

What is Linux?

Linux is an operating system just like Windows, iOS, and Mac OS. In fact, Android, one of the world's most popular platforms, is operated by the Linux operating system. An operating system is a software that manages all of your desktop or laptop's hardware resources. To put it simply, the operating system manages the communication between your software and your hardware. The software would not operate without the operating system (OS).

The History of Linux

Linus Torvalds invented Linux. While still a student at the University of Helsinki, he started developing Linux to create a system similar to MINIX, a UNIX operating system. In 1991 he released version 0.02; Version 1.0 of the Linux kernel, the core of the operating system, was released in 1994. At about the same time, American software developer Richard Stallman and the FSF made efforts to create an open-source UNIX-like operating system called GNU. In contrast to Torvalds, Stallman and the FSF started by creating utilities for the operating system first. These utilities were then added to the Linux kernel to create a complete system called GNU/Linux, or, less precisely, just Linux.

- Linus announces the kernel to the world (1991).
- Release of the first "major" Linux distributions (1993).
- Linux kernel hits 1.0 (1994).
- KDE (1996) / GNOME (1999) projects are released.
- First release of Linux-based Android (1998).
- Kernel development moves to Git (2005).
- Today Linux is everywhere. Supercomputers, smartphones, desktop, web servers, tablets, laptops and home appliances like washing machines, DVD players, routers, modems, cars, refrigerators, etc use Linux OS.

LET'S PRACTICE

When did Linus announced kernel to the world? (1991)

THE COMPONENTS OF LINUX

There are several components of the Linux operating system.

Hardware

Computer hardware is the physical components that a computer system requires to function. It encompasses everything with a circuit board that operates within a PC or laptop; including the motherboard, graphics card, CPU (Central Processing Unit), power supply, and so on.

Boot-loader

The software that manages the boot process of your computer.

Kernel

It is the core part of the operating system and manages the CPU, memory, and peripheral devices. The kernel is the lowest level of the OS.

GNU Core

The GNU Core Utilities are the basic file, shell and text manipulation utilities of the GNU operating system. These are the core utilities that are expected to exist on every operating system.

X server

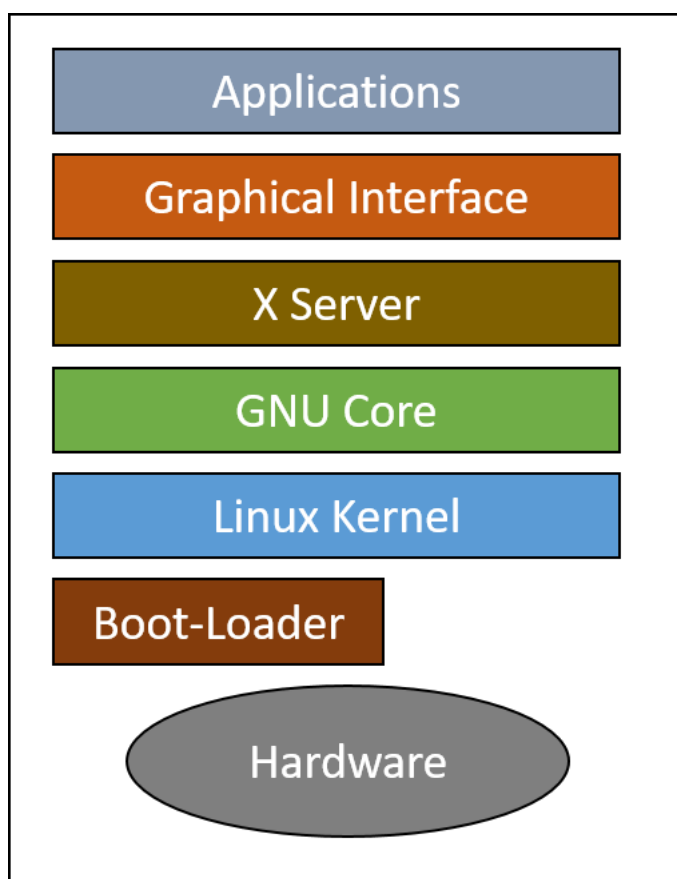
This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just X.

Graphical User Interface

The Linux GUI is often referred to as a Desktop Environment. This is the piece that the users actually interact with. Several desktop environments are available to choose from (GNOME, Cinnamon, Mate, Pantheon, Enlightenment, KDE, Xfce, and so on).

Applications

Desktop environments do not have the full range of applications available. Just like Windows and macOS, Linux provides thousands upon thousands of software titles of high quality that can be easily found and installed.



Popular Linux Distributions

Linux has a number of different versions to suit any type of user. These versions are called distributions (or, in the short form, “distros”).

Linux Mint

Linux Mint is a community-driven Linux distribution based on Ubuntu or Debian that strives to be a "modern, elegant and comfortable operating system which is both powerful and easy to use." Linux Mint provides full out-of-the-box multimedia support by including some proprietary software, such as multimedia codecs, and comes bundled with a variety of free and open-source applications.

Debian

Debian is an independent Linux distribution that was started by the joint association of individuals with the desire to make a free Operating System available to the world. It is designed to be a computing powerhouse capable of running on different types of hardware especially when being used for development.

Ubuntu

Ubuntu is one of the most popular, stable, and best fitted for the newcomers, which is a Debian based Linux distro. It has its own software repositories, which regularly synced with the Debian repository so that all the applications get stable and latest release.

OpenSUSE

openSUSE is a professionally and community-supported Operating System designed to provide field experts with a reliable computing environment for their work. It is distributed in 2 versions, Leap – a Long Term Support (LTS) release, and Tumbleweed – a rolling release, and they both use the YaST package manager for delivering applications to users. openSUSE is an independent Linux distribution that is sponsored by SUSE Linux in collaboration with other companies and it aims to dramatically simplify the process of developing and packaging software for developers and vendors.

Manjaro

Manjaro is an accessible, friendly, open-source Linux distribution and community. Based on Arch Linux, providing all the benefits of cutting-edge software combined with a focus on getting started quickly, automated tools to require less manual intervention, and help readily available when needed.

Fedora

Fedora is a free and open-source Linux distribution built with the aim of encouraging cutting edge technologies by spearheading innovation, working closely with Linux communities, and being amongst the first to integrate new technologies. It is sponsored by the Fedora Project and owned by Red Hat, developed independently of any other Linux distribution.

Red Hat Enterprise Linux (RHEL)

Red Hat Enterprise Linux (RHEL) is a Linux-based operating system from Red Hat designed for businesses. RHEL can work on desktops, on servers, in hypervisors or in the cloud. Red Hat and its community-supported counterpart, Fedora, are among the most widely used Linux distributions in the world.

Which of these Linux distributions appeared first? (Debian)

Linux Embedded Systems

Embedded System

An embedded system is a computer system that is dedicated to one or two specific functions. This system is embedded as part of a complete computer system including hardware such as mechanical and electrical components.

Embedded Linux

Embedded Linux is a type of Linux operating system/kernel that is designed to be installed and used within embedded devices and appliances. It is a compact version of Linux that offers features and services in line with the operating and application requirement of the embedded system.

Android OS is a type of embedded Linux, customized to be used on smartphones. Many systems that use Linux embedded are:

- Smart TVs
- Tablet PCs
- Navigation devices
- Wireless routers
- Other industrial and consumer electronic equipment

What is open source?

Open-source software is software with source code that anyone can inspect, modify, and enhance.

"Source code" is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works. Programmers with access to the source code of a computer program can enhance the system by adding features to it or repairing sections that don't always work properly.

How does open-source software vary from other types of software?

Most software has source code that can be changed only by the individual, team, or organization that developed it — and retains sole control over it. That type of software is called "proprietary" or "closed source" software.

Only the original authors of proprietary software can legally copy, inspect, and alter that software. And in order to use proprietary software, computer users must agree (usually by signing a license displayed the first time they run this software) that they will not do anything with the software that the software's authors have not expressly permitted. Microsoft Office and Adobe Photoshop are examples of proprietary software.

Open-source software is different. Its authors make its source code available to others who would like to view that code, copy it, learn from it, alter it, or share it. LibreOffice and the GNU Image Manipulation Program are examples of open-source software.

What is open source software? (Software where the code is available)

Open-source Licensing

Open source licenses are licenses that comply with the Open Source Definition — in brief, they allow the software to be freely used, modified, and shared. To be approved by the Open Source Initiative (also known as the OSI), a license must go through the Open Source Initiative's license review process.

Types of open source license

There are generally two categories: permissive licenses and copyleft licenses.

A license that is permissive is simple and the most basic form of open source license. It allows you to do whatever you want with the software, as long as you abide by the requirements for notification. Permissive licenses provide the software as-is, with no warranties. So permissive licenses can be summarized as follows:

- Do whatever you want with the code
- Use at your own risk
- Acknowledge the author/contributor

Copyleft licenses add requirements to the permissive license. In addition to the requirements listed above, copyleft licenses also require that:

- If you distribute binaries, you must make the source code for those binaries available
- The source code must be available under the same copyleft terms under which you got the code
- You cannot place additional restrictions on the licensee's exercise of the license

Popular Licenses

The following OSI-approved licenses are popular, widely used, or have strong communities:

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License version 2.0
- Free Software Foundation (FSF)
- The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom. Free software is about having control over the technology we use in our homes, schools, and businesses, where computers work for our individual and communal benefit, not for proprietary software companies or governments who might seek to restrict and monitor us. The Free Software Foundation exclusively uses free software to perform its work.
- The Free Software Foundation is working to secure freedom for computer users by promoting the development and use of free (as in freedom) software and documentation—particularly the GNU operating system—and by campaigning against threats to computer user freedom like Digital Restrictions Management (DRM) and software patents.
- Open Source Initiative (OSI)
- The Open Source Initiative (OSI) is a non-profit organization dedicated to the promotion of open-source software. OSI was founded in 1998 by Bruce Perens and Eric Raymond. OSI is quite distinct from the Free Software Foundation (FSF) led by Richard Stallman. Although they have similar history and motivation, OSI considers its ends as more pragmatic and business-driven, while FSF is based on anti-establishment and moralistic viewpoints. The OSI is actively engaged in building open source community, public advocacy, education, and promoting awareness regarding the significance of non-proprietary or open-source software. In order to establish an open-source environment around the world, OSI preserves and supports the Open Source Definition and also provides the OSI-Certified Open Source Software Certification Program. To achieve this OSI certification, the software should be

distributed using a license that ensures the legal right to freely read, use, modify, and re-distribute the software.

