* The Linux Foundation
  + Linux
    - Created in 1991 by Linus Torvalds
    - Free Open Source computer operating system.
    - Written to be a free and open source version of UNIX.
    - Files are stored in a hierarchical system with the top node being **root** or **/**.
    - Everything is represented as file-like objects.
  + The Linux foundation is a nonprofit organization that sponsors the work of Linus Torvalds.
    - Its objectives are
      * Promote Linux and provide neutral collaboration and education.
      * Improve Linux as a technical platform.
      * Sponsor the work of Linux Torvalds.
* Linux Philosophy and Concepts
  + Major distribution families (distro families)
    - Linux Kernel
      * Debian
        + Ubuntu (individual Distros)

Linux Mint

* + - * + Used

For Servers

For Desktop

For Cloud Deployments

* + - * Fedora
        + RHEL (Red Hat Enterprise Linux)

CentOS

Oracle Linux

* + - * + Used

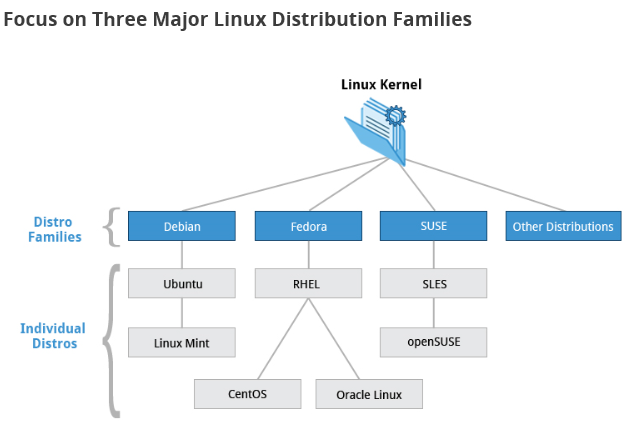
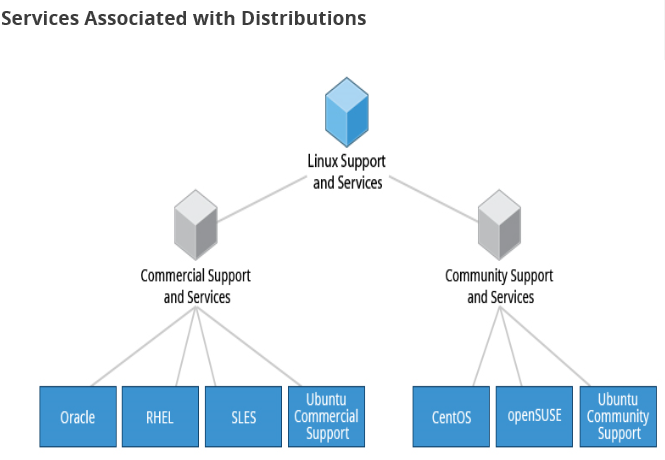
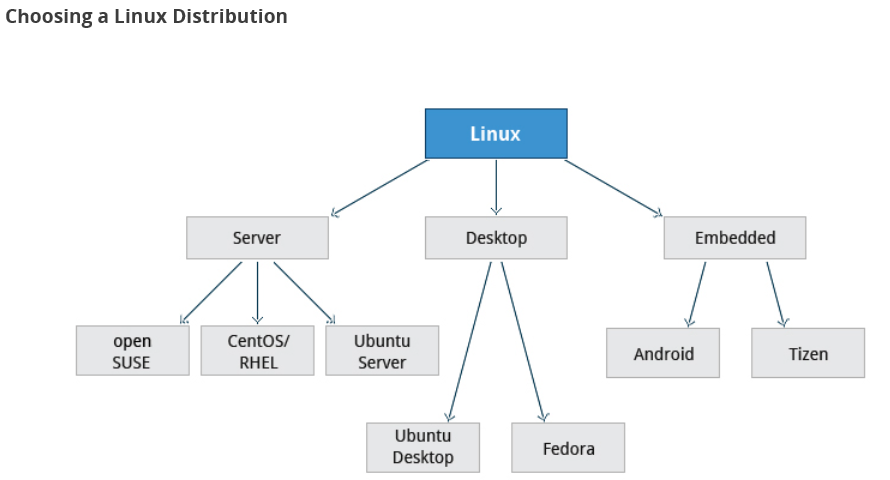
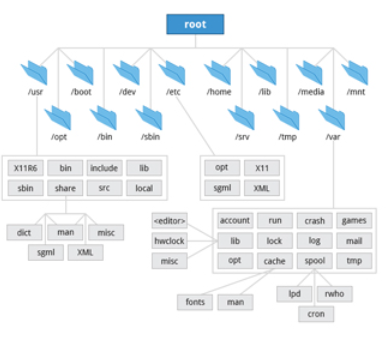
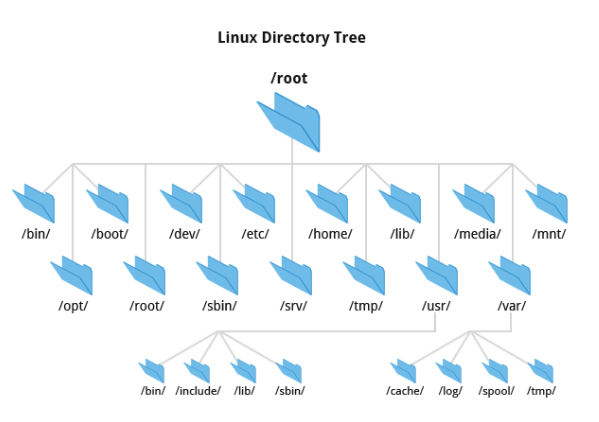
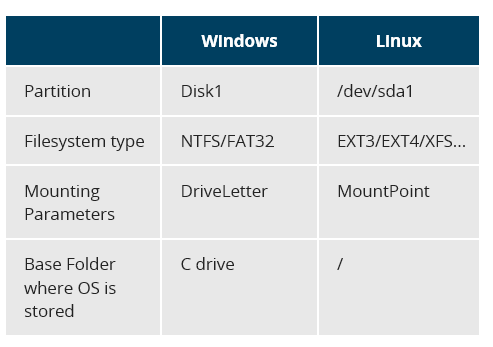
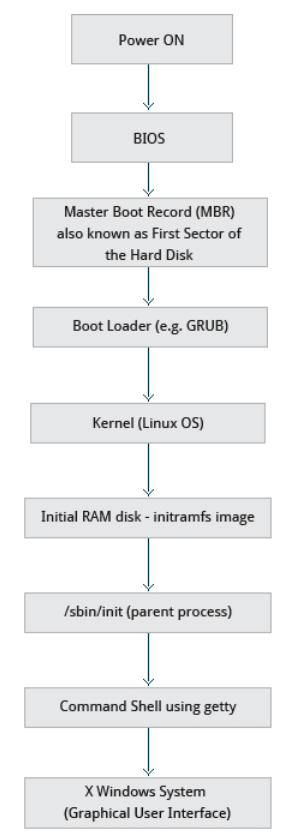
by Oracle

