Ubuntu Guide

What is Ubuntu?

Ubuntu is a free and open-source Linux-based operating system developed by Canonical. It is one of the most popular Linux distributions, widely used for personal, professional, and server environments. Ubuntu is known for its ease of use, security, and large community support.

Why Use Ubuntu?

1. **Open Source & Free** - Ubuntu is completely free and open source, making it an attractive option for individuals and businesses.
2. **User-Friendly Interface** - Ubuntu offers a clean and intuitive interface, making it accessible even for beginners.
3. **Security & Stability** - With regular security updates and a strong development community, Ubuntu is one of the most secure operating systems.
4. **Software Availability** - It supports a vast repository of open-source applications and tools.
5. **Customization & Flexibility** - Users can customize Ubuntu as per their requirements, from UI themes to system configurations.
6. **Community Support** - A large community and extensive documentation make troubleshooting easier.
7. **Compatibility with Cloud & Servers** - Ubuntu is widely used in cloud computing and server management due to its robustness.
8. **Performance & Efficiency** - Ubuntu is optimized for performance and is widely used in lightweight computing environments.
9. **Developer-Friendly** - Ubuntu supports a range of development tools, making it a preferred OS for developers and engineers.

Different Versions of Ubuntu

Ubuntu has different versions catering to various needs:

1. **Ubuntu Desktop** - The standard version for personal use, featuring a GUI and pre-installed applications.
2. **Ubuntu Server** - Designed for server environments, offering optimized performance and security features.
3. **Ubuntu Core** - A minimal, containerized version of Ubuntu for IoT and embedded devices.
4. **Ubuntu Studio** - A version tailored for multimedia production, including audio, video, and graphics tools.
5. **Kubuntu, Xubuntu, Lubuntu** - Lightweight variants with different desktop environments for varied user experiences.

Basic Linux Commands

1. `pwd` – Print current working directory

```
pwd
```

Example Output:

```
/home/user
```

2. `ls` – List files in a directory

```
ls
```

Example Output:

```
documents downloads pictures music
```

3. `ls -l` – Detailed list of files

```
ls -l
```

Example Output:

```
drwxr-xr-x 2 user user 4096 Mar 27 10:00 Documents
-rw-r--r-- 1 user user  123 Mar 27 10:05 file.txt
```

4. `ls -a` – Show hidden files

```
ls -a
```

Example Output:

```
.  ..  .bashrc  .profile  Documents  Downloads
```

5. `cd <directory>` – Change directory

```
cd Documents
```

6. `cd ..` – Move up one level

```
cd ..
```

7. `mkdir <dir>` – Create a new directory

```
mkdir my_folder
```

8. `rmdir <dir>` – Remove an empty directory

```
rmdir my_folder
```

9. `rm <file>` – Delete a file

```
rm file.txt
```

10. `rm -r <dir>` – Remove a directory and its contents

```
rm -r my_folder
```

11. `cp <source> <destination>` – Copy files

```
cp file.txt /home/user/Documents/
```

12. `cp -r <source> <destination>` – Copy directories

```
cp -r folder1 folder2
```

13. `mv <source> <destination>` – Move or rename files

```
mv oldname.txt newname.txt
```

14. `touch <file>` – Create an empty file

```
touch newfile.txt
```

15. `cat <file>` – Show file contents

```
cat file.txt
```

16. `less <file>` – View file page-wise

```
less largefile.txt
```

17. `more <file>` – View file with pagination

```
more file.txt
```

18. `head <file>` – View first 10 lines

```
head file.txt
```

19. `tail <file>` – View last 10 lines

```
tail file.txt
```

20. `tail -f <file>` – View file changes in real-time

```
tail -f logfile.log
```

21. `echo "Hello World"` – Print text

```
echo "Hello World"
```

22. `clear` – Clear the terminal

```
clear
```

23. `history` – Show command history

```
history
```

24. `uptime` – Show system uptime

```
uptime
```

25. `whoami` – Show the current user

```
whoami
```

26. `who` – Show logged-in users

```
who
```

27. `w` – Show detailed user activity

```
w
```

28. `date` – Show system date

```
date
```

29. `cal` – Display a calendar

```
cal
```

30. `df -h` – Show disk space usage

```
df -h
```

31. `du -sh <dir>` – Show directory size

```
du -sh Documents
```

32. `lsblk` – List block devices

```
lsblk
```

33. `blkid` – Display UUID of partitions

```
blkid
```

34. `mount /dev/sdb1 /mnt` – Mount a drive

```
mount /dev/sdb1 /mnt
```

35. `umount /mnt` – Unmount a drive

```
umount /mnt
```

36. `free -h` – Show memory usage

```
free -h
```

37. `uname -a` – Show system information

```
uname -a
```

38. `hostname` – Display system hostname

```
hostname
```

39. `uptime` – Show system uptime

```
uptime
```

40. `env` – Show environment variables

```
env
```

41. `export VAR=value` – Set an environment variable

```
export MY_VAR=hello
```

