한 최신 업데이트를 검사한 후, 다운로드받아 설치한다.
<code>sudo apt-get dist-upgrade</code>	전제 시스템을 새로운 버전으로 업그레이드 한다. 여기에는 패키지 삭제도 포함되어 있다. 이 방법은 시스템을 업데이트라는 방법으로는 잘 사용되지 않는다.
<code>sudo apt-get autoclean</code>	불완전하게 다운로드된 패키지나 오래된 패키지를 삭제하고자 할 때 실행한다. 부분적으로 디스크 공간을 절약할 수 있다.
<code>sudo apt-get clean</code>	디스크 공간 확보를 위해 /var/cache/apt/archives 디렉터리에 캐시되어 있는 패키지들을 모두 삭제한다.



패키지 관리

APT 명령어	설 명
<code>sudo apt-get --purge remove <package></code>	주어진 패키지와 그 설정파일을 삭제한다. --purge 옵션을 사용하면 설정파일은 보존된다.
<code>sudo apt-get -f install</code>	깨진 패키지를 위해 정상 여부를 검사한다. 이 명령어는 의존성 문제에 대한 메시지 수정을 시도한다.
<code>apt-config -V</code>	설치된 APT 유틸리티의 버전정보를 출력한다.
<code>sudo apt-key list</code>	APT가 알고 있는 gpg 키 목록을 출력한다.
<code>apt-cache stats</code>	설치된 모든 패키지에 대한 상태정보를 출력한다.
<code>apt-cache depends</code>	패키지가 설치되어 있는지 여부에 관계없이 패키지에 대한 의존성을 출력한다.
<code>apt-cache pkgnames</code>	시스템에 설치되어 있는 모든 패키지 목록을 보여준다.



패키지 관리

● 일반 사용자가 **sudo**를 사용할 수 있게 등록하기

- ◆ 일반 사용자가 관리자의 권한을 획득하기 위해 **sudo** 명령을 사용하는데 이 명령은 아무나 사용할 수 없다. 사용할 사용자를 등록하는 방법을 알아보도록 하겠다.
- ◆ **sudoer**에 관한 설정은 **/etc/sudoers** 파일에 들어있다. 이 파일을 수정을 할 때는 루트(관리자)로 로그인해서 **visudo** 명령을 내려야 한다.
- ◆ 그러면 **/etc/sudoers** 파일을 자동으로 불러오고 편집 후 저장할 때 오류를 검사해 준다. 일반 에디터로 수정을 하다가 오류가 있는 상태로 저장할 경우 꼼짝 못하게 되는 경우가 생기기 때문이다.
- ◆ **/etc/sudoers** 파일에서 다음 라인을 넣어준다. 주석 처리되어 있는 경우 주석 기호만 삭제하면 된다.



패키지 관리

- ◆ 이렇게 해 주면 "sudo" 그룹에 속한 사용자는 sudo 명령을 사용할 수 있다. sudo 명령 사용시 암호를 입력하지 않으려면
%sudo ALL=(ALL) NOPASSWD: ALL
- ◆ 입력하면 된다. ALL은 모든 명령을 실행시킬 수 있다는 뜻이고, NOPASSWD는 sudo 명령 사용시 암호를 입력하지 않아도 된다는 뜻입니다. **권장하지 않는다.**

```
root@study: ~  
GNU nano 2.5.3      파일: /etc/sudoers.tmp  
  
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$  
  
# Host alias specification  
  
# User alias specification  
  
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
  
# Members of the admin group may gain root privileges  
%admin   ALL=(ALL) ALL  
  
# Allow members of group sudo to execute any command  
%sudo    ALL=(ALL:ALL) ALL  
  
# See sudoers(5) for more information on "#include" directives:
```



패키지 관리

- ◆ 이제 sudo 그룹에 sudo를 이용할 사용자를 등록해 주면 된다.

```
root@study: ~  
root@study:~# usermod -a -G sudo jeong  
root@study:~#  
root@study:~# groups jeong  
jeong : jeong adm cdrom sudo dip plugdev lxd lpadmin sambashare  
root@study:~#  
root@study:~#
```

- ◆ 이와 같은 방법으로 일반 사용자를 추가로 지정할 수 있다. sudo 명령을 내릴 때 암호를 물어보는데 이때 입력하는 암호는 자신의 암호이다.