* + - * SUSE
        + SLES (SUSE Linux Enterprise Server)

OpenSUSE

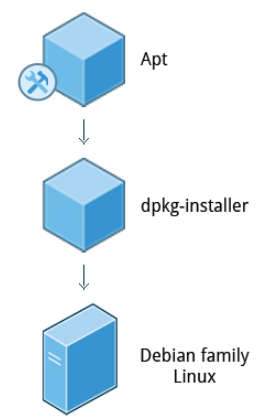
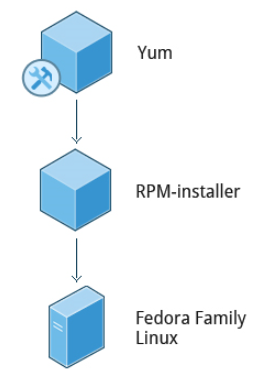
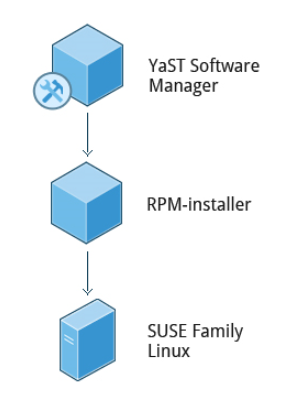
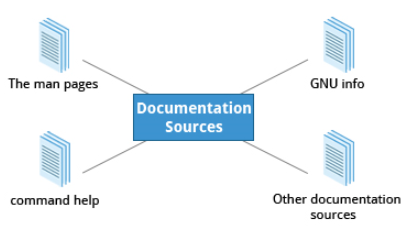
* + - * + Used