View the Lesson (Major Open-Source Applications)

Desktop Applications

Firefox

Mozilla Firefox, also known as the Firefox browser, or simply Firefox, is a free and open-source web browser developed by the Mozilla Foundation and its subsidiary, Mozilla Corporation. It is introduced in 2004 as part of the Mozilla Application Suite. Firefox included almost all the features offered by other browsers at that time, as well as a number of new features, including a bookmarks toolbar and tabbed browsing.

Thunderbird

Thunderbird is a free and open-source email, newsfeed, chat, and calendaring client, that's easy to set up and customize.

LibreOffice

LibreOffice is community-driven and developed software, and is a project of the not-for-profit organization, The Document Foundation. LibreOffice is free and open-source software, originally based on OpenOffice.org (commonly known as OpenOffice), and is the most actively developed OpenOffice.org successor project. LibreOffice includes several applications that make it the most versatile Free and Open Source office suite on the market:

- Writer (word processing)
- Calc (spreadsheets)
- Impress (presentations)
- Draw (vector graphics and flowcharts)
- Base (databases)
- Math (formula editing)

GIMP

GIMP is an acronym for GNU Image Manipulation Program. It is a freely distributed program for such tasks as photo retouching, image composition, and image authoring.

Server Applications

Apache Web Server

Apache Web Server is an open-source web server creation, deployment, and management software. Initially developed by a group of software programmers, it is now maintained by the Apache Software Foundation. Apache Web Server is designed to create web servers that can host one or more HTTP-based websites. Notable features include the ability to support multiple programming languages, server-side scripting, an authentication mechanism, and database support. Apache Web Server can be enhanced by manipulating the code base or adding multiple extensions/add-ons.

NGINX

NGINX is open-source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for HTTP, TCP, and UDP servers.

MySQL

MySQL was a free-software database engine originally developed and first released in 1995. It was originally produced under the GNU General Public License, in which source code is made freely available.

MySQL is very popular for Web-hosting applications because of its plethora of Web-optimized features like HTML data types, and because it's available for free. It is part of the Linux, Apache, MySQL, PHP (LAMP) architecture, a combination of platforms that are frequently used to deliver and support advanced Web applications. MySQL runs the back-end databases of some famous websites, including Wikipedia, Google, and Facebook- a testament to its stability and robustness.

Samba

Samba is an open-source software suite that runs on Unix/Linux based platforms but can communicate with Windows clients like a native application. So Samba can provide this service by employing the Common Internet File System (CIFS). At the heart of this CIFS is the Server Message Block (SMB) protocol.

Samba performs:

- File & print services
- Authentication and Authorization
- Name resolution
- Service announcement (browsing)

Owncloud

ownCloud is a client-server suite of applications for creating and using file hosting service. The functionality of ownCloud is similar to dropbox. However, your files are stored on your connected hardware.

Development Languages

Shell

Shell is a command language interpreter that executes commands read from the standard input device such as a keyboard or from a file. A shell script is a list of commands in a computer program that is run by the Unix shell. The most common Linux shell is named Bash. The name is an acronym for Bourne-again shell. Bash (like many other shells) has the ability to run an entire script of commands, known as a "Bash shell script" (or "script").

C

C is a high-level and general-purpose programming language used for a wide range of applications from Operating systems like Windows and iOS to software that is used for creating 3D movies.

Java

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. Java is Platform Independent, portable, robust, and dynamic, with the ability to fit the needs of virtually any type of application.

JavaScript

JavaScript is a high-level, cross-platform, object-oriented computer programming language. It is also one of the core technologies of the web, along with HTML and CSS. JavaScript is used to create client-side dynamic web pages. Java and javascript are completely different and distinct languages.

Perl

Perl is a highly capable, feature-rich programming language with over 30 years of development. Perl runs on over 100 platforms from portables to mainframes and is suitable for both rapid prototyping and large scale development projects.

Python

Python is a high-level programming language, with applications in numerous areas, including web programming, scripting, scientific computing, and artificial intelligence. Python has extensive object-oriented programming support with a clean and consistent syntax.

PHP

PHP is a popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.

Package Management Tools

Contemporary distributions of Linux-based operating systems install software in pre-compiled packages, which are archives that contain binaries of software, configuration files, and information about dependencies. Furthermore, package management tools keep track of updates and upgrades so that the user doesn't have to hunt down information about bug and security fixes. Without package management, users must ensure that all of the required dependencies for a piece of software are installed and up-to-date, compile the software from the source code (which takes time and introduces compiler-based variations from system to system), and manage configuration for each piece of software. Without package management, application files are located in the standard locations for the system to which the developers are accustomed, regardless of which system they're using.

Package management systems attempt to solve these problems and are the tools through which developers attempt to increase the overall quality and coherence of a Linux-based operating system.

dpkg

On Linux operating systems that use Debian package management, the dpkg command queries, installs, removes, and maintains Debian software packages and their dependencies.

apt-get

APT stands for the Advanced Packaging Tool. apt-get is a command-line tool that helps in handling packages in Linux. Its main task is to retrieve the information and packages from the authenticated sources for installation, upgrade and removal of packages along with their dependencies.

rpm: Red Hat Package Manager

RPM Package Manager (also known simply as RPM), originally called the Red-hat Package Manager, is a program for installing, uninstalling, and managing software packages in Linux. It is an open packaging system, which runs on Red Hat Enterprise Linux as well as other Linux and UNIX systems.

yum: yellowdog updater modified

Yum is an automatic updater and package installer/remover for rpm systems. It automatically computes dependencies and figures out what things should occur to install packages. It makes it easier to maintain groups of machines without having to manually update each one using rpm. Yum has a plugin interface for adding simple features. Yum can also be used from other python programs via its module interface.

How to install and run Linux?

Windows Subsystem for Linux Installation Guide for Windows 10

1- Install the Windows Subsystem for Linux

Before using Windows Subsystem for Linux, you must enable the “Windows Subsystem for Linux” optional feature. For this:

- Open PowerShell as an Administrator and type the command below.

Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

- Restart your computer.

2- Download and Install a Linux Distribution

There are three ways to download & install a Linux Distribution:

- Microsoft Store
- Command-Line/Script (read the manual installation instructions)
- Manually (for Windows Server – instructions here)

Downloading and Installing using the Microsoft Store

Please be sure that your Windows build is 16215 or later. You can check your build from this link.

1. Open Microsoft Store and select a Linux distribution. The following links are the related pages for each distribution:
 - Ubuntu 20.04 LTS
 - Ubuntu 16.04 LTS
 - Ubuntu 18.04 LTS
 - OpenSUSE Leap 15
 - OpenSUSE Leap 42
 - SUSE Linux Enterprise Server 12
 - SUSE Linux Enterprise Server 15
 - Kali Linux
 - Debian GNU/Linux
 - Fedora Remix for WSL
 - Pengwin
 - Pengwin Enterprise
 - Alpine WSL

2. After opening the distribution page, select “Get”



3. Initialize your distribution

After you install your Linux distro, you should initialize the new distro instance once, before it can be used.

4. How to run Linux CENTOS 7 online without installing?
5. There are several online options where you can run Linux distributions. In this tutorial, we will run Linux CENTOS 7 on the website <https://linuxzoo.net/>
6. With this online option, you can learn Linux using a remote Linux machine with root access in a very safe way.
7. Step-1: Open <https://linuxzoo.net/>

Step-1: Open <https://linuxzoo.net/>



Welcome to linuxzoo

Learn Linux from the safety of your chair using a remote private linux machine with root access.

- Welcome to linuxzoo
- [Our environment](#)
- [Essential Linux](#)
- [System Administration](#)

Status: Normal. Everything should be stable.

Look at the [Our Environment](#) link, and then [Running Your Machine](#) for getting started.

Quick start hints: register/login. Join Queue. Switch On (in Control tab). Wait for successful boot, click the Connect tab, and then click "telnet: linuxzoo.net" (or type `telnet linuxzoo.net` at your command prompt). Username `root`, password `secure`.

Image	Username	Password
Linux Centos 7	root	secure
	alice	secure
Caine Forensics 10.0	caine	caine

FAQ for VNC: There are a few options to getting a remote graphical desktop. In "connected" you can click on Java VNC, which requires java 7 installed on your machine. JavaScript VNC is more flexible, but it may be slower (it is experimental). Some systems do not like you logging in graphically as root.

Login Details

Email:

Password:

Login | Register for an account

NOT currently queued

Refresh in: 2:00

0 user(s) ahead in queue.

Queue Stats

You must register and login before you can use a virtual machine

Step-2: Read the registering instructions

User registration - READ THIS !!

