"In return for an increase in my allowance, I can offer you free unlimited in-home computer tech support."

A **Linux distribution** (often abbreviated as **distro**) is an operating system made from a software collection that is based upon the Linux kernel and, often, a package management system.

Linux Package Management

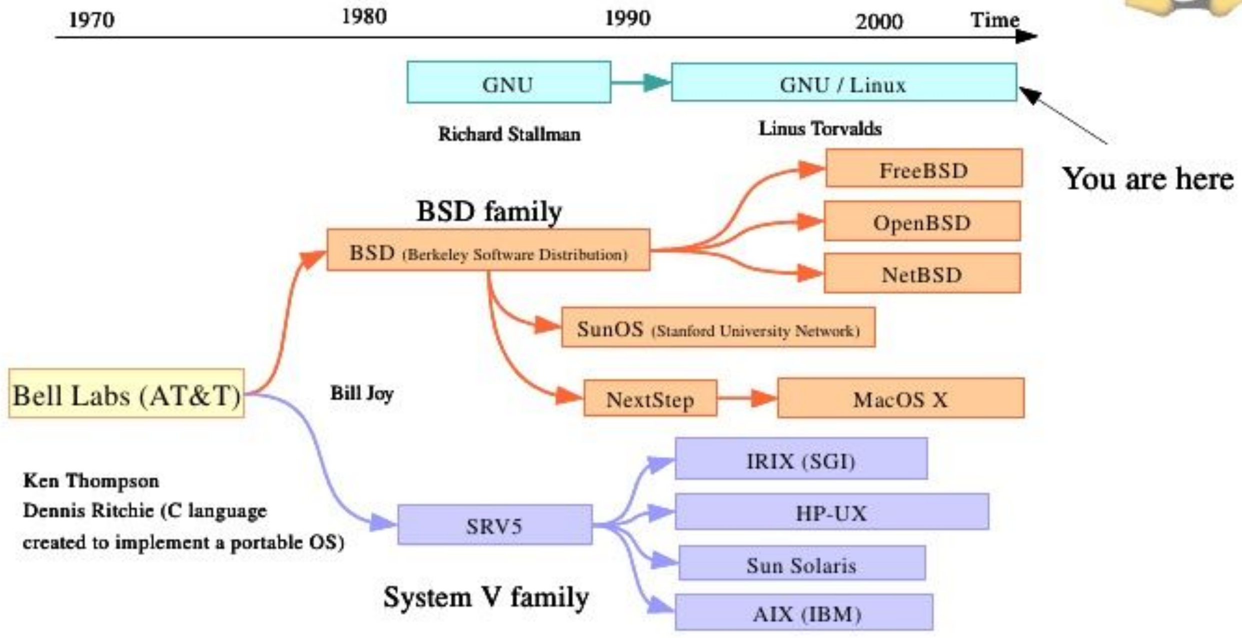
Source: geek-university.com

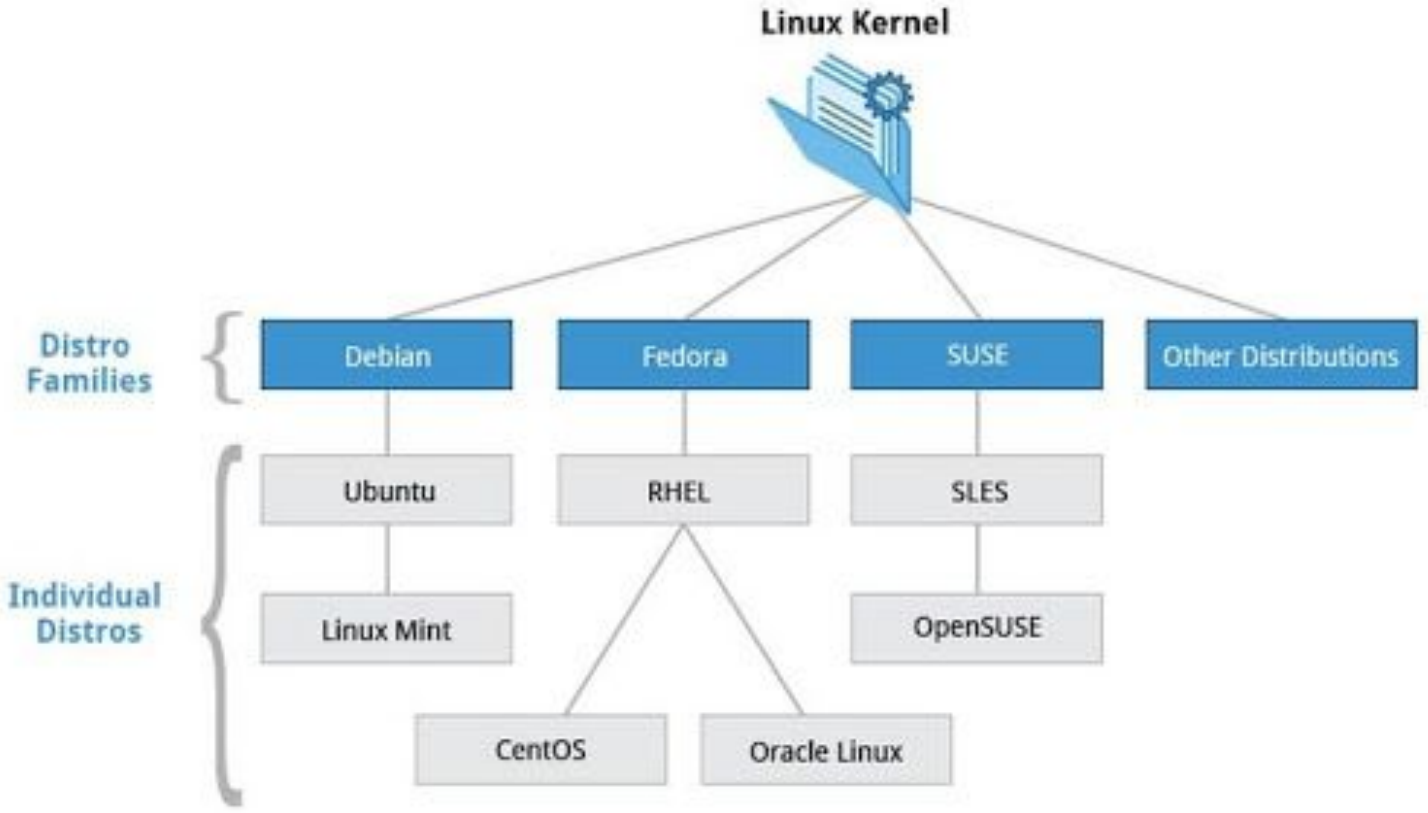
Asst. Prof. Ashwini Mathur

Linux distros compared

	distro	started	pkgs	format	architectures
	Fedora (Redhat)	2003 (1995)	8,000	rpm	x86 x86-64 ppc ppc64
	openSUSE (Novell)	1994	22,000	rpm	x86 x86-64 ppc
	Mandriva	1998	~20,000	rpm	x86 x86-64
	Debian	1993	~25,000	deb	x86 x86-64 ia-64 ppc ppc64 sparc64 arm hppa mips s390 s390x alpha
	Ubuntu (Canonical)	2004	~25,000	deb	x86 x86-64 (ppc)

Unix family Tree




Linux Kernel
Distro Families
Debian
Fedora
SUSE
Other Distributions
Individual Distros
Ubuntu
RHEL
SLES
Linux Mint
OpenSUSE
CentOS
Oracle Linux

Package managers

In Linux, a **package manager** is a collection of software tools that automate the process of installing, upgrading, configuring, and removing software. A package manager maintains a database of information about installed packages (called the **package database**) that enables the package manager to uninstall software, establish whether a new package's dependencies have been met, and determine whether a package you're trying to install has already been installed.

Here is a list of the major functions of a packet manager:

- verifying file checksums to ensure the validity of the installed software.
- verifying digital signatures to authenticate the origin of packages.
- upgrading software with latest versions.
- managing dependencies to ensure a package is installed with all packages it requires.
- creating binary packages.

Two major versions of package management tools exist:

- RPM Package Manager (RPM)
- Debian package manager

RPM - RedHat Package Management

All software on a Red Hat Enterprise Linux system is divided into RPM packages which can be installed, upgraded, or removed. This part describes how to manage the RPM packages on a Red Hat Enterprise Linux system using graphical and command line tools.

RPM Package Manager

Some of the RPM-based distributions are Red Hat, Fedora, Fermi Linux, CentOS, SUSE Enterprise, openSUSE, and Mandriva. RPM is a package management system used to build, install, verify, update, and uninstall software in these distributions. This can be done using the *rpm* command, along with options that specify the action you would like to perform.

Some common options used with the *rpm* command are:

The *RPM Package Manager* (RPM) is an open packaging system, which runs on Red Hat Enterprise Linux as well as other Linux and UNIX systems

The utility works only with packages built for processing by the `rpm` package. For the end user, RPM makes system updates easy. Installing, uninstalling, and upgrading RPM packages can be accomplished with short commands. RPM maintains a database of installed packages and their files, so you can invoke powerful queries and verifications on your system.

