

11장

Package 관리 및 파일 압축하기

전체 내용

RPM Package
설치하기

Yum을 이용한
Package
설치하기

Ubuntu Package
설치하기

파일 Archive와
압축

1 – RPM Package 설치하기

CentOS에서의 Package 설치 개요

RPM Package의 특징과 구조

RPM Package 설치하기

Package 정보 검색하기

RPM Package 삭제하기

1 – RPM Package 설치하기

- CentOS에서의 Package 설치 개요

- 리눅스에서 소프트웨어(프로그램) 설치에는 2가지가 있다
 - Source Code로 설치
 - 소스코드를 묶어서 압축하여 배포하고 컴파일 하여 실행 파일로 만들어 설치한다
 - Package(Binary)로 설치
 - CentOS: RPM
 - Ubuntu: .deb 파일
- RPM
 - 간편하게 설치할 수 있도록 구성된 Package로서 간단한 명령어로 설치하고 정보를 확인하고 삭제할 수 있다
 - 다만 Package의 의존성에 문제가 있으면 관리자가 일일이 확인하고 순서대로 설치해야 하는 단점이 있다
 - Package들의 의존성 문제를 해결한 것이 Yum으로 설치하는 것이다

1 – RPM Package 설치하기

- RPM Package의 특징과 구조
 - RPM(Redhat Package Manager)는 레드햇 회사가 만든 패키지 관리 도구로서 파일을 쉽게 설치할 수 있다
 - RPM 특징
 - DVD 원본을 컴퓨터에 장착하여 mount하여 사용한다
 - Binary 파일로 구성되어 Compile이 필요 없다
 - Package 파일들이 관련 디렉터리로 바로 설치된다
 - 한 번에 설치된 Package 파일을 일괄적으로 삭제할 수 있다
 - 기존에 설치된 Package를 삭제하지 않고 바로 upgrad할 수 있다
 - Package의 설치 상태를 검증할 수 있다
 - Package에 대한 정보를 제공한다
 - RPM의 단점
 - Package의 의존성에 따라 관련 Package가 먼저 설치되어 있지 않으면 설치할 수 없다
 - 이 문제를 해결한 것이 Yum이다. Yum은 인터넷으로 연결하여 설치한다

1 – RPM Package 설치하기

- RPM Package의 특징과 구조
 - RPM Package 이름 구성이해하기

<u>audit-2.3.2-1.fc19.x86_64.rpm</u>				
패키지 이름	패키지 버전	패키지 릴리즈	아키텍처	확장자

- RPM 명령
 - RPM 버전 확인하기
rpm --version

```
[adminuser@centos ~]$ rpm --version
RPM 버전 - 4.8.0
[adminuser@centos ~]$
```

1 – RPM Package 설치하기

- RPM Package 설치하기
 - DVD mount하기
 - **mount -r -t iso9660 /dev/cdrom /mnt/cdrom**

```
[root@centos ~]# mkdir /mnt/cdrom
[root@centos ~]# mount -r -t iso9660 /dev/cdrom /mnt/cdrom
[root@centos ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@centos ~]# cd /mnt/cdrom
[root@centos cdrom]# ls
CentOS_BuildTag  RELEASE-NOTES-en-US.html  TRANS.TBL
EFI              RPM-GPG-KEY-CentOS-6      images
EULA             RPM-GPG-KEY-CentOS-Debug-6  isolinux
GPL             RPM-GPG-KEY-CentOS-Security-6  repodata
Packages        RPM-GPG-KEY-CentOS-Testing-6
[root@centos cdrom]# cd Packages/
```

`rpm -i [설치 옵션] 패키지명`

기능 RPM 패키지를 설치한다.

옵션 -h : 해시 기호(#)를 출력한다.

-v : 설치 과정에 대한 메시지를 출력한다.

--replacefiles : 이미 설치된 다른 패키지의 파일을 덮어쓰면서라도 패키지를 강제로 설치한다.

--replacepkgs : 패키지가 이미 설치되어 있어도 다시 설치한다.

--test : 설치하지는 않고 충돌 사항이 있는지만 점검하고 보고한다.

1 – RPM Package 설치하기

- RPM Package 설치하기
 - RPM Package 설치와 의존성
 - **rpm -ivh** xterm-293-1.gc19.x86_64.rpm
 - 의존성에 문제가 있어서 설치 실패함

`rpm -i [설치 옵션] 패키지명`

기능 RPM 패키지를 설치한다.