Retail Sector

* + - * Other Distributions
      * 
      * 
      * 
  + Kernel
    - Is the glue between hardware and applications.
  + Linux Distribution
    - Consists of the kernel plus a number of other software tools for file-related operations, user management, and software package management.
      * Other tools
        + C/C++ Compiler
        + gdb debugger
        + core system libraries
        + low-level interface for drawing graphics
        + desktop environment
        + system for installing and updating components
  + Desktop environment
    - Graphical user interface on top of the operating system.
  + Command Line
    - An interface for typing commands for the operating system to execute.
* Linux Structure and Installation
  + Filesystem
    - Embodiment of a method of storing and organizing arbitrary collections of data in human-usable form.
    - Linux store their files according to a standar layout called Filesystem Hierarchy Standard (FHS). More details at: ***LFS101\_Ch3\_Sec1\_FSH.pdf***
      * 
      * 
    - Filesystems are case sensitive.
  + Partition
    - Logical Part of the disk.
    - 
  + Boot Load Process
    - 
* Graphical Interface
  + Desktop Environments
    - GNOME Desktop Environment
      * Bundled as the default desktop environment for many distrubutions like Red Hat, Fedora, CentOS, SUSE and Debian.
      * Has menu-based navigation.
    - KDE Desktop Environment
      * Used in OpenSUSE
    - Unity (based on GNOME)
    - Xfce
    - LXDE
  + Default directories
    - Every user with an account on the system will have a **home** directory usually created under **/home**.
    - **Naitilus** (Name of File Manager).
  + gedit
    - Default text editor.
  + Recycle bin location: **.local/share/Trash/files**
  + Linux always uses the UTC time.
* System Configuration
  + Network Manager
    - NTP (Network Time Protocol) is used to set the local time via Internet servers.
    - Wired connections usually do not require complicated or manual configuration. The Network Manager automatically detects the signal and sets the settings via DHCP. For static configurations that don’t use DHCP a manual setup can be performed.
    - Network Manager supports following VPN technologies:
      * OpenVPN
      * Cisco OpenConnect
      * IPSec
  + Installing and updating software
    - Each package in a Linux distribution provides one piece of the system, such as the Linux kernel, the C compiler, the shared software code for interacting with USB devices, or the Firefox web browser.
    - **dpkg** (Debian) and **RPM** (Others) are the most popular package management systems on Linux.
    - Debian Family System
      * Manages packages in Debian systems via **dpkg** package manager.
        + dpkg

Can install, remove and build packages.

Does not automatically download and install packages.

* + - * + apt (Advanced Package Tool): Higher level package manager system.
        + 
    - Fedora Family System
      * 
    - SUSE Family Linux
      * 
* Finding Linux Documentation
  + Linux Documentation Sources
    - The man pages (manual pages)
    - GNU info.
    - The help command and –help option.
    - Other Documentation Sources. E.g. <https://www.gentoo.org/doc/en/>
      * 
  + The man pages
    - Provide in-depth documentation about many programs and utilities as well as other topics.
    - Typing **man** with a topic name as an argument retrieves the information stored in the topic’s **man pages.**
    - Commands
      * man <command>: Shows documentation for the specified command.
      * man –f <command> / whatis <command>: Shows short documentation.
      * man –k <command>: Shows all man pages that discuss the specified command.
      * man <chapter\_number 1..9> <command>: Displays all man pages for a particular chapter.
      * man –a <command>: Displays all pages with the given name in all chapters.
  + GNU Info System
    - GNU project’s standard documentation format.
    - The info system is more free-form and supports linked sub-sections.
      * Command Line Info Browser
        + Commands