Installing packages using *rpm*

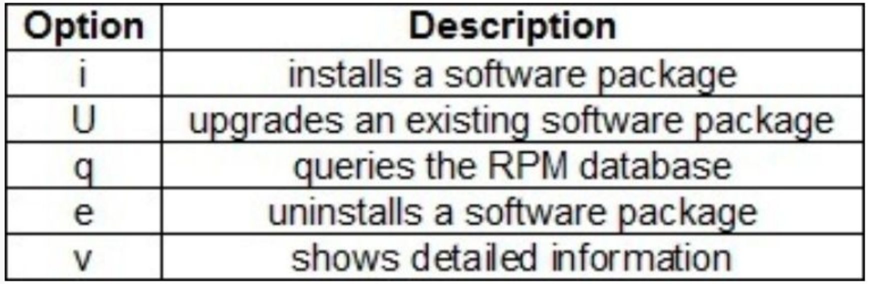
To install a package using the *rpm* command, type *rpm* followed by the *-i* option and the name of the package you would like to install. For example, to install **kdessh**, we would use the following command:

Getting information about packages using *rmp*

To determine the version and release information for software packages, you can use the *rpm* command with the *-q* option:

uninstall packages using *rpm*

To uninstall software packages using *rpm*, use the *-e* option:



Option	Description
i	installs a software package
U	upgrades an existing software package
q	queries the RPM database
e	uninstalls a software package
v	shows detailed information

The convention for naming RPM packages is: ***packagename-a.b.c-x.arch.rpm***. Each part of the package name has a meaning:

Package Name – the name of the package (***packagename***)
Version Number – package’s version number (***a.b.c***)
Build Number – software release number (***x***)
Architecture – the architecture for which the package was built (***arch***)

For example, the package name **kdessh-4.3.5-0.3.3.x86_64.rpm** represents a program named **kdessh**, the version **4.3.5**, the build **0.3.3** for **x86_64** systems.

yum package manager

yum (**Yellowdog Updater, Modified**) is a command-line package manager for RPM-based Linux distributions such as CentOS, Red Hat, and Fedora. *yum* enables you to install a package and all its dependencies, delete a package, upgrade existing packages, search for packages, etc. The syntax of the *yum* command is:

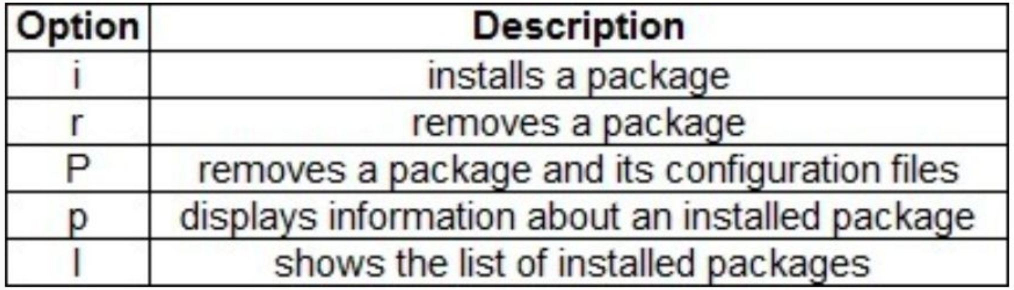
yum [OPTIONS] [COMMAND] [PACKAGE]

To install a new package using *yum*, use the *yum install* command, along with the name of the package. For example, to install the **net-tools** package, we would use the *yum install net-tools* command:

Debian package manager

Debian packages are adopted by several Linux distributions, most notably Ubuntu, Knoppix and Linux Mint. Debian packages usually have the **.deb** extension. To install, remove or list Debian packages, the *dpkg* command is used.

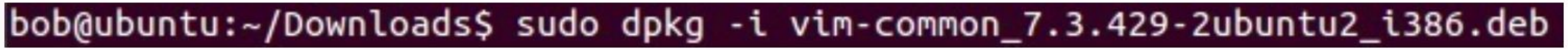
Some common options used with the *dpkg* command are:



Option	Description
i	installs a package
r	removes a package
P	removes a package and its configuration files
p	displays information about an installed package
l	shows the list of installed packages

Installing packages using *dpkg*

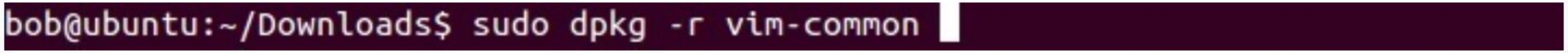
To install a **.deb** package, use the *dpkg* command with the *-i* option:



```
bob@ubuntu:~/Downloads$ sudo dpkg -i vim-common_7.3.429-2ubuntu2_i386.deb
```