To be allowed access to all the virtual machines you must become a registered user. Please make sure the information you give is accurate. Email and other data is NEVER used for junk or spam generation, but may be used to ask for feedback on linuxzoo.

- Username - this MUST be a valid email address, e.g. 00123456@domain.ac.uk.
- Password - type the same password into BOTH password fields.
- First and last name.
- Module information - Needed if you are a Napier student and you want to graduate! Non-napier users should leave this blank.
- Matriculation number - as with module information above
- Auth Code - If you have one from your module leader this will give you more priority on linuxzoo. Leave it blank otherwise.

The user account is usable immediately, but is deleted after a while unless you confirm your registration. The system sends you an email with a web link. Click on this link and your account will not be deleted.

To proceed to registration [click here](#) 

Step-3: Fill in the registration form

← → ↻ linuxzoo.net/vm/usercontrol.cgi

User registration

Email/Username	<input type="text"/>
Password	<input type="password"/> (needed if changing the password)
Password (again)	<input type="password"/> (needed if changing the password)
First Name	<input type="text"/>
Last Name	<input type="text"/>
Matriculation No	<input type="text"/> Leave Matriculation No blank if you are NOT A STUDENT
Programme info	<input type="text" value="Select One"/> Select "Just Interested" if you are NOT A STUDENT
Auth Code	<input type="text"/> Leave Auth Code blank if not known

I have read and accept the [privacy policy](#) and the [terms and conditions](#): ☐ Accept

Step-4: Log in the system with your credentials

Minimize

Login Details

Email:

Password:

Login Account Links

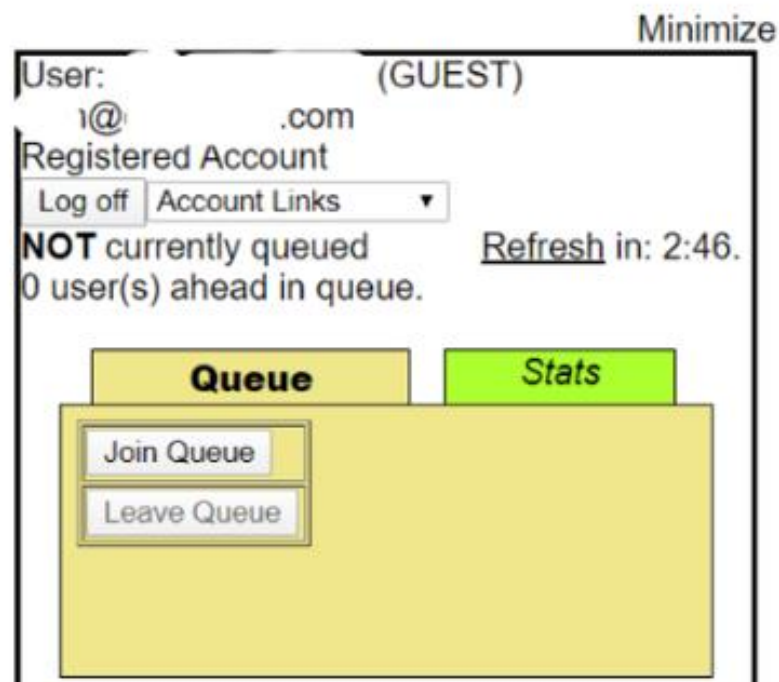
NOT currently queued ☐ [Refresh](#) in: 2:44.

0 user(s) ahead in queue.

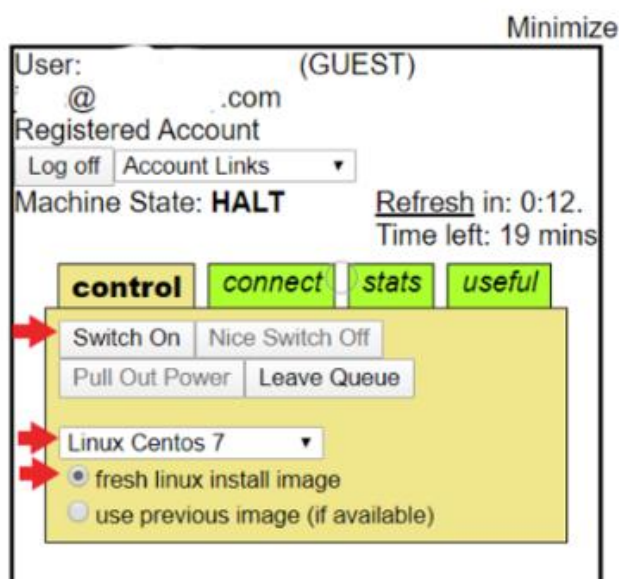
Queue **Stats**

You must register and login before you can use a virtual machine

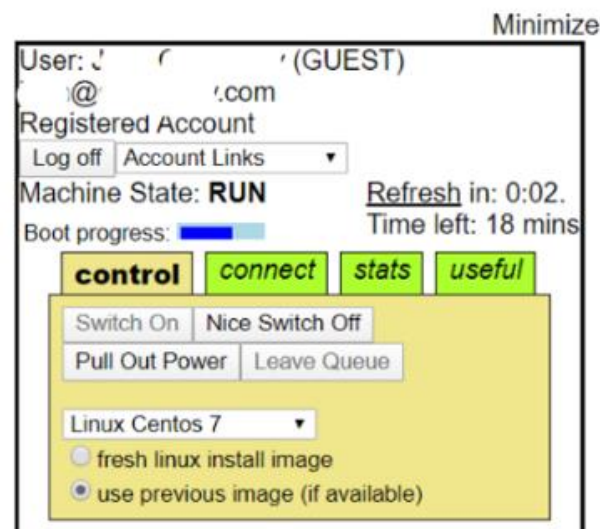
Step-5: After logging in, click “Join Queue” button



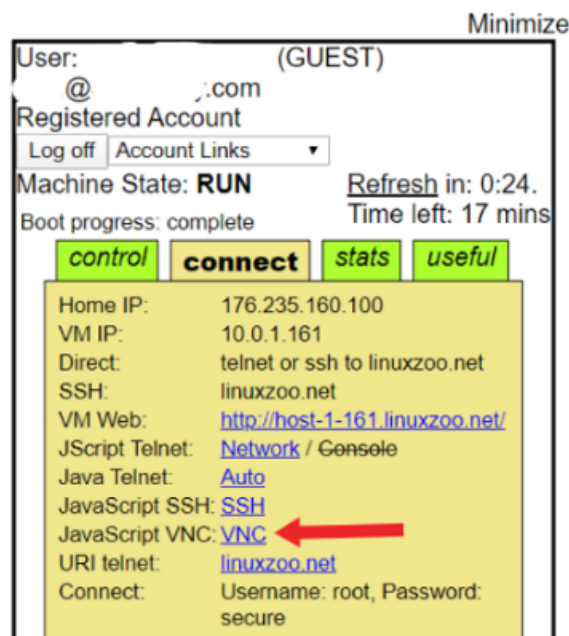
Step-6: In the control tab, Select “Linux CENTOS 7” and “fresh Linux install image” from the options and click the “Switch On” button



Step-7: You will see the machine state running and wait for the boot progress



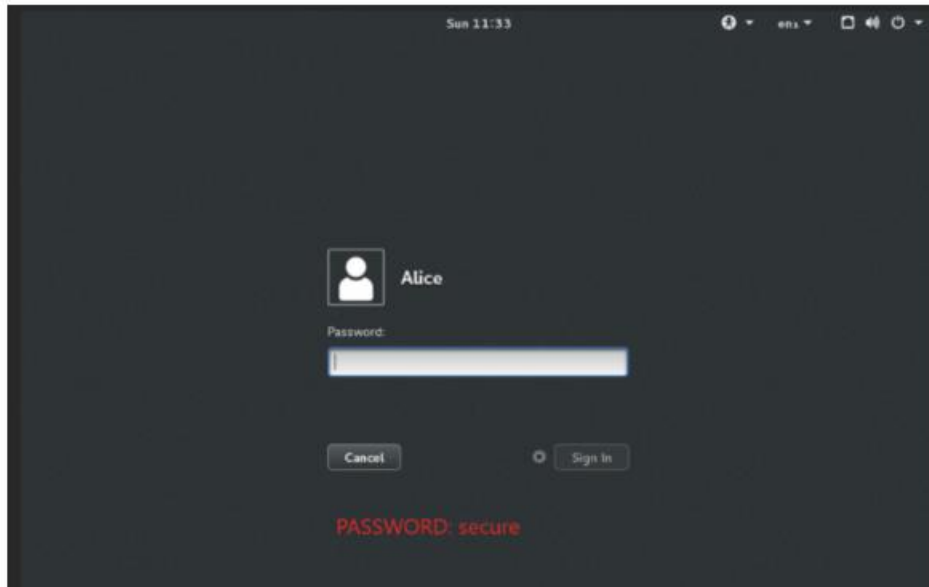
Step-8: After the boot progress is completed, copy the link "Javascript VNC" and open that link in a new browser



Step-9: You will see Linux CENTOS has started and waiting for you to log in. The login credentials are:

username: Alice

password: secure



How to use Linux with Docker Container

Linux can be run inside containers. The best practice to run linux is to use Docker. You can use Docker to run Linux Containers without installing Linux on your computer.

Before starting this tutorial, you should read the Docker documentation and install Docker. (<https://docs.docker.com>). Be sure that your system supports running Docker.

