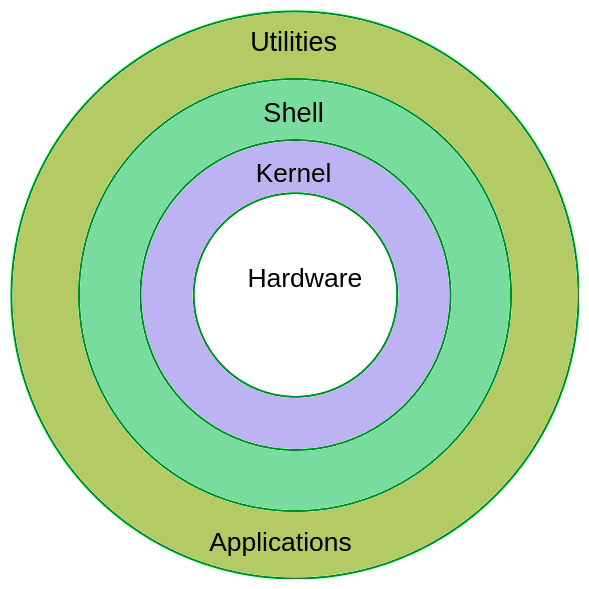
**Linux**

**What is Linux?**

Linux is a free and open-source operating system based on the Unix operating system. It was created by Linus Torvalds in 1991 and has since become one of the most popular operating systems used in servers, supercomputers, mobile devices, and other embedded systems.

Linux is known for its stability, security, and versatility. It is highly customisable and can be tailored to suit many use cases, from small embedded devices to large-scale server deployments. It supports various programming languages and software tools, making it popular among developers and system administrators.

**Why Linux?**

* Open-source
* Stable and secure
* Highly customisable
* Supports a wide range of programming languages and tools
* Compatible with a wide range of devices and architectures
* Cost-effective

**Linux Flavours**

* RHEL : (Red Hat Enterprise Linux)
* CentOS
* Ubuntu
* Amazon Linux
* Fedora
* Linux Mint
* OpenSUSE

**Linux Architecture**

**Utilities:** Refers to various software tools and applications that provide additional functionality to the operating system, such as text editors, file managers, and network tools.

**Shell:** A command-line interface that allows users to interact with the operating system by entering commands and scripts.

**Kernel:** The core of the Linux operating system that provides basic services and device drivers, and communicates with user space through system calls.

**Hardware:** Refers to the physical components of a computer system, such as the CPU, memory, storage, and input/output devices, that are used by the operating system to run applications and services.

**Applications:** User-level programs and services that run on top of the kernel and interact with it through system calls, providing various functionalities to the user, such as web browsers, media players, and development tools.

**File system hierarchy**

* / - This is top level directory
* /root - It is home directory for root user
* /home - It is home directory for other users
* /boot - It contains bootable files for Linux
* /etc - It contains all configuration files
* /usr - by default softwares are installed in this directory
* /bin - It contains commands used by all users
* /sbin - It contains commands used by only root user : (root)

**Linux basic Commands**

* cat : (create & append file)
* touch : (create blank file)
* nano : (create & edit file)
* vi/vim : (create & edit file)
* ls : (list) : (-a, -la)
* cd : (change directory)
* pwd : (print working directory)
* mkdir : (create directory, multiple)
* cp : (copy)
* mv : (move)
* mv : (rename)
* rm : (remove file)
* tree : (see in tree structure)
* rm -rf: (remove directory & recursive)
* grep : (pick & print)
* less : (see output)
* head : (see top 10 lines)
* tail : (see last 10 lines)
* sort : (display in Alphabetic/Numeric order)
* User
* cat /etc/passwd | cut -d: -f1 (user list)
* Group
* Soft Link : (shortcut)
* Hard Link : (backup)
* /tar : (to pack)
* gz : (to compress)
* yum : (to install)
* wget : (to download)
* File/Directory Permissions:
* chmod : (permissions)
* chown : (owner)
* chgrp : (group)
* hostname : (to see hostname)
* ifconfig : (to get ip address)
* cat /etc/\*rele\* : (to get os version)
* apt get install httpd : (to install package)
* yum update httpd : (to upgrade package)
* yum remove httpd : (to uninstall package)
* yum list installed : (to see installed packages)
* service httpd status : (to see status)
* service httpd start : (to start service)
* service httpd reload : (to restart service)
* service httpd restart : (to restart service)
* chkconfig httpd on : (to start service permanently)
* chkconfig httpd off : (to stop service permanently)
* Redirection : (redirecting output)
* which : (to see package installed or not)
* sudo : (to get root privileges)
* whoami : (to see user)
* find -type f : (to see all files in current directory)
* find -type d : (to see all directories in current directory)
* find / -type f : (to see all files under top level root directory)
* find / -type d : (to see all directories under top level root directory)
* find / -type f -name <file\_name> : (to search specific file under top level root directory)
* find / -type d -name <dir\_name> : (to search specific dir under top level root directory)