옵션 `-h` : 해시 기호(#)를 출력한다.

`-v` : 설치 과정에 대한 메시지를 출력한다.

`--replacefiles` : 이미 설치된 다른 패키지의 파일을 덮어쓰면서라도 패키지를 강제로 설치한다

`--replacepkgs` : 패키지가 이미 설치되어 있어도 다시 설치한다.

`--test` : 설치하지는 않고 충돌 사항이 있는지만 점검하고 보고한다.

1 – RPM Package 설치하기

- Package 정보 검색하기

rpm -q [질의 옵션]

기능 RPM 패키지의 정보를 검색한다.

옵션

- a : 전체 패키지 목록을 출력한다.
- f 파일명 : 파일명을 포함한 패키지명을 출력한다.
- p 패키지명 : 패키지명으로 지정한 패키지의 상세 정보를 출력한다.
- i : 자세한 정보를 출력한다.
- R : 의존하는 패키지의 목록을 출력한다.
- l : 패키지 내의 파일을 출력한다.
- s : 패키지 안에 들어 있는 파일의 상태를 출력한다.
- d : 문서 파일만 출력한다.
- c : 설정 파일만 출력한다.

1 – RPM Package 설치하기

- Package 정보 검색하기
 - 전체 Package 목록 출력하기: **-qa** 옵션
 - **rpm -qa**
 - 특정 Package 설치 여부 확인하기: **-qa | grep** 이름
 - **rpm -qa | grep bash** (##강추)
 - 특정 파일을 포함한 Package 확인하기: **-qf** 옵션
 - **rpm -qf ls**
- Package 상세 정보 확인하기: **-qi** 옵션, **-qip** 옵션, **-qif** 옵션
 - Bash 패키지에 대한 상세 정보 확인
rpm -qi bash
rpm -qi /usr/bin/ls (##패키지 이름이 아니기 때문)
 - 해당 파일(file)이 속한 패키지의 상세 정보 출력
rpm -qif /usr/bin/ls

1 - Linux 설치 전 준비사항

- Package 정보 검색하기
 - Package가 설치한 파일 목록 확인하기: **-ql** 옵션
 - **rpm -ql bash**
 - 이미 설치된 Package의 의존성 확인하기: **-qR** 옵션
 - **rpm -qR gedit**
- RPM Package 삭제하기
 - **rpm -e 패키지명**
 - **rpm -qa**
rpm -e gnote-0.6.3-3.el6.x86_64
rpm -qa gnote-0.6.3-3.el6.x86_64

2 - Yum을 이용한 Package 설치하기

Yum의 특징

Yum Package 목록 확인하기

Update 가능한 Package 목록 확인하기

Package 설치하기

Package Update하기

Package 정보 검색하기

Package 삭제하기

2 - Yum을 이용한 Package 설치하기

- Yum(Yellowdog Updater Modified)의 특징
 - Yum은 RPM 기반의 Package 관리도구로 rpm 명령의 패키지를 설치할 때 가장 어려운 점이었던 패키지 의존성 문제를 자동으로 해결함
 - 일반적으로 인터넷에 연결되어 있어야 한다
 - 패키지를 분석하여 패키지의 의존성을 해결하고 원격 자동업데이트와 설치를 할 수 있다
 - 설치 가능한 패키지에 대한 정보가 담긴 저장소(repository)를 가지고 있다
 - Repository의 정보에 의거하여 패키지를 관리한다

yum	
기능	rpm 기반의 패키지를 자동으로 설치한다.
형식	yum [옵션] [명령] [패키지명]
옵션	-h : 도움말을 출력한다. -y : 설치 과정의 모든 질문에 yes로 대답한다. -v : 자세한 메시지를 출력한다.
명령	install : 패키지를 설치한다. update : 패키지를 업데이트한다. check-update : 패키지를 확인한다. remove : 패키지를 삭제한다. list : 패키지 목록을 확인한다. info : 패키지 정보를 확인한다.
사용 예	yum install xterm yum list all

2 - Yum을 이용한 Package 설치하기

- Yum Package 목록 확인하기

[표 9-1] yum list 명령의 옵션

형식	기능
yum list all	이미 설치되었거나 설치 가능한 모든 패키지 목록을 출력한다.
yum list available	yum 저장소에서 설치 가능한 모든 패키지 목록을 출력한다.
yum list updates	yum 저장소에서 업데이트 가능한 패키지 목록을 출력한다.
yum list installed	이미 설치된 패키지 목록을 출력한다.
yum list installed 패키지명	지정한 패키지가 설치되어 있는지 확인한다.