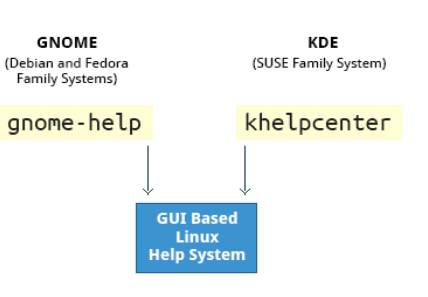
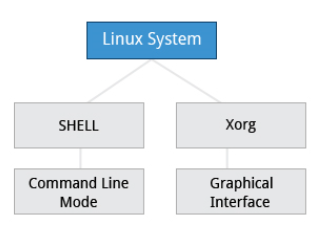
info: View all available info.

info <topic\_name>: View info for specific topic.

n: Go to next node

p: Go to previous node

u: Move one node up in the index.

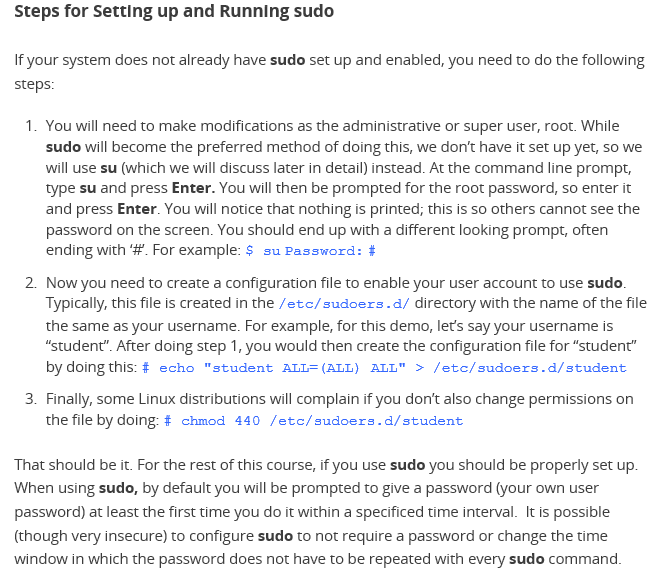
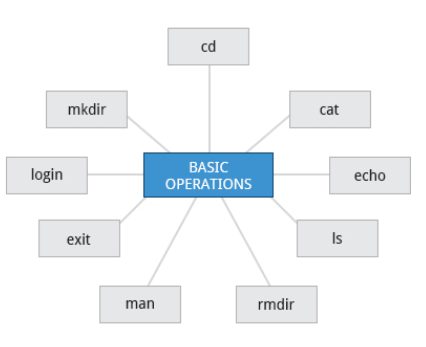
* + help Option
    - Commands
      * <command> --help
* Other Documentation Sources
  + <http://linuxcommand.org/tlcl.php>
  + Ubuntu: <https://help.ubuntu.com/>
  + CentOS: <https://www.centos.org/docs/>
  + OpenSUSE: <http://en.opensuse.org/Portal:Documentation>
  + GENTOO: <http://www.gentoo.org/doc/en>
  + Other commands
    - gnome-help: Shows the graphical help section.
    - 
* Command Line Operations
  + Terminal emulator: Program that emulates a stand alone terminal within a window on the desktop.
    - gnome-terminal
    - xterm
    - rxvt
    - konsole
    - terminator
  + Certain Linux distributions distinguish version of the install media between desktop (with X) and server (usually without X window system).
  + Linux production servers are usually installed without X.
  + Removing X from a production server can be very helpful in maintaining a lean system which can be easier to support and keep secure.
  + 
  + Virtual Terminals
    - They are console sessions that use the entire display and keyboard outside the graphical environment.
    - Multiple can be opened at once but only one is visible at a given time.
    - One virtual terminal is usually reserved for the graphical environment. VT7 for Ubuntu and VT1 for CentOS/RHEL and OpenSUSE.
    - To switch between the VTs press **Ctrl. + Alt + F1…Fx.** or press **Alt + F1…Fx**. when in a VT not running the X environment.
  + The Command Line
    - Most input lines entered at the shell prompt have three basic elements:
      * Command: Name of the program being executed.
      * Options (Switches): Modify what the command may do. Usually start with one or two dashes. E.g. –p or –print,
      * Arguments: Represent what the command operates on.
  + Turning Off the Graphical Desktop
    - Debian
      * Desktop Manager runs as a service which can be stopped.
        + Commands