After you have installed Docker, select one of the Linux images from Docker Hub (<https://docs.docker.com>)

The official images are strongly recommended to be used for base images. The most used Linux images are :

- ubuntu
- debian
- centos
- fedora

You can run Linux by running this command via your bash or any other supported CLI:

```
$ docker run -it --rm <docker_image_name> bash
```

You can log out from the container with the “exit” command. To stop the container, you can run the following commands:

```
$ docker kill < docker_image_name >
```



```
$ docker rm -v < docker_image_name >
```

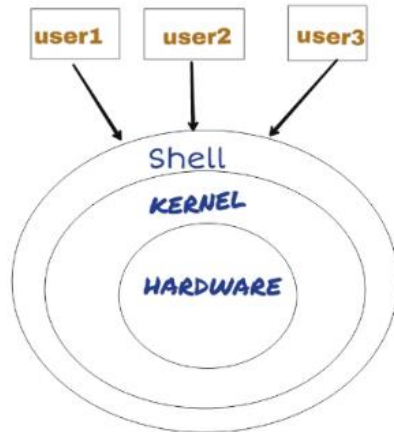
To remove all stopped containers run the following:

```
$ docker rm $(docker ps -a -q)
```

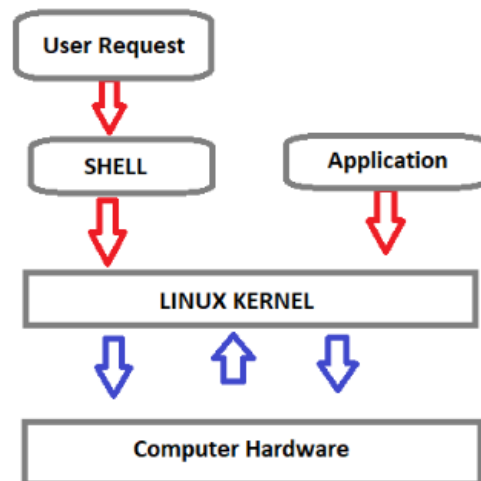
[View the Lesson \(Command Line Basics\)](#)

What is the "Shell"?

- Shell is an interface between an end-user and the Linux system.



- In other words, Shell is a program that receives commands from the user, relays them to the operating system to process and displays the output. Shell is one of the main parts of Linux OS. Each Linux distro comes with a GUI (Graphical User Interface), but essentially Linux has a CLI (Command-Line Interface).



Shell Types

C Shell : If you are using a C-type shell, the `%` character is the default prompt.

- C shell (csh)
- TENEX/TOPS C shell (tcsh)

Bourne Shell : If you are using a Bourne-type shell, the `$` character is the default

Shell Types

C Shell : If you are using a C-type shell, the `%` character is the default prompt.

- C shell (csh)
- TENEX/TOPS C shell (tcsh)

Bourne Shell : If you are using a Bourne-type shell, the `$` character is the default prompt.

- Bourne shell (sh)
- Korn shell (ksh)
- Bourne Again shell (bash)
- POSIX shell (sh)

An enhanced version of SH is called BASH (which stands for **B**ourne **A**gain **S**hell) and serves as the main shell program on the most Linux systems.

- The standard Linux shell (BASH) is both a command-line interpreter and a programming language.



The most common interpreter is BASH or the Bourne Again Shell, but there are others available as well and some of them does not use the dollar sign.

Command Prompt

The command prompt at the beginning of the command line is a short text string. The command prompt for Linux generally shows the current user, the current host, and the appropriate directory.

The command prompt is easily modified to display as desired with more or less information. At the end of the prompt list, the `$`(dollar sign) signifies the current user being unprivileged.

- Dollar sign (\$) means you are a normal user and indicates you are logged in with the normal permissions.

- Hash (#) means you are the system administrator (root) and indicates you are logged in with root privileges.

- The "root" account on a Linux computer is the account with full privileges.
- Root access is often necessary for performing commands in Linux, especially commands that affect system files. Because root is so powerful, it's recommended to only request root access when necessary, as opposed to logging in as the root user.

Basic Shell Commands

This is a list of most frequently used Linux commands.

- The **~** (**tilde**) symbol stands for your home directory.
- The **pwd** (stands for print working directory) command will allow you to know in which directory you're located.
- The **ls** command will show you the list of folders and files in your current directory.
- The **cp** command will make a copy of a file.
- The **cd** command will allow you to change directories.
- The **rm** command removes or deletes a file in your directory.
- The **rmdir** command will delete an empty directory.
- The **mkdir** command will allow you to create directories
- The **mv** command will move a file to a different location or will rename a file.
- **cd** - Navigate to the last directory you were working in.
- **cd ~** or just **cd** Navigate to the current user's home directory.
- **cd ..** Go to the parent directory of current directory (mind the space between **cd** and **..**)

Quoting

Quoting is used to disable special treatment of certain characters and words, as well as to prevent parameter expansion and preserve what is quoted.

The bash shell knows rare, special characters like **\$ (dollar sign)**, which is used to extend the value of the element. For example **\$PATH** is used to extend the value of **PATH** element which is predefined variable in bash to hold system paths as shown below.

```
clarusway@f85a0c1549f4:~$ echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

The special character **\$** can also be used with the user's custom element. For example, in the bash shell, the user can define an element like **greeting="Hello from Clarusway"** and extend the value of this element as shown below.

```
clarusway@f85a0c1549f4:~$ greeting="Hello from Clarusway"
clarusway@f85a0c1549f4:~$ echo $greeting
```

Hello from Clarusway

There are three types of quotes:

1. **Double Quotes** : The double quote " preserve the literal value of most characters contained within the quotes, exceptions include **\$ (for variables)**, **' (for single quoting)**, **\ (for escaping a character)** .

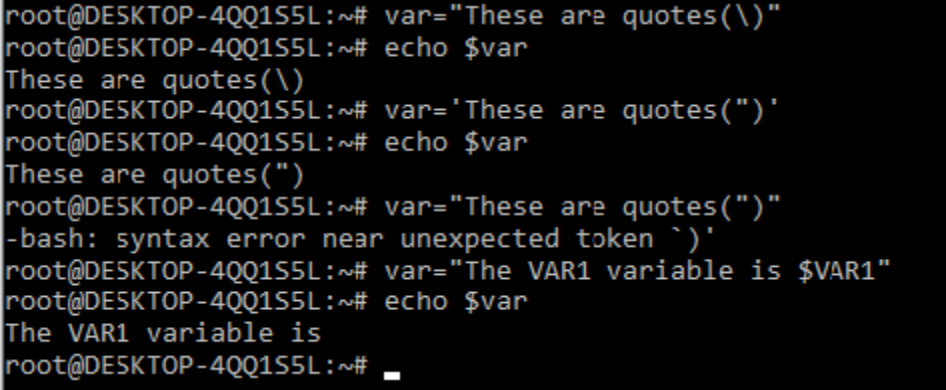
```
clarusway@f85a0c1549f4:~$ echo $SHELL
/bin/bash
clarusway@f85a0c1549f4:~$ echo "$SHELL"
/bin/bash
clarusway@f85a0c1549f4:~$ echo "path to shell $SHELL"
path to shell /bin/bash
```

2. **Single Quotes**: The single quote ('quote') protects everything enclosed between single quotation marks.

```
clarusway@f85a0c1549f4:~$ echo $SHELL
/bin/bash
clarusway@f85a0c1549f4:~$ echo '$SHELL'
$SHELL
clarusway@f85a0c1549f4:~$ echo 'path to shell $SHELL'
path to shell $SHELL
```

3. **Backslash:** Use the backslash to change the special meaning of the characters or to escape special characters within the text such as quotation marks.

```
clarusway@f85a0c1549f4:~$ echo "\"$SHELL"
$SHELL
clarusway@f85a0c1549f4:~$ echo "path to shell \"$SHELL"
path to shell $SHELL
```