- Yum Repository에서 설치 가능한 Package 목록 확인하기

- **yum list available | more**
- **yum list openssh**

```
Installed Packages
openssh.x86_64      5.3p1-104.el6      @anaconda-0
Available Packages
openssh.x86_64      5.3p1-104.el6_6.1  updates
[root@centos adminuser]#
```

- Yum Repository의 Package 중에서 Update 가능한 목록만 확인하기
 - **yum list updates**

2 - Yum을 이용한 Package 설치하기

- Package 설치하기
 - **yum install 패키지명 -y**
 - **yum install autotrace -y**
 - **yum install firefox -y**
- 시스템에 이미 설치된 Package 목록 확인하기
 - **yum list installed**
 - **yum list installed | grep ssh**
 - **rpm -qa | grep ssh**
 - **ps -ef | grep ssh** (##실행 중인 프로세스 찾기)
 - **find / -name samba** (##원하는 파일 찾기)
- Update 가능한 Package 목록 확인하기
 - **yum check-update** (=yum list updates)

2 - Yum을 이용한 Package 설치하기

- Package Update하기
 - 이미 설치된 특정한 Package만을 업데이트할 때는
 - **yum update 패키지명**
 - **yum update mysql**
 - yum check-update에서 나타난 모든 것을 업데이트하기
 - **yum update** (##강추)
 - 업데이트할 때도 설치 과정과 동일하게 의존 관계의 패키지가 있을 경우 함께 업데이트된다
- Package 정보 검색하기
 - **yum info 패키지명**
 - **yum info autotrace**
- Package 삭제하기
 - **yum remove 패키지명**

2 - Yum을 이용한 Package 설치하기

- 비슷한 이름의 Package 찾기: **yum search** 패키지명
 - Package의 정확한 이름을 찾을 때
 - **yum list** 패키지명
 - Package의 정확한 이름을 모르고 비슷한 이름으로 찾을 때
 - **yum search vsftpd**
- 특정한 파일이 어느 패키지 소속인지 확인하기
 - **yum provides** /etc/httpd/conf/httpd.conf

2 - Yum을 이용한 Package 설치하기

- 설치된 그룹 및 설치할 그룹(패키지) 확인하기
 - **yum grouplist | less**
- 필요한 그룹 설치하기
 - **yum -y groupinstall "Base"**
 - **yum groupinstall "Desktop" "X Window System" "Fonts"**
- 각 그룹에 대한 정보를 보려면
 - **yum groupinfo "Base"**
- 각 그룹만 Update하기
 - **yum groupupdate 'DNS Name Server'**
- 특정한 그룹 제거하기
 - **yum groupremove 'DNS Name Server'**

2 - Yum을 이용한 Package 설치하기

- Yum Repository 확인
 - 사용 가능한 것
yum repolist
yum repolist enabled
 - 사용할 수 없는 것
yum repolist disabled
 - 모든 Yum Repository 확인
 - **yum repolist all**

2 - Yum을 이용한 Package 설치하기

- 특정한 Yum Repository에서 Package 설치하기: **epel**
 - EPEL: Extra Packages for Enterprise Linux
 - EPEL is a Fedora Special Interest Group that creates, maintains, and manages a high quality set of additional packages for Enterprise Linux, including, but not limited to, Red Hat Enterprise Linux (RHEL), CentOS and Scientific Linux (SL), Oracle Linux (OL).
 - epel-release 검색하기
yum list epel-release
 - epel-release 설치하기
yum install epel-release
 - 사용 가능한 Repository 목록 확인하기
yum repolist

```
[root@centos1 ~]# yum repolist enabled
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
* base: ftp.neowiz.com
* epel: mirror.rise.ph
* extras: ftp.neowiz.com
* updates: ftp.neowiz.com
```

<u>repo id</u>	repo name
base	CentOS-6 - Base
<u>epel</u>	Extra Packages for Enterprise Linux 6 - x86_64