sudo service gdm stop

sudo service lightdm stop

* + - RPM-based systems
      * The Desktop Manager is run directly by **init** when set to run level 5. Switching to a different runlevel stops the desktop.
        + Commands

sudo telinit 3

* + sudo
    - Allows users to run programs using the security privilages of another user, generally root (superuser).
    - Is similar to that of **run as** in **Windows**.
    - On your own systems you may need to set up and enable **sudo** to work correctly.
    - For **Ubuntu**, **sudo** is already always set up for you during installation.
    - For **Fedora** or **openSUSE**  families you will likely need to set up **sudo** to work properly for you after initial installation.
    - Setup if no **sudo** access:
      * 
  + Basic Operations
    - 
    - Rebooting and Shutting Down
      * Commands
        + halt | poweroff | shutdown –h | shutdown –h now: Shutdown system.
        + reboot | shutdown –r: Restart system.
        + sudo shutdown –h <time e.g. 10:00> “<message>”: Schedule shutdown and send message to users before shutting down.
        + shutdown –c: Cancel scheduled shutdown.
    - Locating Applications
      * Software packages can be installed in various directories which can be:
        + /bin
        + /usr/bin
        + /sbin
        + /usr/sbin
        + /opt
      * Commands to check for where a utility is installed
        + which <application\_name>: E.g. which diff
        + whereis <application\_name: E.g. whereis diff
    - Accessing directories
      * When opening a terminal, the default directory should be your home directory.
      * Commands
        + echo $HOME: Print path of home directory.
        + pwd: Display the present working directory.
        + cd: Change directory
        + cd ~ | cd: Change to home directory.
        + cd ..: Change to parent directory.
        + cd -:Change to previous directory.
    - Understanding Absolute and Relative Paths
      * Absolute: Begins with the root directory and follows the tree. Always starts with **/**.
      * Relative: Starts from the present working directory. Never start with **/**.
    - Exploring the Filesystem
      * Commands:
        + tree –d: View directories in tree form. Only lists directories.
        + cd /: Change to root directory.
        + ls: Lists contents of present working directory.
        + ls –a: List all files including hidden files and directories.
        + tree: Displays a tree view of the filesystem.
    - Hard and Soft (Symbolic) Links
      * **ln** can be used to create **hard links** and with the **–s** option, **soft links**.
      * Hard links
        + ln <filename1> <filename2>:Create hard link.
        + Use **ls -li <filename1> <filename2>** to check for hard links.
        + A hard link implies that more than one name is associated to a file.
      * Symbolic (Soft links)
        + ln –s <filename1> <filename2>: Create soft link.
        + Use **ls –li <filename1> <filename2>** to check for soft links.
        + Symbolic links don’t take extra space on the filesystem (unless names are very long).
        + Help with creating shortcuts. E.g from home directory to long pathnames.
        + Unlike hard links, soft links can point to objects even on different filesystems (partitions).
        + Modifying or deleting a linked file may have different effects.
    - Navigating the Directory History
      * For remembering more than just the last directory visited, use the **pushd** command instead of **cd** which pushes your starting directory onto a list.
      * Use **popd** to go back to those directories in the stack in reverse order.
      * To view the list of directories in the stack use **dirs** command.
  + Searching For Files
    - Standard File Streams
      * When commands are executed, by default there are three standard **file stream (descriptors)** always open for use
        + stdin: Standard input.
        + stdout: Standard output.
        + Stderr: Standard error.
      * Open files are represented internally by what are called **file descriptors** represented by numbers starting at 0. **stdin** is 0, **stdout** is 1, **stderr** is 2. Any new file that is open typically start with 3.
    - I/O Redirection
      * Commands
        + If we have a program called **do\_something**

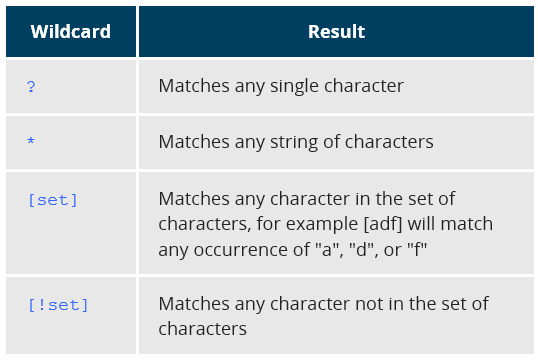
do\_something < <input\_file>: do\_something uses the file as input.

do\_something > <output\_file>: do\_something sends output to file.

do\_something 2> <error\_file>: do\_something sends error output to file.