```
root@DESKTOP-4QQ1S5L:~# var="These are quotes(\)"
root@DESKTOP-4QQ1S5L:~# echo $var
These are quotes(\)
root@DESKTOP-4QQ1S5L:~# var='These are quotes("'
root@DESKTOP-4QQ1S5L:~# echo $var
These are quotes(")
root@DESKTOP-4QQ1S5L:~# var="These are quotes(")"
-bash: syntax error near unexpected token `)'
root@DESKTOP-4QQ1S5L:~# var="The VAR1 variable is $VAR1"
root@DESKTOP-4QQ1S5L:~# echo $var
The VAR1 variable is
root@DESKTOP-4QQ1S5L:~#
```

File Permission

Linux File Ownership

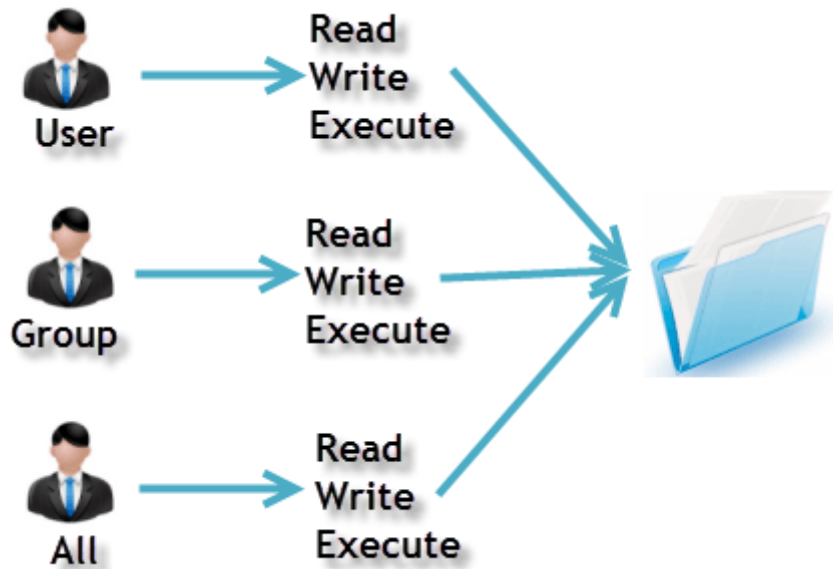
Each file and directory on Linux system has 3 types of owners assigned,

- **User :** A user is the owner of the file.
- **Group :** A user- group can contain multiple users.
- **Other/All :** Any other user who has access to a file.

Permission

Each file and directory on your Linux system has 3 permissions defined for all the 3 owners.

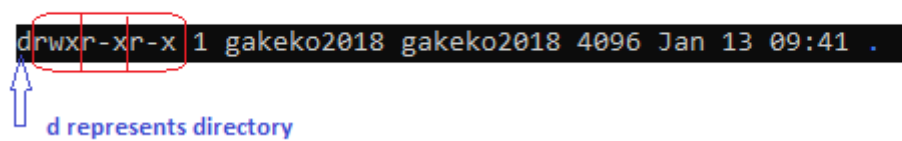
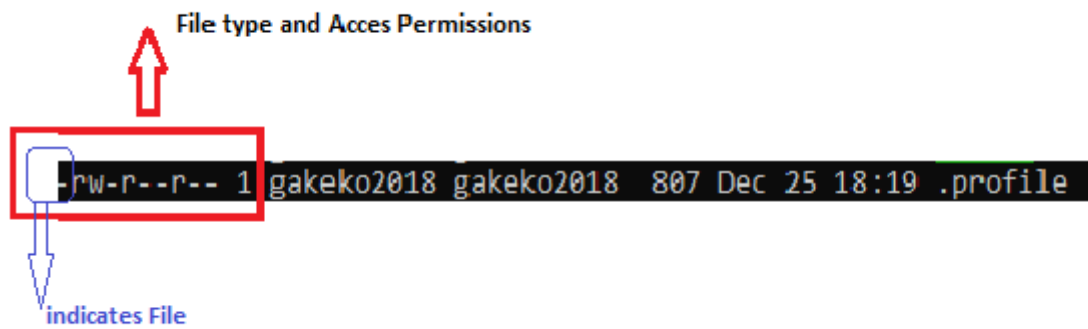
- **Read :** The read permission gives you the authority to open and read a file.
- **Write :** The write permission gives you the authority to modify the contents of a file.
- **Execute :** In Linux, you cannot run a program unless the execute permission is set.



ls

ls -a

```
gakeko2018@DESKTOP-JA07K2U:~$ ls
cert.pem
gakeko2018@DESKTOP-JA07K2U:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .local  .profile  .ssh  cert.pem
gakeko2018@DESKTOP-JA07K2U:~$ ls -al
total 12
drwxr-xr-x 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .
drwxr-xr-x 1 root      root      4096 Dec 25 18:19 ..
-rw-r--r-- 1 gakeko2018 gakeko2018 236 Jan 14 12:21 .bash_history
-rw-r--r-- 1 gakeko2018 gakeko2018 220 Dec 25 18:19 .bash_logout
-rw-r--r-- 1 gakeko2018 gakeko2018 3771 Dec 25 18:19 .bashrc
drwxrwxrwx 1 gakeko2018 gakeko2018 4096 Jan 13 09:38 .local
-rw-r--r-- 1 gakeko2018 gakeko2018 807 Dec 25 18:19 .profile
drwx----- 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .ssh
-r----- 1 gakeko2018 gakeko2018 1675 Jan 13 09:38 cert.pem
```



r = read permission

w = write permission

x = execute permission

- = no permission



Changing Permission with chmod Command

We can use the **chmod** command which stands for **change mode**.

Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

chmod permissions filename

The table below gives for all for permissions types.

Symbol	Permission Type
---	No Permission
--x	Execute
-w-	Write
-wx	Execute+Write
r--	Read
r-x	Read+Execute
rw-	Read+Write
rwX	Read+Write+Execute

Example

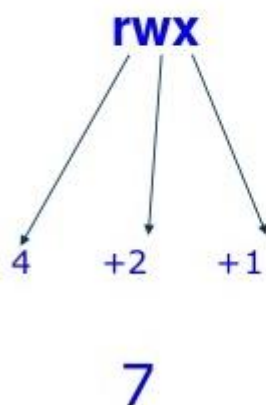
```

root@DESKTOP-4QQ1S5L:~# ls -l
total 0
-rw-rw-rw- 1 root root 0 Dec 29 17:53 file1
-r--r--rwx 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~# chmod 754 file2
root@DESKTOP-4QQ1S5L:~# ls -l file2
-rwxr-xr-- 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~#

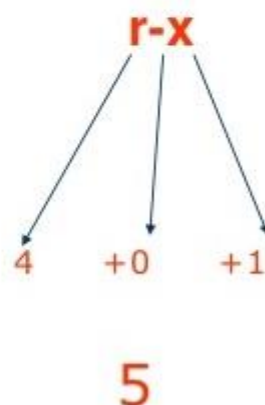
```

Permissions

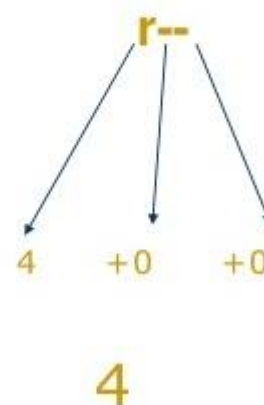
read=4;



write=2;

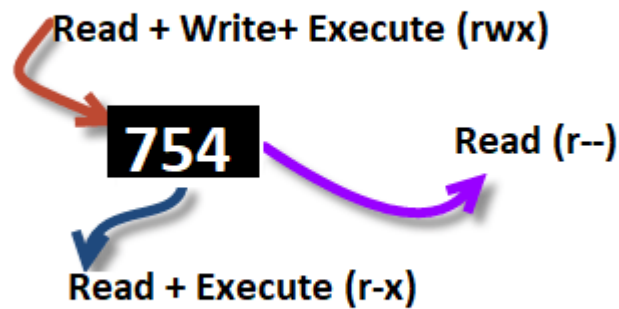


execute=1



754 code says;

- Owner can read, write and execute
- Usergroup can read and execute
- Other can only read



Ping & SSH Command

Ping Command

Ping or Packet Internet Groper is a network administration utility used to check the connectivity status between a source and a destination computer/device over an IP network. It also helps you assess the time it takes to send and receive a response from the network.

ping host-name/IP

Example

ping 54.93.34.220

Press CTRL+C (in MacOS => CMD+C) to exit.

```
gakeko2018@DESKTOP-JA07K2U:~$ ping 54.93.34.220
PING 54.93.34.220 (54.93.34.220) 56(84) bytes of data.
64 bytes from 54.93.34.220: icmp_seq=1 ttl=243 time=62.6 ms
64 bytes from 54.93.34.220: icmp_seq=2 ttl=243 time=93.5 ms
64 bytes from 54.93.34.220: icmp_seq=3 ttl=243 time=66.8 ms
64 bytes from 54.93.34.220: icmp_seq=4 ttl=243 time=67.6 ms
64 bytes from 54.93.34.220: icmp_seq=5 ttl=243 time=62.7 ms
64 bytes from 54.93.34.220: icmp_seq=7 ttl=243 time=84.6 ms
64 bytes from 54.93.34.220: icmp_seq=8 ttl=243 time=64.6 ms
64 bytes from 54.93.34.220: icmp_seq=9 ttl=243 time=72.0 ms
```

SSH

- **ssh** stands for “Secure Shell”.
- It is a protocol used to securely connect to a remote server/system.
- **ssh** is secure in the sense that it transfers the data in an encrypted form between the host and the client

ssh user@host(IP/Domain_name)
Example

```
ssh -i cert.pem ec2-user@54.93.34.220
```

```
gakeko2018@DESKTOP-JA07K2U:~$ ssh -i cert.pem ec2-user@54.93.34.220
The authenticity of host '54.93.34.220 (54.93.34.220)' can't be established.
ECDSA key fingerprint is SHA256:lvCnUtJiig4s2U4aojBonZOSbzGPBMOpB9yPPoGjVEo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.93.34.220' (ECDSA) to the list of known hosts.

  _ | _ | _ |
  _ | ( _ | _ | /
  _ | \ _ | _ |

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-35-15 ~]$
```

whoami Command

Displays user, group and privileges information for the user who is currently logged on to the local system.

- whoami command is used both in Linux Operating System and as well as in Windows Operating System.
- It is basically the concatenation of the strings “who”, ”am”, ”i” as whoami.

whoami

Example:

```
clarusway@f85a0c1549f4:~$ whoami
```

```
clarusway
```

Using the Command Line to Get Help

Man Pages

A man page (short for manual page) is a form of software documentation usually found on a Unix or Unix-like operating system.

Man pages are the traditional package documentation for application usage. What that means is that they're typically installed when you install a package. So if we install a package to do some task, the man page for that package will typically be installed at the same time. This gives us the ability to take a look at that documentation and make sure that we're using it in a manner consistent with its design.

Usage

The man page for a particular command is invoked by preceding the command with man.

```
man <command>
```


man ls

```
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  list information about the FILES (the current directory by default). Sort entries alphabetically if none of
  -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      with -l, print the author of each file

  -b, --escape
      print C-style escapes for nongraphic characters

  --block-size=SIZE
      scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes in units of 1,048,576
      bytes; see SIZE format below

  -B, --ignore-backups
      do not list implied entries ending with ~

  -c      with -lt: sort by, and show, ctime (time of last modification of file status information); with -l:
          show ctime and sort by name; otherwise: sort by ctime, newest first

  -C      list entries by columns

  --color[=WHEN]
      colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

  -d, --directory
      list directories themselves, not their contents

Manual page ls(1) line 1 (press h for help or q to quit)
```

NAME

Program or Function name(s) followed by descriptions of functionality.

SYNOPSIS

A short overview of available options

DESCRIPTION

Detailed information about arguments and options.