2 - Yum을 이용한 Package 설치하기

- 특정한 Yum Repository에서 Package 설치하기: **epel**

- epel에 있는 Package 종류 확인하기

yum --enablerepo=epel list all

- php로 시작하는 package 검색하기

yum --enablerepo=epel search php

yum --enablerepo=epel list all | grep ^php

- phpmyadmin 설치하기

yum --enablerepo=epel install phpmyadmin

- phpmyadmin 설치 여부 확인하기

yum list installed | grep phpMyAdmin

```
[root@centos1 ~]# yum list installed | grep phpMyAdmin
phpMyAdmin.noarch      4.0.10.10-1.el6 @epel
[root@centos1 ~]#
```

2 - Yum을 이용한 Package 설치하기

- 특정한 Yum Repository에서 Package 설치하기: **RPMForge**
 - 사용하고 있는 Linux Server Version 확인하기

uname -a

```
[root@centos1 ~]# uname -a
Linux centos1 2.6.32-504.el6.x86_64 #1 SMP Wed Oct 15 04:27:16 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
[root@centos1 ~]#
```

- 32 bit system: i686 i686 i386 GNU/Linux
 - 64 bit server: x86_64 x86_64 x86_64 GNU/Linux.
- RPMForge Repository in RHEL/CentOS 설치하기
 - 소스 파일 다운로드하기
wget http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
 - 설치하기
rpm -Uvh rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
 - The RPMForge repository will be installed under **/etc/yum.repod** directory as a file **rpmforge.repo**

2 - Yum을 이용한 Package 설치하기

- 특정한 Yum Repository에서 Package 설치하기: **RPMForge**
 - RPMForge Repository Key 가져오기(import)
 - DAG's GPG key를 다운로드하기
wget http://dag.wieers.com/rpm/packages/RPM-GPG-KEY.dag.txt
 - DAG's GPG key 설치하기
rpm --import RPM-GPG-KEY.dag.txt
 - RPMForge Repository를 사용하여 Package 설치하기
 - 사용 가능한 Repository 목록 확인하기
yum repolist enabled
 - RPMForge에 있는 모든 Package 확인하기
yum --enablerepo=rpmforge list all
 - tftp 패키지 검색하기
yum --enablerepo=rpmforge search tftp
yum --enablerepo=rpmforge list all | grep ^tftp
 - tftp-server 설치하기
yum --enablerepo=rpmforge install tftp-server

2 - Yum을 이용한 Package 설치하기

- 특정한 Yum Repository에서 Package 설치하기: **RPMForge**
 - RPMForge Repository 더 이상 사용하지 않기

- **vi /etc/yum.repos.d/rpmforge.repo**

여기서 enabled=1" 에서 "**enabled=0**로 변경한다

```
### Name: RPMforge RPM Repository for RHEL 7 - dag
### URL: http://rpmforge.net/
[rpmforge]
name = RHEL $releasever - RPMforge.net - dag
baseurl = http://apt.sw.be/redhat/el7/en/$basearch/rpmforge
mirrorlist = http://mirrorlist.repoforge.org/el7/mirrors-rpmforge
#mirrorlist = file:///etc/yum.repos.d/mirrors-rpmforge
enabled = 0
protect = 0
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmforge-dag
gpgcheck = 1
```

- RPMForge가 더 이상 사용하지 않는 지 확인하기
yum repolist enabled

2 - Yum을 이용한 Package 설치하기

- View History of Yum
 - **yum history**
- Clean Yum Cache
 - By default yum keeps all the repository enabled package data in **/var/cache/yum/** with each sub-directory
 - To clean all cached files from enabled repository, you need to run the following command regularly to clean up all the cache and make sure that **there is nothing unnecessary space is using**
 - **yum clean all**

3 – Ubuntu Package 설치하기

APT 명령으로 Package 관리하기

Dpkg 명령어로 Package 관리하기

3 - Ubuntu Package 설치하기

- APT(Advanced Package Tool)의 개요
 - **apt-cache**: APT cache(Package Database)에서 정보를 검색하여 출력
 - **apt-get**: Package 저장소를 업데이트하고 Package를 설치하거나 제거할 수 있다

apt-cache		
기능	APT 캐시에 질의하여 여러 가지 정보를 검색한다.	
형식	apt-cache [옵션] 서브 명령	
옵션	<ul style="list-style-type: none">-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.
사용 예	pkgnames	사용 가능한 모든 패키지의 이름을 출력한다.
	apt-cache stats apt-cache show vsftpd apt-cache search vsftpd	