* + - Pipes
      * UNIX/Linux philosophy is to have many simple and short programs (commands) cooperate together to produce quite complex results.
      * Pipes are used to pipe the output of one command into another as its input.
        + command1 | command2 | command 3

E.g. **ls | head -3**: Shows the first three elements of the result of executing ls command.

* + Searching for Files
    - locate
      * Performs a search through a previously constructed database of files and directories on the system, matching all entries that contain a specified character string.
      * Utilizes the database created by **updatedb**. Mentioned database is usually updated once a day but can be updated anytime by running **updatedb.**
      * Commands
        + locate <string>: List of files that contain the specified string in their name.
        + locate <string> | grep <string>: List of files that contain both strings in their name.
      * Wildcard and Matching File Names
        + 
        + E.g.

ls g?: List any two letter files that start with g.

ls g[a-m]???: List any file letter files that start with g and second letter is between a-m.

* + - Finding Files In a Directory
      * **find** recuses down the filesystem tree from any particular directory and locates files that match specified conditions.
      * Commands
        + find: lists all files in the current directory and all of its subdirectories.
        + Options

-name: Only list files with a certain pattern in their name.

-iname: Ignore case.

-type: Restrict to certain type of file (d, l, f).

* + - * + Examples

find /usr –name gcc: Files or directories named “gcc”.

find /usr –type d –name gcc: Directories named “gcc”.

find /urs –type f –name test1: Files named “test1”.

* + - * Using Advanced find options
        + Commands

Options

–exec: Allows to run commands on the files that match the search criteria.

To find and remove all files that end with .swp: find –name “\*.swp” –exec rm {} ‘;’

{} will be filled with the filenames of the search.

–ok: Same as –exec but will prompt for permission.

* + - Finding Files Based on Time and Size
      * Options:
        + -ctime <days>: Displays files whose meta-data (file ownership, permissions, etc) were last changed.
        + -atime <days>: Search for accessed/last read.
        + -mtime <days>: Search for modified/last written.
        + Days can be preceeded by + (greater) or –(less).
        + Similar applies for minutes as –cmin, -amin ,-mmin.
        + -size <number>

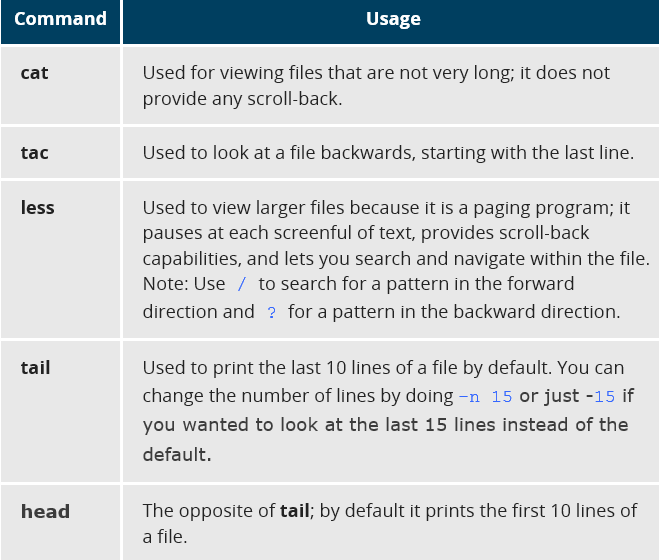
E.g. find / -size +10M: Finds files greater than 10MB.

