

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
<div><div></div>Linux Introduction</div>

Linux Introduction

Contents

- [Overview](#)
 - [Anatomy of the OS](#)
 - [Linux](#)
 - [History of Linux](#)
 - [Linux & Windows](#)
 - [Uses of Linux](#)
 - [Linux Distributions](#)
 - [Linux Software Packages](#)
- [Tutorial](#)
- [Exercises](#)

Overview

Linux is an Operating System (OS). You can think of OS as an application that takes control of the hardware.

Anatomy of the OS

The Operating System is comprised of a number of different components:

- Bootloader
- Kernel
- Daemon(s)
- Shell
- Graphical (X) Server
- Desktop Environment
- Applications

Since the OS manages all of the communication between the software and hardware. Without it, nothing can run.

Bootloader

This is the program that starts when the device is turned on. It boots up the OS. Typically user doesn't need to interact with the Bootloader.

If a user has more than one OS on their device, the user chooses on start up which Operating System they want to use and then works with that one.

Kernel

This is the core of the system. The kernel manages all resources, the CPU, memory etc. It has complete control over the system.

Users can use programs that interact with the kernel.

Daemons

These are background processes that are ran for the OS. A process is a program that is running. These processes are typically started by the Operating System but can be started by the user.

The Daemon processes have many uses. Including logging, managing connections and scheduling.

<div><div></div><div>Linux Distributions</div></div> <div><div></div><div>Bash Interpreter</div></div> <div><div></div><div>Sessions in Linux</div></div> <div><div></div><div>Lists and Directories</div></div> <div><div></div><div>Editing Text</div></div> <div><div></div><div>Aliases, Functions and Variables</div></div> <div><div></div><div>User Administration</div></div> <div><div></div><div>Ownership</div></div> <div><div></div><div>Data Streams</div></div> <div><div></div><div>Pipes and Filters</div></div> <div><div></div><div>Scripting in Linux</div></div> <div><div></div><div>Sudoers</div></div> <div><div></div><div>Managing systemd Services</div></div> <div><div></div><div>Systemd Service Configuration</div></div> <div><div></div><div>OpenSSH</div></div> <div><div></div><div>Screens in Linux</div></div>
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

The first process started on a Linux system is called **system**.
It was previously known as **init**.
All other processes are derivatives of this process.
This is started by the kernel. If it doesn't, a kernel panic will occur.

Kernel Panic - A safety measure taken by an **OS's** kernel upon detecting an internal fatal error in which it either is unable to safely recover or can not have the system continue to run without having a much higher risk of major data loss.

The Shell

A user normally interacts with the system through the **shell**.
This is the process that allows users to type in commands to run.
It is completely **text based**.

The shell will be used quite extensively throughout these modules.
There will be times where you need to work with a Linux machine and you won't have access to a **Graphical User Interface**.
In this case you will have a shell to work with.

Graphical Server

Typically referred to as the **X Window, X11** or **X**.
It provides the **Graphical User Interface (GUI)** for a Linux system.
Window display, mouse movement, etc. are all handled by the **Graphical Server**.

The Graphical Server is just the baseline for a **GUI**.
Other programs will adjust the looks of the GUI.
GNOME is a popular example of one of these programs.
The Graphical Server simply hosts the desktop environment.

Desktop Environment

This is the part of the system that user will generally interact with. It is made up of programs running on top of an **OS**.

A desktop environment is normally made up of:

- Windows and toolbars
- Folders
- Desktop Icons and Widgets
- Applications

Typically server **OS's** do not have a desktop environment and you will only have a shell to interact with.

Linux

Linux is an **open source** kernel.
Most people have used Windows on their computer, but Linux is extremely popular and is seen in a number of systems.

Linux has a massive **community** with lots of distributions being open source (free) and there is a lot of support around the **OS** and how it works.

Due to its high efficiency, open source development, and modular nature, Linux is predominantly used as the Operating System of choice for Infrastructure Hardware. From Routers to Web Servers, Microservices to Monoliths, the chances that they are built on Linux is very high.

History of Linux

Linux was developed by **Linus Torvalds** in the 90's.
He developed a kernel that was to be used alongside Minix, a non-free system similar to Unix.
It was released in 1991, and eventually combined with GNU to produce a free

operating system.

Linus is still actively contributing to the development of Linux and any change has to be approved by him, before it ends up in the next version of kernel.

The initial release prompted a number of volunteers to work with **Torvalds** to improve the kernel.

Linux started being used in larger scale systems for networking, database hosting and web hosting.

Unix - An operating system written in C. Linux is a clone of Unix

GNU - Another free operating system as well as a collection of software. Can work alongside Linux.

Linux & Windows

Linux and Windows have many differences.

- Linux is free of charge and community driven.
- The kernel is customisable
- It's open source meaning the community is there for support
- It has been developed into many **Distros** or **Flavours** with various priorities

Uses of Linux

The Linux Kernel is very versatile. It can be installed on anything that has storage space and a CPU, with some modifications.

Examples of Linux in strange places:

- The Linux motorcycle - Mavizen's TTX02 / 2013 Cadillac XTS
- The Linux Coffee Pot / Smart Slow Cooker
- US Navy Sonar Systems
- The Large Hadron Collider
- The International Space Station
- Air Traffic Control Systems

Linux Distributions

There are a variety of different versions of Linux, known as **distributions** or **flavours**.

All these distributions are built on the Linux kernel, but can have minor or major differences.

Some distributions are tailored for desktop use, whilst others are more commonly used for servers.

- *Red Hat/CentOS*: the focus here is on stability and security for enterprise systems (utilising the SELinux Kernel Module to enhance security), most commonly used for servers.
- *Debian*: the focus here is on a steady and stable release cycle where the development community validate each package before they release it to the public. As a result the official packages can lag behind as dependability is prioritised over the bleeding edge.
- *Ubuntu*: the focus here is on Usability as a Desktop environment, meant as the first Linux Distro most people will have experience of, it typically comes pre-installed on many netbooks and can even run from a Live medium such as a DVD or USB thumb drive.

Linux Software Packages

Linux normally works with packages, which are bundles of software.

There are a number of packages available on **Linux**, but you can install your own packages too.

Linux uses a **package manager** as a tool to do all the hard work of installing, updating and uninstalling the packages.

All we have to do is write the command, for example installing Java JDK would be a command like **apt-get install default-jdk**

When you use the `apt`, `yum`, `pacman` or any other package manager, software is downloaded from a central repository. Apart from central repositories there are private ones, and they can be added to the list of repositories for the package manager to look for software.

As this is Linux, there is always the option to handle all the heavy lifting manually without utilising a package manager, but this will only be useful in very rare circumstances where software needs to be customised and building from source is the only option.

The distribution of focus throughout these modules will be `Ubuntu`. It is designed to be simplistic enough for beginners, but not limited enough to hinder the potential of a more advanced user.

Tutorial

There is no tutorial for this module.

Exercises

Now conduct some of your own research about what are the more popular distributions of Linux.

► Examples