Removing packages using *dpkg*

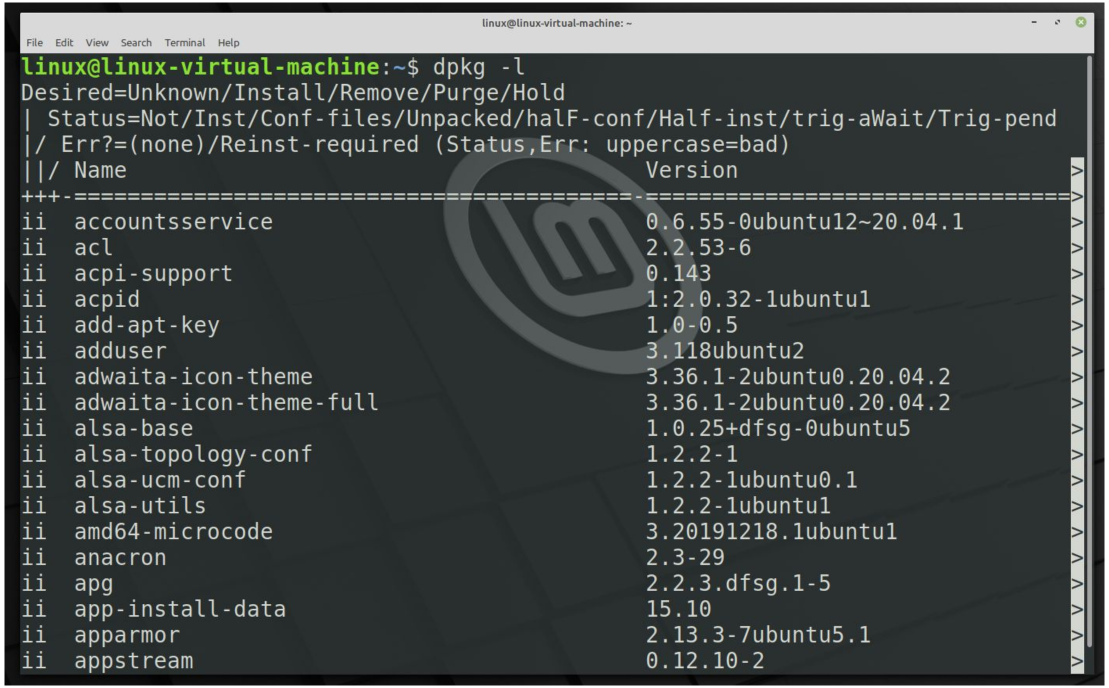
To remove a **.deb** package, use the *dpkg* command with the *-r* option



```
bob@ubuntu:~/Downloads$ sudo dpkg -r vim-common
```

Display the list of installed packages using *dpkg*

To list all installed packages on a system, use the *dpkg* command with the *-l* option

