6.1 Common RPM Commands

Command	Function
rpm	Uses the Red Hat Package Manager (RPM) to manage packages. Package options are: **--rebuilddb** rebuilds the database indices from the installed package headers. **--initdb** creates a new database. **--checksig** checks the authenticity of the package. The option checks the package's digital signing key against the package to ensure it has not been altered. **-i** installs a package. Uses the entire package filename when installing. **-h** prints hash marks as the package archive is unpacked. **-v** displays a verbose version of the installation. **--test** tests a package for uninstalled dependencies without actually installing it. **--nodeps** installs the package without checking for dependencies. This is not recommended. **--force** installs the package regardless of whether a newer version of the package is already installed, package files overwrite files from previously installed packages, or if the package replaces other installed packages. **-e** uninstalls (e.g., erases) a package. To uninstall a package, use the package name, not the file name. If dependencies exist, the dependent packages must first be removed. **-U** updates an installed package to the newest version. **-F** upgrades the package, but only if an earlier version currently exists on the system. **-q** queries the computer for information about installed packages. Use this with -a to list all packages and -l to show the files associated with the package. **-V** verifies that packages are free from errors by performing an MD5 checksum on the package. RPM only gives output when packages have errors. If errors are present, the command displays the error code and the file name. The error codes are: **S** indicates a problem in the size of a file. **M** indicates a problem with a file's mode. **5** indicates a problem with the MD5 checksum of a file. **D** indicates a problem with a file's revision numbers. **L** indicates a problem with a file's symbolic link. **U** indicates a problem with a file's ownership. **G** indicates a problem with a file's group. **T** indicates a problem with the modification time of a file. **c** indicates the specified file is a configuration file. **'.'** in place of a code letter indicates that no error is present in that area.
rpm2cpio	Converts RPM packages into a cpio archive. This is useful for extracting files from an RPM package without installing and searching for specific files.

6.2 YUM and DNF

Command	Function
yum	Installs RPM packages, including their dependencies. Be aware of the following actions and options: **list** displays lists of packages. **install** installs a package. Use the entire package filename for installations. **list updates** displays whether updates are available for packages. **update** updates RPM packages. **list available** lists packages that are available to install. **search** searches all packages for a specified term. **info** displays detailed package information. **provides** display which packages are associated with a specific file. **whatprovides** " " **remove** uninstalls a package. **erase** " " **-y** bypasses confirmation prompts.
yumdownloader	Downloads a package without installing it.
createrepo	Creates a repository list of RPM packages stored locally or on a network. Be aware of the following options: **-g** specifies an XML file for the repository. **-x** excludes specific file globs.
dnf	Installs RPM packages, including their dependencies. Be aware of the following actions and options: **list** displays lists of packages. **install** installs a package. Use the entire package filename for installations. **list updates** displays whether updates are available for packages. **update** updates RPM packages. **list available** lists packages that are available to install. **search** searches all packages for a specified term. **info** displays detailed package information. **provides** display which packages are associated with a specific file. **whatprovides** " " **remove** uninstalls a package. **erase** " " **-y** bypasses confirmation prompts.

6.3 Debian Package (dpkg)

Command	Function
dpkg	Installs Debian packages on Debian distributions. Be aware of the following dpkg options: **-i** installs a package. **--configure** reconfigures an unpacked package. **-r** removes the package, but does not delete the configuration files. **-P** completely uninstalls the package, including the configuration files. **-p** lists information about a currently installed Debian package. **-I** *(uppercase i)* lists information about packages that are not installed. **--info** " " **-l** *(lowercase l)* displays all packages with names that match a specified pattern. **-L** shows the installed files for a package. **-S** finds a package associated with specified files. **-C** searches for packages that have been installed only partially on the system. **-B** disables packages that have dependencies on the package being removed. **--ignore-depends** ignores dependency checking for specified packages. **-no-act** prevents changes from being written. **-G** prevents a package from being installed if a newer version of the package already exists on the computer. **-E** does not install the package if the same version of the package is already installed. **-R** installs the package recursively. The **dpkg-reconfigure** command reconfigures an already installed package.
apt-cache	Retrieves information about the Debian package database. Be aware of the following apt-cache options: • **showpkg** displays information about a package in the database. • **stats** shows the number of packages installed, dependency information, and other package cache statistics. • **unmet** lists any missing dependencies in the package cache. • **depends** shows all of the package's dependencies. • **pkgnames** displays whether a package is installed on the system. When the package name is left off, the command shows information for all packages on the computer. • **search** searches for a package in the cache.

apt-get

Downloads and install packages. Be aware that apt-get:

- Is similar to the yum utility on an RPM distribution.
- Uses the file /etc/apt.conf or the files in the directory /etc/apt/apt.conf.d to configure apt behavior.
- Gets its information about the application repositories from the /etc/apt/sources.list file, which is built from files in the directory /etc/apt/sources.list.d.
- Automatically calculates and resolves package dependencies when installing, updating, and removing packages.