3 - Ubuntu Package 설치하기

- apt-cache 명령어 사용
 - Apt cache 통계 정보 보기
 - **apt-cache stats**
 - 사용 가능한 Package 이름 보기
 - **apt-cache pkgnames**
 - Package 이름 검색하기
 - **apt-cache search** vsftpd
 - 특정한 Package에 대한 정보 확인하기
 - **apt-cache show** vsftpd
 - 특정한 Package의 의존성 검사하기
 - **apt-cache showpkg** vsftpd

3 - Ubuntu Package 설치하기

- apt-get 명령어 사용

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

- Package 정보 update하기: **apt-get update**

- 이 명령어는 새로운 Package 정보를 가져와서 APT cache를 수정하는 것이다
- **sudo apt-get update**

3 - Ubuntu Package 설치하기

- apt-get 명령어 사용
 - Package upgrade하기: **apt-get upgrade**
 - 현재 설치되어 있는 모든 Package 중에서 새로운 버전이 있는 것을 모두 업그레이드한다
 - **sudo apt-get upgrade**
 - 특정한 Package 설치 및 upgrade하기: **apt-get install 패키지명**
 - 하나 이상의 Package를 설치하거나 업그레이드한다
 - **sudo apt-get install netcat**
 - **sudo apt-get install nethogs goaccess**
 - Package를 설치할 때 업그레이드를 하지 않으려면
 - **sudo apt-get install netcat --no-upgrade**
 - 새로운 Package를 설치하지 않고 Upgrade만 할 때
 - **sudo apt-get install netcat --only-upgrade**

3 - Ubuntu Package 설치하기

- apt-get 명령어 사용
 - Package 삭제하기: **apt-get remove**
 - Remove는 삭제할 때 해당 패키지의 설정 파일을 남겨둔다. 이는 나중에 같은 패키지를 다시 설치할 때 재활용하기 위함이다
 - **sudo apt-get remove netcat**
 - Purge는 삭제할 때 해당 패키지의 설정 파일을 모두 삭제한다
 - **sudo apt-get purge netcat**
 - Disk 공간 정리하기: **apt-get clean**
 - 검색했거나 내려받은 패키지 파일들을 삭제하고 디스크 공간을 정리한다
 - **sudo apt-get clean**

3 - Ubuntu Package 설치하기

- apt-get 명령어 사용

- Package 자동 정리 및 삭제하기: **apt-get autoremove**

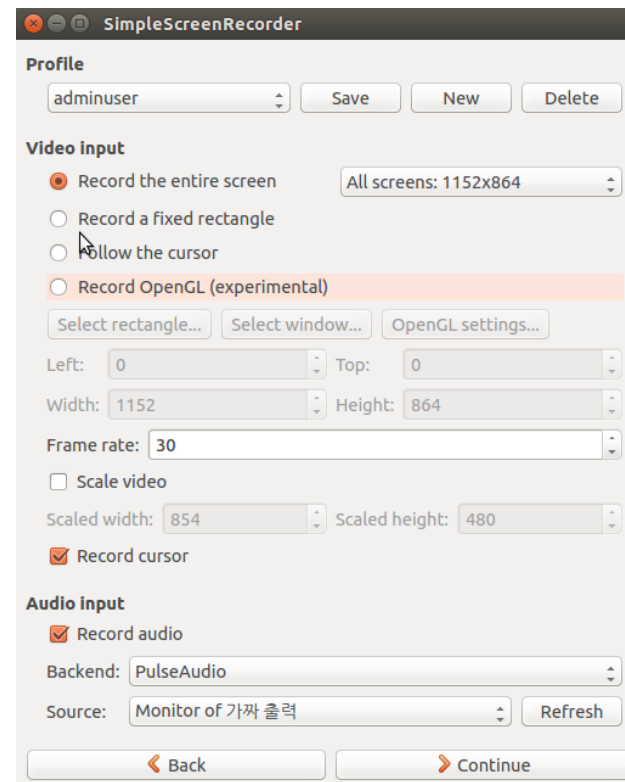
- 새로운 패키지를 설치할 때 "더 이상 필요 없는 패키지가 있다" 라는 메시지가 나오는 경우, 이런 것들을 자동으로 정리해 주는 것이다