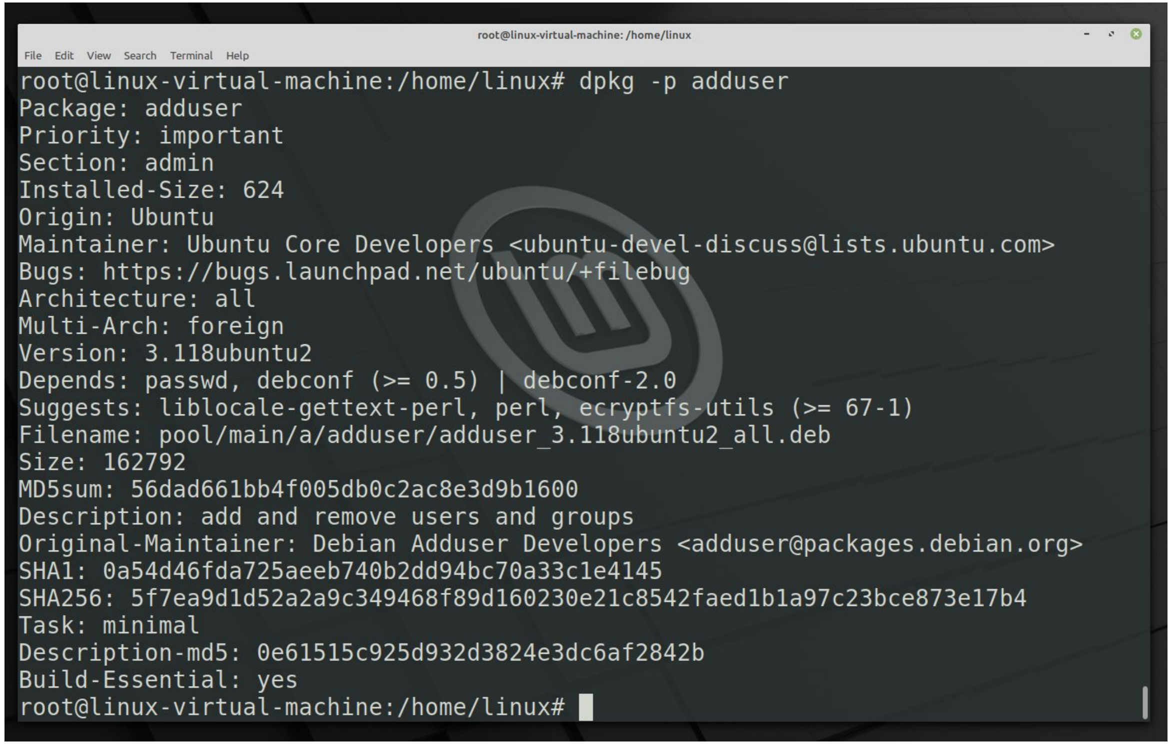


linux@linux-virtual-machine: ~
File Edit View Search Terminal Help

```
linux@linux-virtual-machine:~$ 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  accountsservice                                0.6.55-0ubuntu12~20.04.1       >
ii  acl                                            2.2.53-6                       >
ii  acpi-support                                   0.143                          >
ii  acpid                                          1:2.0.32-1ubuntu1              >
ii  add-apt-key                                    1.0-0.5                        >
ii  adduser                                        3.118ubuntu2                   >
ii  adwaita-icon-theme                             3.36.1-2ubuntu0.20.04.2        >
ii  adwaita-icon-theme-full                        3.36.1-2ubuntu0.20.04.2        >
ii  alsa-base                                      1.0.25+dfsg-0ubuntu5           >
ii  alsa-topology-conf                             1.2.2-1                        >
ii  alsa-ucm-conf                                  1.2.2-1ubuntu0.1               >
ii  alsa-utils                                     1.2.2-1ubuntu1                 >
ii  amd64-microcode                                3.20191218.1ubuntu1            >
ii  anacron                                        2.3-29                         >
ii  apg                                            2.2.3.dfsg.1-5                 >
ii  app-install-data                               15.10                          >
ii  apparmor                                       2.13.3-7ubuntu5.1              >
ii  appstream                                      0.12.10-2                      >
```

Display package information using *dpkg*

To display information about an installed package, use the *-p* option along with the name of the software:


root@linux-virtual-machine: /home/linux
File Edit View Search Terminal Help

```
root@linux-virtual-machine:/home/linux# dpkg -p adduser
Package: adduser
Priority: important
Section: admin
Installed-Size: 624
Origin: Ubuntu
Maintainer: Ubuntu Core Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Architecture: all
Multi-Arch: foreign
Version: 3.118ubuntu2
Depends: passwd, debconf (>= 0.5) | debconf-2.0
Suggests: liblocale-gettext-perl, perl, ecryptfs-utils (>= 67-1)
Filename: pool/main/a/adduser/adduser_3.118ubuntu2_all.deb
Size: 162792
MD5sum: 56dad661bb4f005db0c2ac8e3d9b1600
Description: add and remove users and groups
Original-Maintainer: Debian Adduser Developers <adduser@packages.debian.org>
SHA1: 0a54d46fda725aeeb740b2dd94bc70a33c1e4145
SHA256: 5f7ea9d1d52a2a9c349468f89d160230e21c8542faed1b1a97c23bce873e17b4
Task: minimal
Description-md5: 0e61515c925d932d3824e3dc6af2842b
Build-Essential: yes
root@linux-virtual-machine:/home/linux# 
```

APT - Advanced Packaging Tool

Advanced Packaging Tool (APT) is a package manager originally designed for Debian as a front-end for the *dpkg* utility. It is used to install or upgrade all necessary dependent applications so that **.deb** packages can be installed.

The APT suite of tools includes a couple of programs. Two of the most commonly used are:

- **apt-cache** – provides information about the Debian package database.
- **apt-get** – used to install, upgrade or remove software packages.