Be aware of the following apt-get options:

- **update** updates the list of packages available from the sources in /etc/apt/sources.list with the latest information about available packages.
- **upgrade** upgrades all installed packages to the latest versions in accordance with the information found in the sources listed in /etc/apt/sources.list.
- **dist-upgrade** similar to the upgrade option, but will also install new packages as needed and remove packages as needed.
- **install** Installs a package using the package name. The package name is not the filename. During the installation, apt-get retrieves the most recent version of the package.
- **remove** removes a specified package, but leaves the configuration files.
- **purge** removes the package and the configuration files.
- **source** retrieves the latest version of the package. The command accesses the /etc/apt/sources.list file to determine whether the latest package version is installed.
- **check** checks the package database for consistency and errors.
- **clean** removes unneeded package information files and logs. This command is needed when not using the dselect utility to install Debian packages.
- **autoclean** removes information files about packages that can no longer be downloaded.
- **-d** downloads packages without installing them.
- **-f** attempts to fix a computer with unsatisfied dependencies. (Use with: apt-get install and apt-get remove)
- **-m** ignores package files that cannot be accessed or located.
- **-q** shows less progress information.
- **-s** simulates package installation without doing an actual install.
- **-y** automatically provides a yes response to yes/no questions in the package installation script.

apt

The apt command is similar in design and function to the apt-get tool suite mentioned above. The apt command manages dpkg packages on Debian- and Ubuntu-based distributions. You can also use it to locate, download, and install packages found in online repositories.

The syntax for using apt is as follows:

- **apt install *package_name*** installs the specified package.
- **apt remove *package_name*** uninstalls the specified package.
- **apt search *search_term*** looks for packages with the search term found in the configured repositories.
- **apt update** updates repositories with the latest list of available packages found in the configured repositories.
- **apt dist-upgrade** upgrades all installed packages with any available updated. packages.

aptitude

Views the list of packages and perform package management tasks such as installing, upgrading, and removing packages in the Advanced Packaging Tool (APT). The aptitude command is APT's front end. This command displays a list of software packages and allows the user to interactively pick packages to install or remove.

6.4 Shared Libraries

Type	Description
Dynamic	Dynamic libraries aren't directly integrated into the code of the application that uses them. Dynamic libraries: • Are linked to the application that shares its code. • Have a .so or .so.version extension (.so stands for shared object). • Are typically stored in /usr/lib/ and /usr/local/lib/. • Can degrade program load time if the library is already in use by another program. • Are similar to Dynamic Link Libraries (DLLs) in Windows. Be aware of the following management programs and files for dynamic libraries: • **/lib/ld.so** is a program that finds and loads the needed shared libraries. It also prepares the program to run and executes it. • **/etc/ld.so.conf** is a file that contains a list of directories to search for shared libraries. Some lines in the file begin with the include directive, which lists files that are to be included as if they were part of the main file. • **/etc/ld.so.cache** is a cached list of libraries found in the directories specified in /etc/ld.so.conf. The system uses this cached list instead of loading /etc/ld.so.conf every time a program runs. Use the following methods to configure dynamic libraries on a Linux system: • Modify /etc/ld.so.conf to add the library paths. • Use the LD_LIBRARY_PATH environment variable to specify additional directories to search for library files.
Static	Static libraries are integrated into the code of an application itself when that code is compiled. Static libraries: • Have an .a filename extension. • Are used when dynamic libraries are not available. • Increase the size of the application. • Eliminate dependency issues associated with dynamic libraries.

6.4 Library Management Commands

Command	Function
ldd	Discovers which libraries are used by another library (e.g., library dependencies). • Check the complete dependency chain when using ldd to track down problems. • Run ldd as root (recommended). Be aware of the following options: **-v** displays all information. **-u** displays unused direct dependencies. **--version** displays the version number of ldd.
ldconfig	Reloads the library cache every time libraries are added or removed and updates the symbolic links. This creates the necessary links and cache for the most recently shared libraries found in the directories specified on the command line. These are found in the /etc/ld.so.conf file and in the trusted directories (/lib and /usr/lib). Be aware of the following options: **-v** summarizes the directories and files it registers as it reloads the cache. **-N** updates symbolic links, but does not update the cache. **-n** updates the links contained in the directories specified on the command line. **-X** updates the cache but does not update symbolic links. **-f** changes the configuration file from the /etc/ld.so.conf default. **-C** changes the cache location from the /etc/ld.so.cache default. **-r** treats a new directory as if it were the root directory. This is helpful when you're recovering a badly corrupted system or installing a new OS. **-p** displays the current library cache, including all the library directories and their respective libraries.