- **sudo apt-get autoremove**

```
adminuser@linux200:~$ sudo apt-get install nfs-common nfs-kernel-server rpcbind
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
rpcbind 패키지는 이미 최신 버전입니다.
rpcbind 패키지 수동설치로 지정합니다.
nfs-common 패키지는 이미 최신 버전입니다.
nfs-kernel-server 패키지는 이미 최신 버전입니다.
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
  cfengine2 fai-client fai-setup-storage libdevmapper-event1.02.1
  liblinux-lvm-perl libparse-recdescent-perl live-boot live-boot-doc
  live-boot-initramfs-tools live-tools lvm2 watershed
Use 'apt-get autoremove' to remove them.
0개 업그레이드, 0개 새로 설치, 0개 제거 및 312개 업그레이드 안 함.
```


3 - Ubuntu Package 설치하기

- 특정한 Apt-Get Repository에서 Package 설치하기
 - **sudo add-apt-repository ppa:maarten-baert/simplescreenrecorder**
 - **sudo apt-get update**
 - **sudo apt-get install simplescreenrecorder**



3 - Ubuntu Package 설치하기

- Dpkg 명령어로 Package 관리하기
 - CentOS의 yum과 비슷한 것이 Ubuntu의 APT이다
 - CentOS의 RPM과 비슷한 것이 Ubuntu의 Dpkg이다

dpkg

기능 데비안의 패키지 관리 명령이다.

형식 dpkg [옵션] 파일명 또는 패키지명

옵션 -l : 설치된 패키지의 목록을 출력한다.
 -l 패키지명 : 패키지의 설치 상태를 출력한다.
 -s 패키지명 : 패키지의 상세 정보를 출력한다.
 -S 경로명 : 경로명이 포함된 패키지를 검색한다.
 -L 패키지명 : 패키지가 설치된 파일의 목록을 출력한다.
 -c .deb 파일명 : 지정한 .deb 파일의 내용을 출력한다.
 -i .deb 파일 : 해당 파일을 설치한다(sudo).
 -r 패키지명 : 해당 패키지를 삭제한다(sudo).
 -P 패키지명 : 해당 패키지와 설정 정보를 모두 삭제한다(sudo).
 -x .deb 파일 디렉터리 : 해당 파일을 지정한 디렉터리에 풀어 놓는다.

사용 예 dpkg -l

dpkg -s netcat

dpkg -S /bin/ls

sudo dpkg -i netcat_1.10-40_all.deb

3 - Ubuntu Package 설치하기

- Dpkg 명령어로 Package 관리하기
 - Package 목록 보기
 - **dpkg -l**
 - **dpkg -l zip** (##zip 패키지)
 - Package가 설치한 파일의 목록 검색하기
 - **dpkg -L zip**
 - Package 상세 정보 보기
 - **dpkg -s zip**
 - 특정한 파일이 포함된 패키지를 검색하기
 - **dpkg -S /bin/l**s

3 - Ubuntu Package 설치하기

- Dpkg 명령어로 Package 관리하기
 - Package 파일(.deb)이 가지고 있는 내용을 보기
 - **sudo apt-get download** netcat (##먼저 패키지를 다운로드만 한다)
 - **dpkg -c netcat_1.10-40_all.deb**
 - 다운로드 받은 Package 파일(.deb)을 설치하기 (root 권한 필요)
 - **sudo dpkg -i netcat_1.10-40_all.deb** (##강추)
 - **apt-get download** netcat-traditional
 - **sudo dpkg -i netcat-traditional_1.10-40_i386.deb**
 - Package 삭제하기
 - -r은 설치된 패키지만 삭제하고, -P 옵션은 패키지와 설정 정보를 모두 삭제
 - **sudo dpkg -r netcat**
 - **sudo dpkg -P netcat-traditional**

4 – 파일 Archive와 압축

파일 Archive하기

파일 압축하기

4 – 파일 Archive와 압축

- 파일 Archive하기

- Archive 뜻

- 새 봄을 맞이하여 주부가 겨울 옷을 보관함에 넣어서 따로 보관하는 것
 - 겨울 옷을 보관할 때 부피가 너무 크면 압축하여 보관한다
 - 다시 겨울이 오면 압축하여 보관한 것을 꺼내어 사용한다
 - 예: email archiving 기술