Several front-ends to APT exist, which provide more advanced installation functions and more intuitive interfaces. These include:

- **Synaptic Package Manager** – GTK+ graphical user interface.
- **Ubuntu Software Center** – a GTK+ graphical user interface developed by the Ubuntu project.
- **aptitude** – a console client with CLI.
- **KPackage** – part of KDE.

apt-get command

apt-get is a command from the APT suite of tools that is used to install, upgrade or remove software packages in Debian and Debian-based Linux distributions. This command has some neat features, such as ease of use over simple terminal connections (SSH) and the ability to be used in system administration scripts.

The *apt-get* command obtains information about available packages from the sources listed in **/etc/apt/sources.list** and then uses that information to upgrade or install packages. Here is an example **sources.list** file:

Update the list of available packages

To obtain updated information about packages available from the installation sources listed in **/etc/apt/sources.list**, use the *apt-get update* command:

Upgrade all installed packages to the newest versions

To upgrade all installed packages, use the *apt-get upgrade* command. You can use the *-u* option to display the complete list of packages which will be upgraded:

Install a package by package name

To install a package by its name, use the *apt-get install PACKAGE_NAME* command:

apt-get utilities

Remove a package by package name

To remove a package by its name, use the *apt-get remove PACKAGE_NAME* command:

Check the package database for consistency

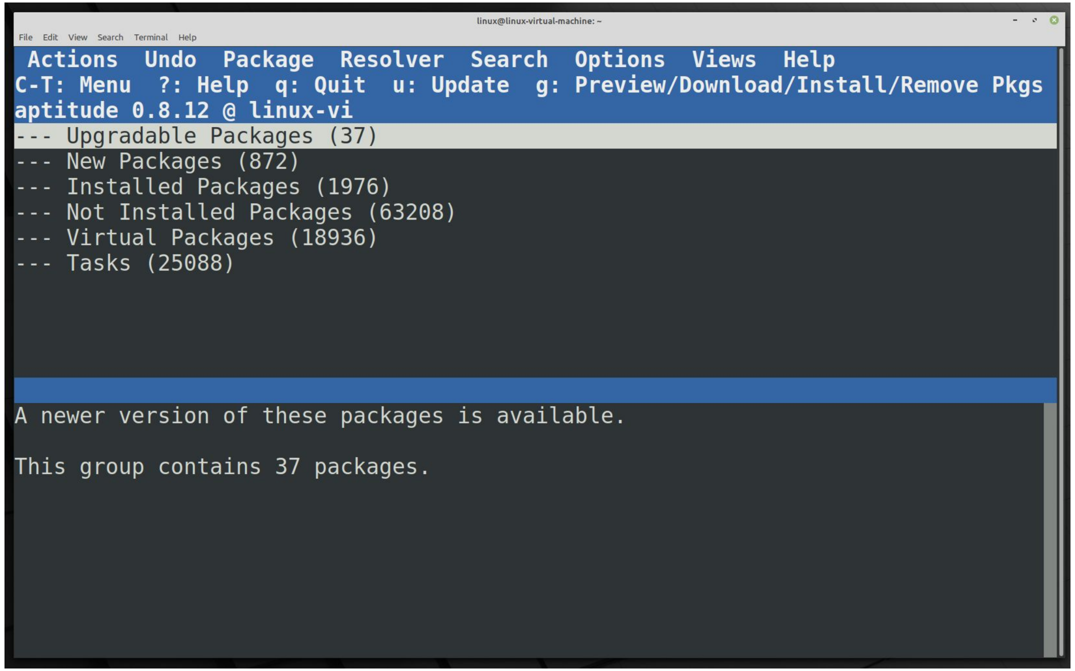
To check the package database for consistency and broken package installations, use the *apt-get check* command.

Remove unused package files

To clear out information about retrieved files from the Debian package database, use the *apt-get clean* command. This command removes everything but the lock file from **/var/cache/apt/archives** and **/var/cache/apt/archives/partial**:

aptitude

aptitude is a front-end to the **Advanced Packaging Tool (APT)** in Debian and Debian-based distributions. This program allows users to view the list of packages and perform package management tasks such as installing, removing or upgrading packages