```
-Z, --context
    print any security context of each file

-l      list one file per line. Avoid '\n' with -q or -b

--help  display this help and exit

--version
    output version information and exit

The SIZE argument is an integer and optional unit (example: 10K is 10*1024). Units are K,M,G,T,P,E,Z,Y (pow-
ers of 1024) or KB,MB,... (powers of 1000).

Using color to distinguish file types is disabled both by default and with --color=never. With --color=auto,
ls emits color codes only when standard output is connected to a terminal. The LS_COLORS environment variable
can change the settings. Use the dircolors command to set it.

Exit status:
  0      if OK,
  1      if minor problems (e.g., cannot access subdirectory),
  2      if serious trouble (e.g., cannot access command-line argument).

AUTHOR
  Written by Richard M. Stallman and David MacKenzie.

REPORTING BUGS
  GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
  Report ls translation bugs to <http://translationproject.org/team/>

COPYRIGHT
  Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later
  <http://gnu.org/licenses/gpl.html>.
  This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent per-
  mitted by law.

SEE ALSO
  Full documentation at: <http://www.gnu.org/software/coreutils/ls>
  or available locally via: info '(coreutils) ls invocation'

GNU coreutils 8.28                                January 2018                    LS(1)
Manual page ls(1) line 178/220 (END) (press h for help or q to quit)
```

Info Pages

Info pages are additional documentation with more robust capability in detail. Info Page normally provides more detailed information about a command than its respective man page. Additionally, Info uses a structure for linking these pages together, and they may be assembled into a larger collection.

Usage

The info page for a particular command is invoked by preceding the command with info.

info <command>

info echo

```
text: printf invocation, Up: Printing text

5.1 'echo': Print a line of text
=====

'echo' writes each given STRING to standard output, with a space between
each and a newline after the last one.  Synopsis:

    echo [OPTION]... [STRING]...

Due to shell aliases and built-in 'echo' functions, using an
unadorned 'echo' interactively or in a script may get you different
functionality than that described here.  Invoke it via 'env' (i.e., 'env
echo ...') to avoid interference from the shell.

The program accepts the following options.  Also see *note Common
options::.  Options must precede operands, and the normally-special
argument '--' has no special meaning and is treated like any other
STRING.

-n'
    Do not output the trailing newline.

-e'
    Enable interpretation of the following backslash-escaped characters
    in each STRING:

    '\a'      alert (bell)
    '\b'      backspace
    '\c'      produce no further output
    '\e'      escape
    '\f'      form feed
    '\n'      newline
    '\r'      carriage return
    '\t'      tab

-----Info: (coreutils)echo invocation, 78 lines --Top-----
Welcome to Info version 6.5.  Type H for help, h for tutorial.
```

The main difference between Man and Info is the amount of content that they have; Info contains a whole lot more than Man does.

If no info page exists, info can pull documentation from the man page.

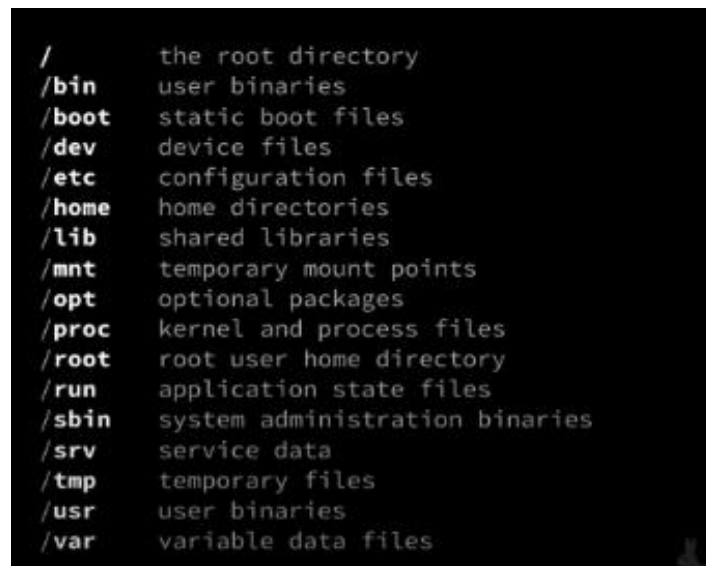
Using Directories and Listing Files

Files and Directories

The file system hierarchy standard (FHS) defines the structure of the file systems on Linux.

In the FHS, all files and directories appear under the root directory / , even if they are stored on different physical or virtual devices.

Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS, and are not considered authoritative for platforms other than Linux.



/	the root directory
/bin	user binaries
/boot	static boot files
/dev	device files
/etc	configuration files
/home	home directories
/lib	shared libraries
/mnt	temporary mount points
/opt	optional packages
/proc	kernel and process files
/root	root user home directory
/run	application state files
/sbin	system administration binaries
/srv	service data
/tmp	temporary files
/usr	user binaries
/var	variable data files

/ (Root) : Primary hierarchy root and root directory of the entire file system hierarchy. Every single file and directory starts from the root directory.

/bin : Essential command binaries that need to be available in single user mode.

/boot : Boot loader files.

/dev : Essential device files.

/etc : Host-specific system-wide configuration files.

/home : Users' home directories, containing saved files. Home directories for all users to store their personal files

/lib : Libraries essential for the binaries in /bin/ and /sbin/

/mnt : Temporarily mounted filesystems.

/opt : Optional application software packages.

/proc : Contains information about system process.

/sbin : Essential system binaries, e.g., fsck, init, route.

/srv : Site-specific data served by this system, such as data and scripts for web servers, data offered by FTP servers, and repositories for version control systems. srv stands for service.

/tmp : Temporary files. Often not preserved between system reboots, and may be severely size restricted.

/usr : Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.

Hidden Files and Directories

Files and directories can be hidden from the basic listing. Files and directories in Linux may be set as hidden by preceding their name with a . (a single period).

```
root@DESKTOP-4QQ1S5L:~# ls
root@DESKTOP-4QQ1S5L:~# ls -a
.  ..  .bash_history  .bashrc  .profile  .viminfo
root@DESKTOP-4QQ1S5L:~# touch .file5
root@DESKTOP-4QQ1S5L:~# ls -a
.  ..  .bash_history  .bashrc  .file5  .profile  .viminfo
root@DESKTOP-4QQ1S5L:~#
```

Hidden files can be listed via ls using the -a option (meaning show all files).

Home Directories

User home directories contain files specific to users. A directory under /home is typically created for every ordinary user on the Linux system.

```
[cloud_user@ip-10-0-1-10 ~]$ pwd
/home/cloud_user
[cloud_user@ip-10-0-1-10 ~]$ cd
[cloud_user@ip-10-0-1-10 ~]$ pwd
/home/cloud_user
[cloud_user@ip-10-0-1-10 ~]$ cd ~
[cloud_user@ip-10-0-1-10 ~]$ pwd
/home/cloud_user
[cloud_user@ip-10-0-1-10 ~]$ cd $HOME
[cloud_user@ip-10-0-1-10 ~]$ pwd
/home/cloud_user
```

cd (change directory) without a specified path

~ used to represent the user's home directory

\$HOME environment variable used to store home directory path

To navigate to your home directory, use cd or cd ~

Absolute and Relative Paths

The path is the unique location of a file or directory.

The path to a file or directory location can be specified as an absolute path (starts with a /), or a relative path. Below is the pwd command show the absolute path of current working directory

```
[cloud_user@ip-10-0-1-10 ~]$ pwd
```

```
/home/cloud_user
```

Relative Path

Below, the cat command is used to show the content of file1 specified with a relative path.

```
[cloud_user@ip-10-0-1-10 ~]$ cat file1
```

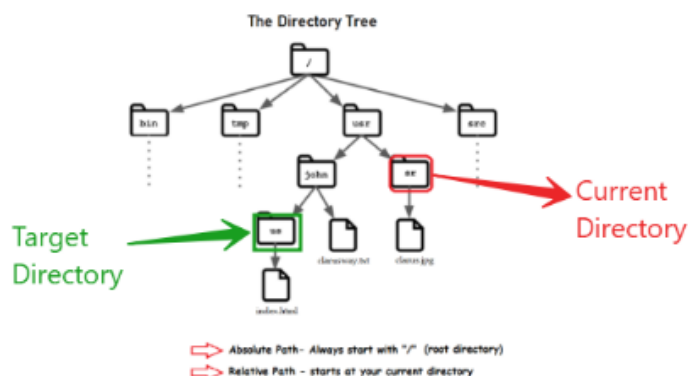
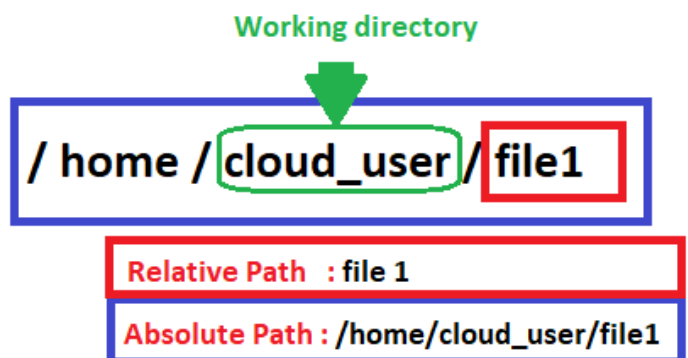
Good Day

Absolute Path

Below, the cat command is used to show the content of same file1 specified with a absolute path.