- 파일 Archive

- 여러 개의 파일을 묶어서 하나로 만든 것이다
 - 이 때 동시에 압축을 진행하기도 한다

- **Tar**(Tape Archive) 명령어

- 다시 시스템과 파일을 주고 받거나 백업을 하기 위해서 여러 개의 파일이나 디렉토리를 하나의 Archive file로 만들거나 또는 기존의 Archive에서 파일을 추출하기 위해 사용하는 명령어이다

4 – 파일 Archive와 압축

- 파일 Archive하기

- Archive 생성하기

- 디렉터리에 있는 모든 파일을 하나의 파일로 만든다

- **cd** /home/

- **tar cvf** file.tar **dir1**

- **ls -l** file.tar

- C: create

- V: 처리중인 파일의 이름을 화면에 상세히 출력

- F: archive 파일을 사용하겠다

- Archive 파일 내용 보기:

- **tar tvf** file.tar

- T: table of contents

4 – 파일 Archive와 압축

- 파일 Archive하기

- Archive 파일 풀기

- **mkdir** backup_dir1/
mv file.tar **backup_dir1/**
cd backup_dir1/
ls
tar xvf file.tar
ls **dir1/** (##여기에 모든 파일이 풀린 것이 확인된다)
 - X: extract

- Archive 업데이트 하기: **tar uvf** file.tar directory

- Archive 한 후에 파일이 수정, 생성 및 삭제된 경우에 사용한다

4 – 파일 Archive와 압축

- 파일 Archive하기

- Archive를 하면서 동시에 압축하기-1

- **cd /home/**
su root
tar cvzf file.tar.gz adminuser/
ls -l file.tar.gz
 - Gzip으로 압축한 것이다

- Archive를 하면서 동시에 압축하기-2

- **tar cvjf file.tar.bz2 adminuser/**
ls -l file.tar.bz2
 - bzip2로 압축한 것이다
 - Gzip에 비해 압축률은 좋지만 속도는 느리다

```
[root@centos home]# ls -l file*  
-rw-r--r--. 1 root root 195989 Jun  9 22:19 file.tar.bz2  
-rw-r--r--. 1 root root 275384 Jun  9 22:17 file.tar.gz
```

4 – 파일 Archive와 압축

- 파일 Archive하기
 - ~.tar.gz 파일을 압축해제하기 (tar xvzf 파일이름)
 - **mkdir uncompressed/**
 - **mv file.tar.gz uncompressed/**
 - **cd uncompressed/**
 - **ls -l file.tar.gz**
 - **tar xvzf file.tar.gz (##강추)**
 - ~.tar.bz2 파일 압축해제하기
 - **tar xvjf file.tar.bz2**

4 – 파일 Archive와 압축

- 파일 압축하기

- Gzip으로 파일 압축하기: **gzip**

- **gzip** file.tar (##file.tar은 非압축파일 / gzip하면 압축된 file.tar.gz 생성된다)

- Gzip으로 압축된 파일에 대한 정보 확인하기

- **gzip -l** file.tar.gz

- Gzip으로 압축된 파일 내용 보기: **zcat**

- **zcat** file.tar.gz

- Gzip으로 압축된 파일 풀기: **gunzip**

- Tar으로 Archive하면서 동시에 압축한 파일(file.tar.gz)을 푼다

- **gunzip** file.tar.gz (## 이렇게 하면 file.tar 파일이 생긴다)

- 풀린 압축 파일의 Archive를 해제한다

- mkdir** backup_dir2/

- mv** file.tar **backup_dir2/**

- cd** backup_dir2/

- ls

- tar xvf** file.tar

- ls** dir2/ (##여기에 모든 파일이 풀린다)

4 – 파일 Archive와 압축

- 파일 압축하기

- Bzip2는 gzip에 비해 압축률은 좋지만 속도가 약간 느리다
- bzip2로 파일 압축하기: **bzip2**
 - **bzip2** file.tar
- bzip2로 압축된 파일 내용 보기: **bzcat**
 - **bzcat** file.tar.**bz2**
- bzip2로 압축된 파일 풀기: **bunzip2**
 - Tar으로 Archive하면서 동시에 압축한 파일(file.tar.**bz2**)을 푼다
 - **bunzip2** file.tar.**bz2**
 - 풀린 압축 파일의 Archive를 해제한다
mkdir backup_dir3/
mv file.tar **backup_dir3/**
cd backup_dir3/
ls
tar xvf file.tar
ls dir3/ (##여기에 모든 파일이 풀린다)