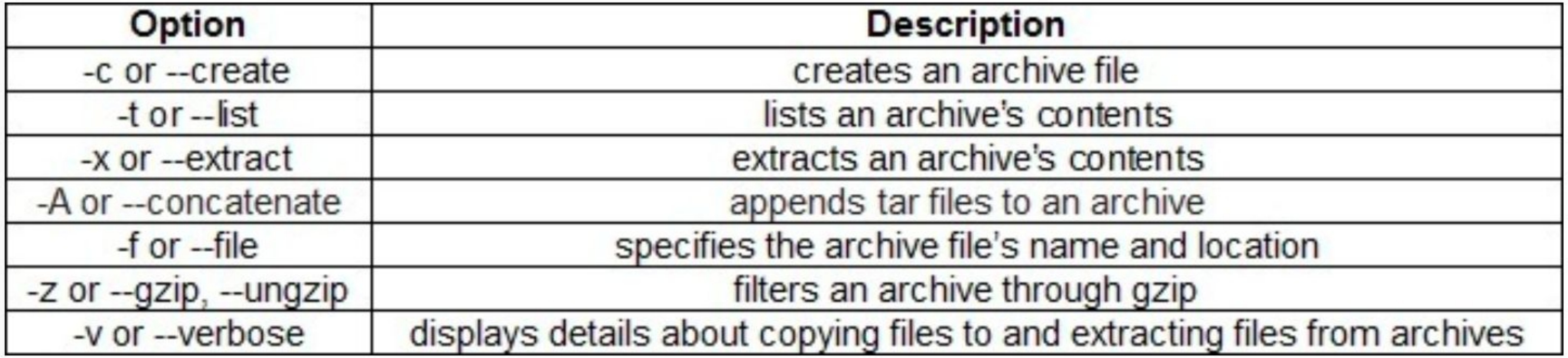

linux@linux-virtual-machine: ~
File Edit View Search Terminal Help

```
 Actions  Undo  Package  Resolver  Search  Options  Views  Help
C-T: Menu  ?: Help  q: Quit  u: Update  g: Preview/Download/Install/Remove Pkgs
aptitude 0.8.12 @ linux-vi
--- Upgradable Packages (37)
--- New Packages (872)
--- Installed Packages (1976)
--- Not Installed Packages (63208)
--- Virtual Packages (18936)
--- Tasks (25088)

A newer version of these packages is available.

This group contains 37 packages.
```

tar (tape archive) program

The **tar** (**tape archive**) program creates archives by combining files and directories into a single file. **Tarballs** (archive files created by tar and usually compressed with **gzip** or **bzip2**) are often used to distribute software packages in the Linux world.

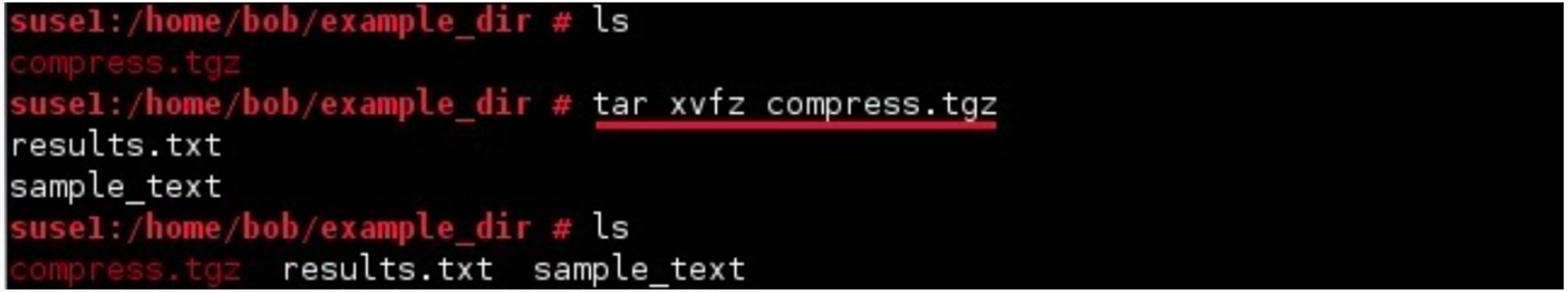
You can use many options with the *tar* program to modify how it functions when creating and extracting archives. Here is a list of the most common ones:



Option	Description
-c or --create	creates an archive file
-t or --list	lists an archive's contents
-x or --extract	extracts an archive's contents
-A or --concatenate	appends tar files to an archive
-f or --file	specifies the archive file's name and location
-z or --gzip, --ungzip	filters an archive through gzip
-v or --verbose	displays details about copying files to and extracting files from archives

Compression tools such as **gzip**, **bzip2**, and **xz** are often used with tar to apply compression to the tarball as a whole rather than to the individual files. Typically, files compressed with these utilities have **.gz**, **.bz2**, or **.xz** extensions, respectively.