c: B

k: KB

M: MB

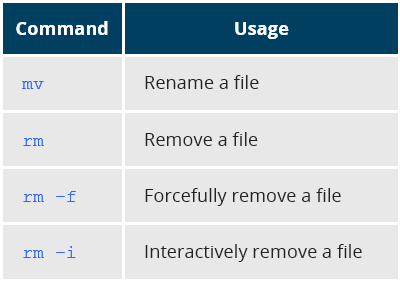
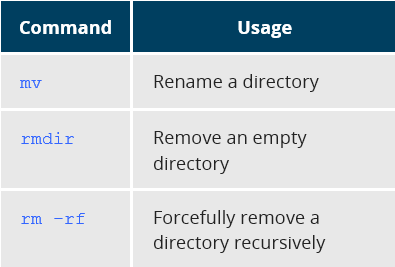
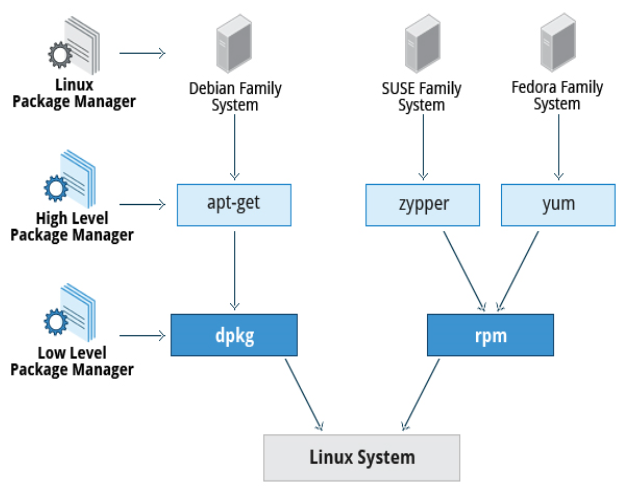
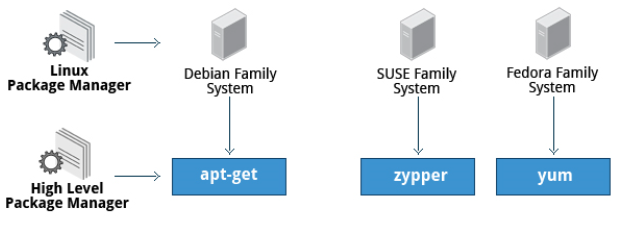
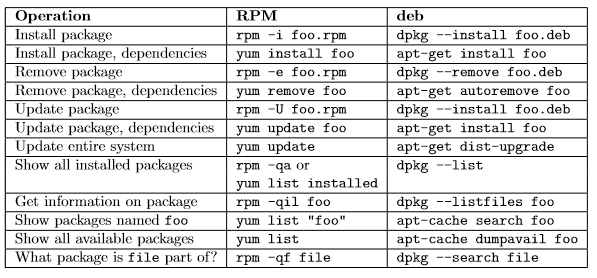
G: GB

* + Working With Files
    - Viewing files
      * 
    - touch and mkdir
      * touch
        + Often used to set our update the access, change, and modify times of files. By default it resets a file’s time stamp to match the current time.
        + Commands

touch <filename>: Also allows creation of empty files

* + - * + Options

–t : Allows to set the date and time stamp of the file. E.g. touch –t 03201600 <filename> to set time stamp to 4p.m. March 20th.

* + - * mkdir
        + Creates a new directory.
      * rmdir
        + Deletes directory only if it is empty.
        + To remove a directory and all its contents use **rm –rf**.
    - Removing a file
      * 
      * Interactively will ask for a confirmation before removing.
    - Renaming or Removing a Directory
      * 
    - Modifying the Command Line Prompt
      * PS1 variable controls what is displayed as the prompt on the command line: 
  + Installing Software
    - Package Managers: Two Levels
      * 
      * Low Level Package Manger takes care of the details of unpacking individual packages, running scripts, getting software installed correctly.
      * High Level Package Manager works with groups of packages, download packages from the vendor, and figures out dependencies.
      * 
      * 

Shortcuts

* Ctrl. + L: View location in filesystem explorer.
* Ctrl. + ALT + L: Lock screen. Windows + L equivalent.
* Ctrl. + H: Show hidden files
* Alt. + F2: Open Nautilus
* Ctrl. + F: Search
* Ctrl. + L: In Nautilus will show current location on the top of the screen.

Commands

* rm: delete file
* cp <filename> <newfilename>
* cd + <path>|”..”: Change folder
* sudo passwd <user>: Change the password of user.
* ssh username@remote-server.com: Log in to remote server.