**Advanced**

* find: Search for files and directories based on different criteria such as name, size, or modification time.
* awk: A powerful text processing tool that can be used for filtering, formatting and manipulating text data.sed: A stream editor for filtering and transforming text data.
* tar: An archiving utility that can be used to create and extract archives of multiple files and directories.
* gzip: A compression tool that can be used to compress and decompress files and directories.
* dd: A utility for low-level copying and converting data, used for tasks such as creating bootable USB drives or cloning disks.
* iptables: A powerful firewall tool that can be used to set up and manage firewall rules.
* rsync: A tool for synchronizing files and directories between local and remote systems.
* nc/netcat: A utility for network communication, can be used for port scanning, file transfer, and other network-related tasks.
* lsof: A command for listing all open files and sockets on the system.
* du: A command for displaying disk usage of files and directories.
* df: A command for displaying disk free space on file systems.
* screen: A terminal multiplexer that allows multiple terminal sessions to be run within a single terminal window.
* systemctl: A tool for managing system services on modern Linux systems.
* journalctl: A tool for querying and analyzing system logs.

**Shell scripting**

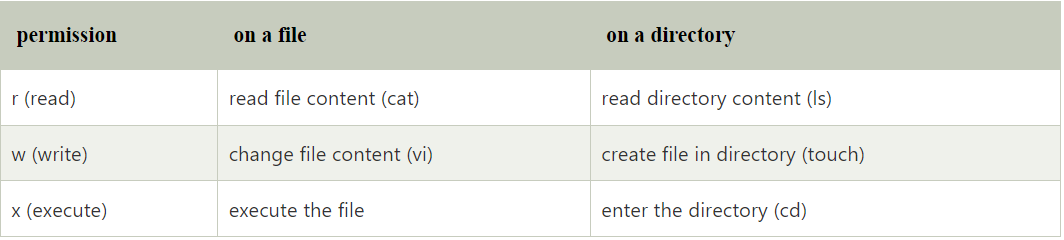
Shell Script is a Simple plain text file that contains a series of commands

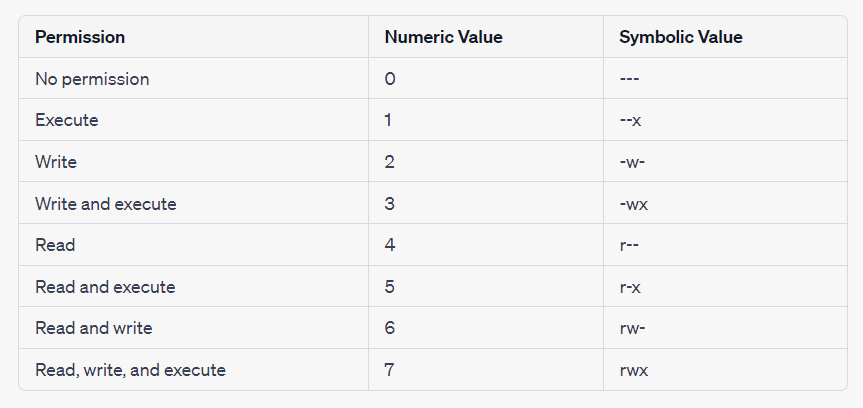
Shell scripting is the use of commands written in a shell language to automate tasks on a Linux or Unix-based operating system. This allows users to execute a series of commands without having to manually enter them each time.<https://github.com/GudditiOrg/Shellscript> Check for sample code that automates the webapp installation and initialize the service .

| root@ubuntu:/home/gudditi# vi first.sh #!/bin/bash echo "My first shell script"  root@ubuntu:/home/gudditi# sh first.sh "My first shell script" root@ubuntu:/home/gudditi# |
| --- |