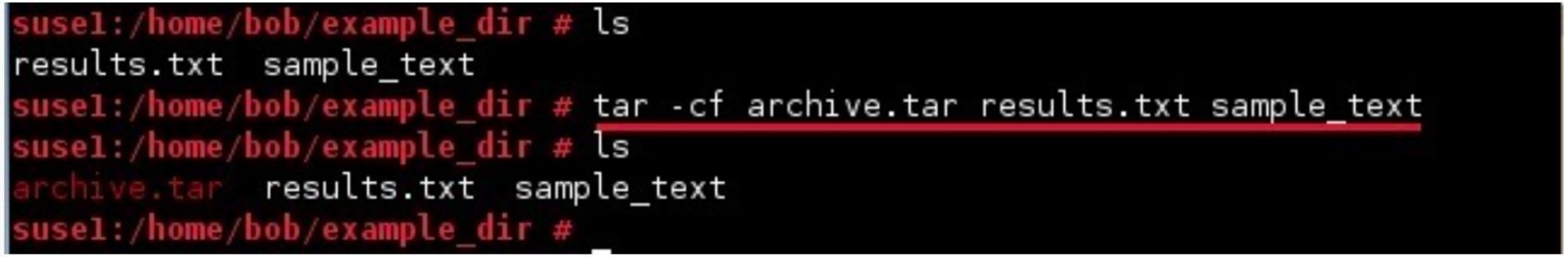
To extract the archive created above, we can use the following command:



```
suse1:/home/bob/example_dir # ls
compress.tgz
suse1:/home/bob/example_dir # tar xvfz compress.tgz
results.txt
sample_text
suse1:/home/bob/example_dir # ls
compress.tgz  results.txt  sample_text
```

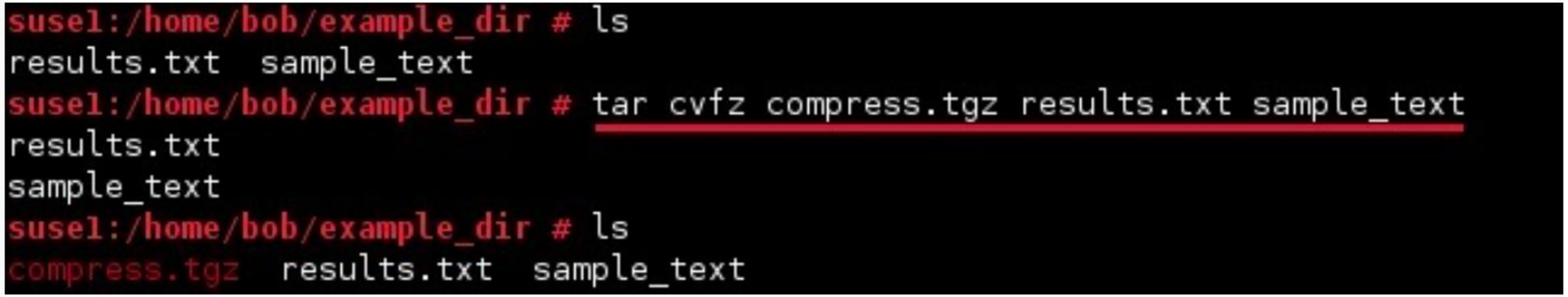
Here are some examples of the *tar* command.

To create a tar archive called **archive.tar** that contains the files **results.txt** and **sample_text**, we can use the following command:



```
suse1:/home/bob/example_dir # ls
results.txt  sample_text
suse1:/home/bob/example_dir # tar -cf archive.tar results.txt sample_text
suse1:/home/bob/example_dir # ls
archive.tar  results.txt  sample_text
suse1:/home/bob/example_dir # _
```

To archive and compress files **results.txt** and **sample.txt** into a tarball file named **compress.tgz**, we can use the following command:



```
suse1:/home/bob/example_dir # ls
results.txt  sample_text
suse1:/home/bob/example_dir # tar cvfz compress.tgz results.txt sample_text
results.txt
sample_text
suse1:/home/bob/example_dir # ls
compress.tgz  results.txt  sample_text
```

Nmap is a robust network security tool written by Gordon Lyon. It was released more than 20 years ago and has since become the de facto standard for network mapping and port scanning. Nmap helps network administrators around the world to discover hosts and services on a computer network and build a map of their network.

Nmap was originally developed for Linux, but it has been ported to most major operating systems, such as Microsoft Windows, HP-UX, Solaris,, etc. Released under the GPL license, Nmap is free to use, study, and modify.

Although usually used for port scanning and network mapping, Nmap can also be used for other purposes, such as:

- host discovery.
- operating system and service version detection.
- finding out network information about targets, such as DNS names, device types, and MAC addresses.
- ability to scan for well-known vulnerabilities.