42. `alias ll='ls -lah'` – Create a command alias

```
alias ll='ls -lah'
```

43. `unalias ll` – Remove an alias

```
unalias ll
```

44. `passwd` – Change password

```
passwd
```

45. `exit` – Logout from terminal

```
exit
```

46. `shutdown -h now` – Shutdown the system

```
shutdown -h now
```

47. `reboot` – Restart the system

```
reboot
```

48. `sleep 5` – Wait for 5 seconds

```
sleep 5
```

49. `time <command>` – Measure execution time of a command

```
time ls
```

50. `man <command>` – Show the manual for a command

```
man ls
```

File Permissions and Ownership in Linux

Linux provides a powerful permission system to control access to files and directories. Below are essential commands related to file permissions and ownership, along with examples.

1. Changing File Permissions (`chmod`)

1. `chmod 777 <file>` – Full permissions

```
chmod 777 myfile.txt
```

Gives read, write, and execute permissions to everyone.

2. `chmod 755 <file>` – Read & execute for all, write for owner

```
chmod 755 script.sh
```

Owner has full permissions, others can only read and execute.

3. `chmod u+x <file>` – Add execute permission to the user

```
chmod u+x run.sh
```

Makes the file executable for the owner.

2. Changing Ownership (`chown` and `chgrp`)

4. `chown user:group <file>` – Change file ownership

```
chown alice:developers report.txt
```

Changes owner to Alice and group to developers.

5. `chgrp group <file>` – Change group ownership

```
chgrp staff document.docx
```

Assigns the file to the 'staff' group.

3. Viewing and Finding File Permissions

6. `ls -l` – View file permissions

```
ls -l myfile.txt
```

Displays detailed file information, including permissions.

7. `umask 022` – Default permission setting

```
umask 022
```

Sets default permissions for new files and directories.

8. `find /path -type f -perm 777` – Find files with 777 permission

```
find /home/user -type f -perm 777
```

Searches for files with full access in the specified directory.

4. Modifying Permissions

9. `chmod +r <file>` – Add read permission

```
chmod +r notes.txt
```

Makes the file readable for everyone.

10. `chmod +w <file>` – Add write permission

```
chmod +w log.txt
```

Allows write access to all users.

11. `chmod +x <file>` – Add execute permission

```
chmod +x script.sh
```

Makes the script executable.

12. `chmod -r <file>` – Remove read permission

```
chmod -r private.txt
```

Prevents users from reading the file.

13. `chmod -w <file>` – Remove write permission

```
chmod -w report.docx
```

Prevents modifications to the file.

14. `chmod -x <file>` – Remove execute permission

```
chmod -x program.bin
```

Prevents execution of the file.

5. Managing File Attributes (`lsattr` and `chattr`)

15. `lsattr` – List file attributes

```
lsattr myfile.txt
```

Displays special file attributes.

16. `chattr +i <file>` – Make a file immutable

```
chattr +i config.cfg
```

Prevents the file from being modified or deleted.

17. `chattr -i <file>` – Remove immutability

```
chattr -i config.cfg
```

Allows modifications to the file again.

18. `chattr +a <file>` – Append only

```
chattr +a log.txt
```

Only allows appending to the file.

19. `chattr -a <file>` – Remove append-only

```
chattr -a log.txt
```

Allows full modifications to the file.

6. Getting File Information

20. `stat <file>` – Get detailed file information

```
stat report.pdf
```

Displays metadata about the file.

21. `getfacl <file>` – Get Access Control List (ACL)

```
getfacl shared.txt
```

Shows ACL permissions for the file.

7. Modifying Access Control Lists (ACL)

22. `setfacl -m u:user:rwx <file>` – Set ACL for a user

```
setfacl -m u:john:rwx project.doc
```

Gives John full permissions on the file.

23. `setfacl -x u:user <file>` – Remove ACL for a user

```
setfacl -x u:john project.doc
```

Removes John's special access.

24. `setfacl -b <file>` – Remove all ACL entries

```
setfacl -b shared.txt
```

Clears all ACL settings on the file.

8. Viewing Directory Permissions

25. `ls -ld <directory>` – View directory permissions

```
ls -ld /var/www
```

Displays permission settings for the directory.

User Management Commands in Linux

Overview

User management is an essential aspect of Linux system administration. Below is a list of common user management commands along with examples to help you manage users efficiently.