```
[cloud_user@ip-10-0-1-10 ~]$ cat /home/cloud_user/file1
```

Good Day



.(a single dot) _____ this represents the current directory.
..(two dot) _____ this represent the parent directory.

```
$ pwd
/usr/sr
$ cd /usr/john/us ← Absolute Path
$ pwd
/usr/john/us
```

```
$ pwd
/usr/sr
$ cd ../john/us ← Relative Path
$ pwd
/usr/john/us
```

Creating, Moving, Deleting

Creating, moving, and deleting files and directories.

touch - create a file

rm - delete the file

cp - used to copy file or folder

mv - used to move file or folder

mkdir - create a folder

rmdir - delete folder

Working with directories

Create a new directory mkdir <NAME>

Copy a directory cp -r <SOURCE> <DESTINATION>

Move a directory mv <SOURCE> <DESTINATION>

Delete a directory rm -r <DIRECTORY>

Delete an empty directory rmdir <DIRECTORY>

Working with files

Create a new file touch filename

Removing Files rm -option filename

Option -r includes the contents of a directory and the contents of all subdirectories when you remove a directory.

Option -i prevents the accidental removal of existing files or directories.

touch create a file

```
user@clarusway-linux: ~  
File Edit View Search Terminal Help  
user@clarusway-linux:~$ ls  
clarusway Downloads Pictures Videos  
classes.html examples.desktop Public  
Desktop lesson.txt Templates  
Documents Music test  
user@clarusway-linux:~$ touch test-file.txt  
user@clarusway-linux:~$ ls  
clarusway Downloads Pictures test-file.txt  
classes.html examples.desktop Public Videos  
Desktop lesson.txt Templates  
Documents Music test  
user@clarusway-linux:~$
```

rm delete a file

```
user@clarusway-linux: ~  
File Edit View Search Terminal Help  
user@clarusway-linux:~$ ls  
clarusway Downloads Pictures test-file.txt  
classes.html examples.desktop Public Videos  
Desktop lesson.txt Templates  
Documents Music test  
user@clarusway-linux:~$ rm test-file.txt  
user@clarusway-linux:~$ ls  
clarusway Downloads Pictures Videos  
classes.html examples.desktop Public  
Desktop lesson.txt Templates  
Documents Music test  
user@clarusway-linux:~$
```

cp copy a file to another location

```
user@clarusway-linux: ~/test  
File Edit View Search Terminal Help  
user@clarusway-linux:~$ cd test  
user@clarusway-linux:~/test$ ls  
user@clarusway-linux:~/test$ cd ..  
user@clarusway-linux:~$ cp test-file.txt test/  
user@clarusway-linux:~$ ls  
clarusway Downloads Pictures test-file.txt  
classes.html examples.desktop Public Videos  
Desktop lesson.txt Templates  
Documents Music test  
user@clarusway-linux:~$ cd test  
user@clarusway-linux:~/test$ ls  
test-file.txt  
user@clarusway-linux:~/test$
```

mv move a file to another location

```
user@clarusway-linux: ~/test  
File Edit View Search Terminal Help  
user@clarusway-linux:~/test$ ls  
user@clarusway-linux:~/test$ cd ..  
user@clarusway-linux:~$ ls  
clarusway Downloads Pictures test-file.txt  
classes.html examples.desktop Public Videos  
Desktop lesson.txt Templates  
Documents Music test  
user@clarusway-linux:~$ mv test-file.txt test/  
user@clarusway-linux:~$ ls  
clarusway Downloads Pictures Videos  
classes.html examples.desktop Public  
Desktop lesson.txt Templates  
Documents Music test  
user@clarusway-linux:~$ cd test
```

Case Sensitivity

Most of the common Linux file systems are case sensitive; this is something to keep in mind when creating and moving directories and files.

Using Case Sensitivity

Lower-case and upper-case letters have different ASCII representation.

Example;

```
touch newfile
```

```
touch Newfile
```

The commands above will create two different files.

Simple Globbing

Globbing is primarily used to match patterns in filenames or text by using a wildcard character to create the pattern.

? (Question mark) : Match any single character

* (Asterisk) : Match any number of character(s)

[] (Brackets) : Match character from a range

^ (Caret) : Used to match starting character

\$ (Dollar sign) : Used to match ending character

{ } (Curly brace) : Used to match more than one pattern

| Pipe : Used for applying more than one condition

Example :

```
clarusway@f85a0c1549f4:~$ ls
```

```
all file1 file2 file3 file4 happiness loneliness reverse
```

```
clarusway@f85a0c1549f4:~$ ls file?
```

```
file1 file2 file3 file4
```

```
clarusway@f85a0c1549f4:~$ ls ?????
```

```
file1 file2 file3 file4
```

```
clarusway@f85a0c1549f4:~$ ls ???1
```

```
file1
```



```
clarusway@f85a0c1549f4:~$ ls file*
file1 file2 file3 file4

clarusway@f85a0c1549f4:~$ ls *
all file1 file2 file3 file4 happiness loneliness reverse

