1. The package manager is is a software tool that helps users install, update, and manage software on their system
2. The which command is to determine te path of the command that will be executed when a particular command is invoked
3. To check if a package is installed, I can use: dpkg -l package\_name, dnf list installed package\_name, yum list installed package\_name, pacnam -Q package\_name
4. The apt command is to simplify the management of software packages on the system.
5. I can use the apt command to install packages from the repositories available for my distribution expel: apt install package\_name; apt command allow to update and upgrade some package to the lastest version expel: apt update, apt upgrade; to remove some package apt remove package\_name
6. The yum command is to simplify package management tasks and provide an efficient way to install, update, and remove software packages on RPM-based distributions.
7. I can use yum command to install, updating, removing or searching packages. Example yum install pakage\_name yum update, yum remove package\_name yum search keyword
8. The difference between apt and yum command are: apt is used in Debian-based distribution (Ubuntu, Debian and linux mint) or yum is used in RPM-based distribution (Fedora CentOS, RHEL)
9. Yes, both the apt and yum commands have various option that can be used:

* Common flags for apt: -y or –yes, --no-install-recommends (to install only the main dependencies of a package), --fix-broken (attempt to fix broken dependencies during installation or upgrade), --purge (to remove a package and its configuration file)
* Common flags for yum: -y, --nogpgcheck (disable GPG signature checks for packages), --skip-broken (skip package with unresolved dependencies), --enablerepo=REPOID (enable a specific repository by its repository ID), --exclude=PACKAGE (exclude a specific package from installation or upgrade)

1. The systemctl command is used to manage and control various aspects of the system and its services.
2. The commonly used of systemctl are:

* Managing system services
  + Starting a service: systemctl start service-name
  + Stopping a service: systemctl stop service-name
  + Restarting a service: systemctl restart service-name
  + Reloading configuration of a service without stopping it: systemctl reload service-name
  + Enabling a service to start automatically at boot: systemctl enable service-name
  + Checking the status of a service: systemctl status service-name
* Viewing system logs
  + Viewing the logs of a service: systemctl status service-name -n
  + Following the live logs of a service: systemctl –follow status service-name
* Managing system targets
  + Changing the system target: systemctl isolate target-name
  + Listing available system targets: systemctl list-units –type=target
* Managing system slices
  + Listing system slices: systemctl list-units –type=slice
* Managing timers
  + Starting a timer: systemctl start timer-name
  + Stopping a timer: systemctl stop timer-name
  + Enabling a timer to activate at boot: systemctl enable/disable timer-name
* Managing other system components

1. The service command is to abstract the details of service management and provide a consistent way to interact with services across different distribution
2. Some common use of the service command:

* Starting and stopping services
  + Starting a service: service service-name start
  + Stopping a service: service service-name stop
* Restarting services
  + Restarting a service: service service-name restart
  + Reloading the configuration of a service without stopping it: service service-name reload
* Checking service status
  + Checking the status of a service: service service-name status
  + Displaying the full information of the service: service service-name fullstatus
* Enabling and disabling services
  + Enabling a service to start automatically at boot: service service-name enable
  + Disabling a service to start automatically at boot: service service-name disable
* Listing available service
  + Listing all available services: service –status-all
  + Displaying a list of services that are currently running: service –status-all ! grep running

1. The difference between systemctl and service command are: systemctl is used in Fedora, CentOS/RHEL7 and above, Ubuntu 15.04 and above. Service is used in CentOS/RHEL6 and older, Uuntu14.10 and older
2. The command sloud be used in modern linux of managing service is systemctl
3. To view the status of a service ref. 11) and 13) question
4. The ping command is to measure the round-trip time and packet loss between the source and destination
5. To test network connectivity between the local system and a remote host I type ping IP-address-remote-host
6. 127.0.0.1 is the loopback address or localhost, it is used to refer to the current device
7. The successful response of ping 127.0.0.1 is that the TCP/IP networking protocols are working
8. To check all the port that the server in listening to, I use: netstat -tuln, ss -tuln (all TCP and UDP sockets); lsof -I -p (all open network connections and listening ports)
9. The nslookup command is a network utility used to query the DNS to obtain information about domain names, IP address, and DNS records
10. To troubleshooting network connectivity by ping command I can: - ping the IP address; - ping a domain name instead of an IP address to test DNS resolution; - test connectivity of the port ping -p number-port example.com; - tracing the network path to a destination by using the -R or -T flags
11. Some test editor in linux: nano, vim, emacs, atom. I prefer vim and nano
12. Vim is to provide a powerful and efficient text editor that offers advanced editing capabilities, customization options, and a wide range of features
13. The command line mode in vim allow to enter EX commands (ex command begin with a colon : ) which are powerful command for performing advanced operations; I can execute file-related operations etc.
14. To open a new file in vim for editing: I open vim, I tape :new my-file.txt, to save and exit vim, I press ESC key to switch back to normal mode then enter :wq
15. Difference between the three modes in vim

* Normal mode: to navigate through the text, execute commands, and perform various editing operations; keyboard keys are h, j, k, l to move the cursor left, down, up, and right respectively
* Insert mode is used for inserting and editing text, to enter insert mode from normal mode press the I key or a, A, o, O; is similar to traditional text editor; to exit insert mode press ESC
* Visual mode: allow to visually select and manipulate blocks of text; to enter visual mode press the v key

1. The default mode in vim is the normal mode
2. The I command in vim is to insert text at the current cursor position; the a command allow you to append text after the cursor position; the o allow you to insert the text on the new line; the 0 command allow you to insert text on the new line
3. To copy and paste the text within vim using yamk command:

* Position the cursor: move the cursor to the starting location of the text you want to copy
* Enter visual mode: press v key to enter visual mode
* Select text: use the navigation keys (h j, k, l) to select the desired text
* Yank the text: press the y key to yank (copy) the selected text, the text is to vim’s internal buffer
* Move the cursor: move the cursor to the desired location where you want to paste the text
* Paste the text: press p key to paste the previously yanked text after the current cursor position

1. To delete line in vim: I position the cursor to the line, I go back to the normal mode, the press d key twice in succession to execute the delete command
2. To undo some command in vim I press u or :undo n normal mode to undo the most recent change; to redo command I press ctrl + r or :redo in normal mode to redo the last undone change
3. To select and manipulate block of text using visual mode in vim: press ctrl + v in normal mode to enter in block-wise; I move the cursor to the starting position of desired text block; I use the navigation key to go to the end of the desired block; I span the selected rows and column of text; then I perform the operation
4. To save changes and exit in vim I tape :wq
5. To exit vim without saving the changes I tape :q!
6. The useful command for indentation and formatting in vim are:

* Indentation commands:
  + >> or <<: to increase and decrease the indentation
  + > {motion} or < {motion}: indent or unindent a block of code
  + ==: auto-indent the current line based on the surrounding code
* Formatting commands:
* Tab settings

1. To enable and disable line number I use the :set number and :set nonumber
2. In vim the ESC permit to comeback to the normal mode