패키지 관리

● APT를 사용하여 패키지 업그레이드하기

- ◆ 시간이 지남에 따라, 기능이 추가되거나 기존에 가지고 있던 문제를 해결한 새로운 버전이 릴리즈되는 등 패키지는 지속적으로 변경된다.
- ◆ apt-get 명령어의 update 옵션을 사용하여, 현재 사용 중인 시스템의 패키지 업데이트 목록을 확인한다.
\$ sudo apt-get update
- ◆ 이 명령어는 현재 리포지토리에서 다운로드 가능한 새 버전의 패키지들을 찾아보고, 사용자의 시스템에 캐시되어 있는 패키지와 버전 목록을 새로운 정보로 업데이트 한다.
- ◆ 두 번째 단계로 apt-get 명령의 upgrade 옵션을 사용하면, 시스템에 설치되어 있는 패키지 중 새로운 버전이 있는 패키지는 모두 업그레이드된다.
\$ sudo apt-get upgrade
- ◆ 패키지를 업그레이드하기 위해서는 항상 패키지 목록을 최신 정보로 업데이트해야 하므로, 이 두 가지 명령어를 함께 사용한다.
\$ sudo apt-get update; sudo apt-get upgrade



패키지 관리

- APT를 사용하여 개별 패키지 업그레이드하기

- ◆ 개별 시스템 패키지의 설치 및 업그레이드는 `apt-get install <패키지명>` 명령을 사용하여 간단하게 처리할 수 있다.
- ◆ install 옵션을 사용하지만, 패키지가 설치되어 있는 경우라면 자동적으로 가장 최신의 버전으로 업데이트된다.

\$ **sudo apt-get install minicom**

- APT를 사용하여 패키지 삭제하기

- ◆ remove 옵션을 사용하면 시스템으로부터 패키지를 삭제할 수 있다
- ◆ 삭제하기 전에 삭제 여부를 확인하는 메시지가 출력

\$ **sudo apt-get remove minicom**



패키지 관리

- APT 패키지 캐시 정보 삭제하기

- ◆ 시스템을 설치하고 나면, 다운로드한 패키지를 /var/cache/apt에 캐시해 두는데, 이들 패키지가 다시 필요할 경우 다운로드 시간을 절약할 수 있다. 패키지 캐시는 결과적으로 많은 디스크 공간을 사용하게 된다.
- ◆ 캐시 데이터를 모두 삭제하고 싶다면 clean 옵션을 사용한다.

\$ sudo apt-get clean



패키지 관리

- **aptitude를 사용하여 소프트웨어 관리하기**

- ◆ aptitude 툴은 충분히 유연하면서도, 주요 패키지 작업 과정(패키지 설치나 업그레이드 전에 apt-get update를 실행하는 것과 같은)을 자동화하여 가능한 쉽게 작업할 수 있도록 해준다. 이러한 이유로 **가능하다면 커맨드라인에서는 aptitude 명령어를 사용할 것을 권장한다.**
- ◆ aptitude 패키지를 설치한다.
\$ **sudo apt-get install aptitude**



패키지 관리

- aptitude 프로그램의 옵션은 대부분 apt-get 명령어와 유사하다.

명 령	설 명
sudo aptitude	curses 인터페이스로 시작한다. Ctrl+t를 사용하면 메뉴에 접근할 수 있으며 q 키는 프로그램을 종료한다.
aptitude help	aptitude 명령어의 도움말을 출력한다.
aptitude search <keyword>	주어진 키워드와 일치하는 패키지들을 보여준다.
sudo aptitude update	APT 리포지토리들로부터 사용 가능한 패키지를 업데이트 한다.
sudo aptitude upgrade	모든 패키지를 최신 버전으로 업그레이드한다.
aptitude show <package>	패키지의 설치 여부에 관계없이 주어진 패키지에 대한 정보를 보여준다.
sudo aptitude download <Package>	주어진 패키지를 설치하지 않고 다운로드만 받는다.
sudo aptitude clean	/var/cache/apt/archives 디렉터리에 다운로드된 .deb 파일들을 모두 삭제한다.
sudo aptitude autoclean	/var/cache/apt/archives 디렉터리에 있는 오래된 .deb 파일들을 모두 삭제한다. 이 작업은 디스크 공간 낭비를 방지하면서 현재 캐시를 유지할 수 있는 방법이다.
sudo aptitude install <Package>	주어진 패키지를 시스템에 설치한다. 특정 버전을 선택하거나 와일드카드 문자를 사용하기 위한 여러 옵션이 있다.
sudo aptitude remove <package>	시스템으로부터 주어진 패키지를 삭제한다.
sudo aptitude dist-upgrade	모든 패키지를 가장 최신 버전으로 업그레이드하는데, 필요한 경우 패키지를 삭제하거나 추가적으로 설치한다.



패키지 관리

- 하나 이상의 패키지 설치

- ◆ aptitude에서는 패키지 선택을 위해 와일드카드를 사용할 수 있다.
aptitude 검색과 같이, minic 단어를 포함한 패키지를 설치하기 위해서는 키워드 앞에 ~n 검색어를 사용한다.

예로, minic 단어를 포함하고 있는 모든 패키지가 설치하려면 다음과 같이 한다.

\$ **sudo aptitude install ~nminic**

- aptitude를 사용하여 패키지 삭제하기

- ◆ remove 옵션과 삭제할 패키지명을 입력하여 삭제할 수 있다.

\$ sudo aptitude remove minicom