**File Permissions:**

* Owner: The user who created the file
* Group: A group of users who have been assigned certain permissions by the owner
* Others: Any user who is not the owner or in the assigned group

****



| root@ubuntu:/home/gudditi# chmod 444 first.sh root@ubuntu:/home/gudditi# ls -ltr -r--r--r-- 1 root root 46 Apr 24 14:07 first.sh  root@ubuntu:/home/gudditi# chmod 755 first.sh root@ubuntu:/home/gudditi# ls -ltr -rwxr-xr-x 1 root root 46 Apr 24 14:07 first.sh |
| --- |

**Conditions:**

conditions can be used to control the flow of the script based on the evaluation of certain expressions or values. Here are some examples of conditions in shell scripts:

**If statement:**

The if statement is used to execute a block of code if a condition is true. Here's an example:

| #!/bin/bash echo "My first shell script"  read age  if [ $age -gt 22 ]; then  echo "you are major "  fi |
| --- |

In this example, if the value of variable x is greater than 5, the code inside the if block will be executed, which in this case is to print a message.

If-else statement:

The if-else statement is used to execute a block of code if a condition is true, and another block of code if the condition is false. Here's an example:

| #!/bin/bash echo "My first shell script"  read age  if [ $age -gt 22 ]; then  echo "you are major "  else  echo "you are not major" fi |
| --- |

If-else-if statement:

if else if is a shorthand for writing multiple if statements in a more concise manner. It is also called else if ladder or nested if.

The if else if statement allows you to test multiple conditions in a hierarchical order. It starts with an if statement, followed by any number of else if statements, and ends with an optional else statement.

| #!/bin/bash echo "My first shell script" echo "Please enter your age: " read age  if [ $age -ge 22 ] && [ $age -lt 60 ]; then  echo "You are not in pension age range" elif [ $age -ge 60 ]; then  echo "You are ready for pension" else  echo "Go and study well" fi |
| --- |

**Lopps:**

**For Loops:**

For loops allow you to repeat a set of commands a specific number of times, based on the values in a sequence or range. Here's an example of a for loop that prints the numbers 1 through 5:

Example : 1

| #!/bin/bash  echo "for Loop example"  for i in 1 2 3 4 5 6 do   echo $i done   Output for Loop example 1 2 3 4 5  6 |
| --- |

Example: 2

| #!/bin/bash  echo "for Loop example"  for i in {1..6} do   echo $i done   Output for Loop example 1 2 3 4 5 6 |
| --- |

**While Loop**

a while loop is a construct that allows you to repeatedly execute a set of commands as long as a certain condition is true.

| #!/bin/bash  count=1  while [ $count -le 5 ] do  echo $count  count=$((count+1)) done  echo "Done!" |
| --- |

### Sample examples 🙂

| gudditi@ubuntu:~$ cat sample.sh #!/bin/bash  path=$(ls -ltr /home/gudditi) echo " $path /n"  gudditi@ubuntu:~$ bash sample.sh  total 16 drwxr-xr-x 6 root root 4096 Apr 24 10:15 Meddna-APP -rwxr-xr-x 1 root root 293 Apr 28 11:31 install.sh -rwxr-xr-x 1 gudditi gudditi 108 May 1 09:51 first.sh -rwxr-xr-x 1 gudditi gudditi 62 May 1 09:56 sample.sh /n gudditi@ubuntu:~$ |
| --- |

**Some adwanced commands:**

cut: Extract columns from a file based on a delimiter.

cut -d ":" -f 1,3 file.txt

paste: Merge lines of files together.

paste file1.txt file2.txt

tr: Translate or delete characters in a file.

tr '[:lower:]' '[:upper:]' < file.txt

awk: Perform text processing operations on a file, such as filtering and pattern matching.

awk '/pattern/ {print $1}' file.txt

sed: Stream editor for filtering and transforming text.

sed 's/pattern/replacement/g' file.txt

grep: Search for a pattern in a file and print matching lines.

grep "pattern" file.txt

sort: Sort lines of text files.

sort file.txt

uniq: Find and remove duplicate lines from a file.

uniq file.txt

wc: Count lines, words, and characters in a file.

wc file.txt

tee: Read from standard input and write to standard output and files simultaneously.

cat file.txt | tee new\_file.txt

CronJobs:

Contab -e : new cron job

Crontab -l : list all jobs

Corntab -r : remove job

https://github.com/GudditiOrg/Shellscript