clarusway@f85a0c1549f4:~$ ls *ness
happiness loneliness

clarusway@f85a0c1549f4:~$ ls *[1-4]
file1 file2 file3 file4

clarusway@f85a0c1549f4:~$ ls *[2-3]
file2 file3

clarusway@f85a0c1549f4:~$ ls *[:digit:]
file1 file2 file3 file4

[:upper:] or [[A-Z]] - match upper-case character
[:lower:] or [[a-z]] - match lower-case character
[:digit:] or [[0-9]] - match digits
[:alpha:] or [[a-zA-Z]] - match either case character
[:alphanum:] or [[a-zA-Z0-9]] - match alphanumeric
```

Manipulating Text Files with Text Editors

Why should I use an editor?

It is very important to be able to use at least one text mode editor. The first step toward independence is learning how to use an editor on your system.

We need to master an editor because we need it to edit files that affect our environment. You might want to start writing scripts, or books, creating websites or new programs as an advanced user. Mastering an editor will boost your productivity and your skills tremendously.

Which editor should I use?

Our focus for text editors is on the ones which can also be used in terminal windows and in systems without a graphical environment. The further advantage of mastering a text editor is its use on remote computers. Because you don't need to move the whole graphical environment over the network, working with text editors greatly enhances network speed.

Using the Vim editor

What's Vim:

Vim stands for “Vi Improved”. Vim is a highly configurable text editor for efficiently creating and changing any kind of text. It is included as "vi" with most UNIX systems and with Apple OS X.

Vim can operate in two modes: command mode and insert mode.

The editor always starts in command mode. When you run `vim filename` command to edit a file, Vim starts out in command mode. It means that, instead of adding those characters, all alphanumeric keys are bound to commands. For example, Typing `h` won't insert the character “h”, it will move the cursor to the left.

To enter the insert mode, type `i` for (“insert”) and now the keys will act the way you would expect. In the insert mode, you can type, move through the text, search, replace, mark blocks and perform other editing tasks. After pressing Escape, you are back in command mode.

Basic commands

Commands that switch the editor to insert mode

Vim Command	Description
<code>a</code>	append: it moves the cursor one position to the right before switching to insert mode
<code>i</code>	insert
<code>o</code>	insert a blank line under the current cursor position and move the cursor to that line.

Moving through the text

Moving through the text is usually possible with the arrow keys. If not, use:

Vim Command	Description
<code>h</code>	move the cursor one character to the left.
<code>j</code>	move the cursor down one line.
<code>k</code>	move the cursor up one line.
<code>l</code>	move the cursor one character to the right.
<code>0</code>	move the cursor to the beginning of the line.
<code>\$</code>	move the cursor to the end of the line.
<code>w</code>	move forward one word.
<code>b</code>	move backward one word.
<code>G</code>	move to the end of the file.
<code>gg</code>	move to the beginning of the file.

Basic operations

Vim Command	Description
-------------	-------------

<code>ndd</code>	delete <i>n</i> lines starting from the current cursor position.
------------------	--

<code>ndw</code>	delete <i>n</i> words at the right side of the cursor.
------------------	--

<code>x</code>	delete the character on which the cursor is positioned
----------------	--

<code>:n</code>	move to line <i>n</i> of the file.
-----------------	------------------------------------

<code>u</code>	undo the last operation.
----------------	--------------------------

<code>Ctrl-r</code>	redo the last undo.
---------------------	---------------------

<code>:w</code>	save (write) the file
-----------------	-----------------------

<code>:q</code>	exit the editor.
-----------------	------------------

<code>:q!</code>	force the exit when you want to quit a file containing unsaved changes.
------------------	---

<code>:wq</code>	save and exit
------------------	---------------

<code>:w newfile</code>	will save the text to newfile.
-------------------------	--------------------------------

<code>:wq!</code>	override read-only permission (if you have the permission to override permissions, for instance when you are using the root account.
-------------------	--

<code>:recover</code>	will recover a file after an unexpected interruption.
-----------------------	---

Searching And Replacing

Vim Command	Description
-------------	-------------

<code>/text</code>	search the text in the file and position the cursor on the first match below its position.
--------------------	--

<code>n</code>	move the cursor to the next instance of the text from the last search. This will wrap to the beginning of the document.
----------------	---

<code>N</code>	move the cursor to the previous instance of the text from the last search.
----------------	--

<code>?text</code>	search for text in the document, going backward.
--------------------	--

<code>:%s/text/replacement text/g</code>	search through the entire document for text and replace it with the replacement text.
--	---

<code>:%s/text/replacement text/gc</code>	search through the entire document and confirm before replacing text.
---	---

Copying And Pasting

Vim Command	Description
-------------	-------------

<code>v</code>	highlight one character at a time.
----------------	------------------------------------

<code>V</code>	highlight one line at a time.
----------------	-------------------------------

Ctrl-v highlight by columns.

y yank text into the copy buffer.

p paste text after the current line.

P paste text on the current line.

np paste it n times.

Using the nano editor

What's nano:

GNU nano is a small and friendly text editor. Besides basic text editing, nano offers features like undo/redo, syntax coloring, interactive search-and-replace, auto-indentation, line numbers, word completion, file locking, backup files, and internationalization support.

Opening and Creating Files

To open an existing file or to create a new file, type nano followed by the file name:

nano filename

nano screen



The default screen of nano consists of five areas. From top to bottom these are the title bar, a blank line, the edit window, the status bar, and two help lines.

The title bar consists of three sections: left, center and right. The section on the left displays the version of nano being used. The center section displays the current filename, or "New Buffer" if the file has not yet been named. The section on the right displays "Modified" if the file has been modified since it was last saved or opened.

The status bar is the third line from the bottom of the screen. It shows important and informational messages.

The two help lines at the bottom of the screen show some of the most essential functions of the editor. These two lines are called the Shortcut List.

Editing Files

Unlike vi, nano is a modeless editor, which means that you can start typing and editing the text immediately after opening the file.

Shortcut List

In both the toolbar and in the help text within the editor, key combos appear with standard Linux shortcut codes. All commands are prefixed with either ^ or M character. The caret symbol (^) represents the Ctrl key. For example, the ^G commands mean to press the Ctrl and G keys at the same time. The letter M represents the Alt key.

Command	Meaning	Description
Ctrl G	Get Help	display a complete list of commands and associated hotkeys.
Ctrl X Exit	Close the program.	You'll be prompted to save the current buffer or to exit without saving.
Ctrl O	Write Out	Save the current buffer to the filesystem.
Ctrl R	Read File	Open a file from the filesystem.
Ctrl W	Where Is	Search for a text string.
Ctrl \	Replace	Replace one text string for another.
Ctrl K	Cut Text	Remove the current line of text.
Ctrl U	Uncut Text	Uncut from the cutbuffer into the current line
Ctrl J	Justify	Justify the current paragraph
Ctrl T	To Spell	Spell-check the current buffer.
Ctrl C	Cur Pos	Display the position of the cursor relative to the contents of the buffer as a whole.
Alt U	Undo	Undo the last operation
Alt E	Redo	Redo the last undone operation
Alt M	Mark Text	Mark text starting from the cursor position
Alt 6	Copy Text	Copy the current line and store it in the cutbuffer

Interview Questions

1. What is Linux?

Linux is an operating system based on UNIX and was first introduced by Linus Torvalds. It is based on the Linux Kernel and can run on different hardware platforms manufactured by Intel, MIPS, HP,

IBM, SPARC, and Motorola. Another popular element in Linux is its mascot, a penguin figure named Tux.

2. What is the difference between UNIX and LINUX?

Unix originally began as a propriety operating system from Bell Laboratories, which later on spawned into different commercial versions. On the other hand, Linux is free, open source and intended as a non-propriety operating system for the masses.

3. What is BASH?

BASH is short for Bourne Again SHell. It was written by Steve Bourne as a replacement to the original Bourne Shell (represented by /bin/sh). It combines all the features from the original version of Bourne Shell, plus additional functions to make it easier and more convenient to use. It has since been adapted as the default shell for most systems running Linux.

4. What is Linux Kernel?

The Linux Kernel is a low-level systems software whose main role is to manage hardware resources for the user. It is also used to provide an interface for user-level interaction.

5. What is the advantage of open source?

Open source allows you to distribute your software, including source codes freely to anyone who is interested. People would then be able to add features and even debug and correct errors that are in the source code. They can even make it run better and then redistribute these enhanced source code freely again. This eventually benefits everyone in the community.

6. What are the basic components of Linux?

Just like any other typical operating system, Linux has all of these components: kernel, shells and GUIs, system utilities, and an application program. What makes Linux advantageous over other operating system is that every aspect comes with additional features and all codes for these are downloadable for free.

7. Describe the root account.

The root account is like a systems administrator account and allows you full control of the system. Here you can create and maintain user accounts, assigning different permissions for each account. It is the default account every time you install Linux.

8. What is CLI?

CLI is short for Command Line Interface. This interface allows the user to type declarative commands to instruct the computer to perform operations. CLI offers greater flexibility. However, other users who are already accustomed to using GUI find it difficult to remember commands including attributes that come with it.

9. What is GUI?

GUI, or Graphical User Interface, make use of images and icons that users click and manipulate as a way of communicating with the computer. Instead of having to remember and type commands, the use

of graphical elements makes it easier to interact with the system, as well as adding more attraction through images, icons, and colors.

10. How do you change permissions under Linux?

Assuming you are the system administrator or the owner of a file or directory, you can grant permission using the `chmod` command. Use `+` symbol to add permission or `-` symbol to deny permission, along with any of the following letters: `u` (user), `g` (group), `o` (others), `a` (all), `r` (read), `w` (write) and `x` (execute). For example, the command `chmod go+rw FILE1.TXT` grants read and write access to the file `FILE1.TXT`, which is assigned to groups and others.

11. What are filenames that are preceded by a dot?

In general, filenames that are preceded by a dot are hidden files. These files can be configuration files that hold important data or setup info. Setting these files as hidden makes it less likely to be accidentally deleted.

12. What is the `pwd` command?

The `pwd` command is short for print working directory command.

13. What are the kinds of permissions under Linux?

There are 3 kinds of permissions under Linux:- Read: users may read the files or list the directory- Write: users may write to the file or new files to the directory- Execute: users may run the file or lookup a specific file within a directory

14. What are the different modes when using `vi` editor?

There are 3 modes under `vi`:- Command mode – this is the mode where you start in- Edit mode – this is the mode that allows you to do text editing- Ex mode – this is the mode wherein you interact with `vi` with instructions to process a file

15. Why we use LINUX?

LINUX is used widely because it is completely different from other operating systems where every aspect comes with something extra i.e. some additional features. Some of the major reasons to use LINUX are listed below

It is an open-source operating system where programmers get the advantage of designing their own custom OS

Software and the server licensing required to install Linux is completely free and can be installed on many computers as required

It has low or minimum but controllable issues with viruses, malware, etc

It is highly secured and supports multiple file systems

16. Enlist some Linux distributors (Distros) along with its usage?

Different parts of LINUX say kernel, system environment, graphical programs, etc are developed by different organizations. LINUX Distributions (Distros) assemble all these different parts of Linux and give us a compiled operating system to be installed and used.

There are around six hundred Linux distributors. Let us see some of the important ones

UBuntu: It is a well known Linux Distribution with a lot of pre-installed apps and easy to use repositories libraries. It is very easy to use and works like a MAC operating system.

Linux Mint: It uses cinnamon and mates desktop. It works on Windows and should be used by newcomers.

Debian: It is the most stable, quicker and user-friendly Linux Distributors.

Fedora: It is less stable but provides the latest version of the software. It has a GNOME3 desktop environment by default.

Red Hat Enterprise: It is to be used commercially and to be well tested before release. It usually provides a stable platform for a long time.

Arch Linux: Every package is to be installed by you and is not suitable for beginners.

17. Explain the Linux Directory commands along with the description?

Enlisted below are the directory commands along with descriptions

pwd: It is a built-in command which stands for 'print working directory'. It displays the current working location, working path starting with / and directory of the user. Basically, it displays the full path to the directory you are currently in.

Is: This command list out all the files in the directed folder.

cd: This stands for 'change directory'. This command is used to change to the directory you want to work from the present directory. We just need to type cd followed by the directory name to access that particular directory.

mkdir: This command is used to create an entirely new directory.

rmdir: This command is used to remove a directory from the system.

18. Explain Linux Shell?

For executing any commands user uses a program known as the shell. Linux shell is basically a user interface used for executing the commands and communicating with Linux operating system. Shell does not use the kernel to execute certain programs, create files, etc. There are several shells available with Linux which includes the following

BASH (Bourne Again SHell)

CSH (C Shell)

KSH (Korn Shell)

TCSH

19. What is a Shell script?

As the name suggests, the shell script is the script written for the shell. This is a program file or says a flat text file where certain Linux commands are executed one after another. Although the execution speed is slow, Shell script is easy to debug and can also simplify everyday automation processes.

20. Explain the Linux 'cd' command options along with the description?

'cd' stands for change directory and is used to change the current directory on which the user is working.

cd syntax : `$ cd {directory}`

Following purposes can be served with 'cd' commands

Change from current to a new directory

Change directory using the absolute path

Change directory using the relative path

Few of the 'cd' options are enlisted below

`cd~`: Brings you to the home directory

`cd-`: Brings you to the previous directory

`..`: Bring you to the parent directory

`cd/`: Takes you to the entire system's root directory

21. What are the basic components of Linux?

Just like other operating systems, Linux has all components like kernel, shells, GUIs, system utilities and application programs.

22. Define Shell

It is an interpreter in Linux.

23. Name the Linux loader.

LILO is the Linux loader.

24. If you have saved a file in Linux. Later you wish to rename that file, what command is designed for it?

The 'mv' command is used to rename a file.

25. What is the maximum length for a filename in Linux?

255 characters.

26. How to delete information from a file in vi?

The following commands are used to delete information from vi editors.

x deletes a current character.

dd deletes the current line.