Commands and Examples

1. `whoami` - Show current user

```
whoami
```

Output:

```
user123
```

2. `id` - Display user ID and group ID

```
id
```

Output:

```
uid=1000(user123) gid=1000(user123) groups=1000(user123),27(sudo)
```

3. `who` - Show all logged-in users

```
who
```

Output:

```
 user1  tty1  2025-03-27 09:00
user2  pts/0  2025-03-27 09:15
```

4. `w` - Show user activity

```
w
```

Output:

```
 10:00:01 up 1:00, 2 users, load average: 0.12, 0.15, 0.10
USER     TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
user1    tty1     :0             09:00    1:00m   0.02s  0.01s bash
```

5. `adduser <username>` - Create a new user

```
sudo adduser newuser
```

6. `passwd <username>` - Set a user's password

```
sudo passwd newuser
```

7. `deluser <username>` - Delete a user

```
sudo deluser newuser
```

8. `usermod -aG <group> <username>` - Add a user to a group

```
sudo usermod -aG sudo newuser
```

9. `groups <username>` - Show user groups

```
groups newuser
```

10. `groupadd <groupname>` - Create a new group

```
sudo groupadd developers
```

11. `groupdel <groupname>` - Delete a group

```
sudo groupdel developers
```

12. `chage -l <username>` - Show password expiry info

```
sudo chage -l newuser
```

13. `chage -M 30 <username>` - Set password expiry

```
sudo chage -M 30 newuser
```

14. `su <username>` - Switch user

```
su - newuser
```

15. `sudo su` - Switch to root user

```
sudo su
```

16. `sudo -i` - Open an interactive root shell

```
sudo -i
```

17. `who -b` - Show last system reboot

```
who -b
```

Output:

```
boot   2025-03-27 08:30
```

18. `finger <username>` - Display user information

```
finger newuser
```

19. `last` - Show last logins

```
last
```

20. `lastlog` - Show last login for all users

```
lastlog
```

21. `pkill -u <username>` - Kill all processes of a user

```
sudo pkill -u newuser
```

22. `w -s` - Show short version of active users

```
w -s
```

23. `who -r` - Show current runlevel

```
who -r
```

24. `who -q` - Show total logged-in users

```
who -q
```

25. `sudo <command>` - Execute commands as root

```
sudo apt update
```

Networking Commands in Ubuntu

This document provides a list of essential networking commands in Ubuntu along with their descriptions and examples.

1. `ping` - Check Connectivity

Used to check if a remote host is reachable.

```
ping google.com
```

Example Output:

```
 PING google.com (142.250.180.78) 56(84) bytes of data.
64 bytes from 142.250.180.78: icmp_seq=1 ttl=118 time=10.2 ms
```

2. `traceroute` - Trace Route to a Host

Displays the path packets take to a host.

```
traceroute google.com
```

Example Output:

```
 traceroute to google.com (142.250.180.78), 30 hops max, 60 byte packets
1 192.168.1.1 (192.168.1.1) 2.502 ms 2.301 ms 2.113 ms
2 10.10.10.1 (10.10.10.1) 10.512 ms 10.222 ms 10.435 ms
...
```

3. `nslookup` - DNS Lookup

Used to query DNS records for a domain.

```
nslookup google.com
```

Example Output:

```
 Server:  8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:
Name: google.com
Address: 142.250.180.78
```

4. `dig` - Get DNS Information

Fetch detailed DNS information for a domain.

```
dig google.com
```

Example Output:

```
 ;; ANSWER SECTION:
google.com.   299 IN A 142.250.180.78
```

5. `host` - Find IP of a Domain

Find the IP address associated with a domain.

```
host google.com
```

Example Output:

```
google.com has address 142.250.180.78
```

6. `wget` - Download a File

Download a file from a URL.

```
wget https://example.com/file.zip
```

Example Output:

```
Saving to: ‘file.zip’
```

7. `curl` - Fetch HTTP Headers

Retrieve HTTP headers from a website.

```
curl -I https://example.com
```

Example Output:

```
HTTP/1.1 200 OK
Date: Mon, 25 Mar 2025 12:00:00 GMT
```

8. `scp` - Secure Copy

Copy files securely between hosts.

```
scp user@remote:/path/to/file /local/destination
```

Example Output:

```
file 100% 10MB 10MB/s 00:01
```

9. `rsync` - Sync Files

Efficiently sync files between directories or hosts.

```
rsync -avz /source/ user@remote:/destination/
```

Example Output:

```
sending incremental file list
document.pdf
```

10. `netstat` - Show Open Ports

Displays active network connections and open ports.

```
netstat -tulnp
```

Example Output:

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
```

11. `ss` - Display Active Connections

Show active sockets and connections.

```
ss -tulnp
```

Example Output:

```
Netid  State    Recv-Q Send-Q Local Address:Port Peer Address:Port
```

12. `ifconfig` - Show Network Interfaces

List available network interfaces.

```
ifconfig
```

Example Output:

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

13. `ip addr show` - Show IP Addresses

Display assigned IP addresses. `bash ip addr show` Example Output: `inet 192.168.1.10/24 brd 192.168.1.255 scope global eth0`

Conclusion

Ubuntu is a powerful and versatile operating system with strong community support. Whether you are a beginner or an advanced user, mastering these commands will enhance your Linux experience. □

□ Contributing

If you have suggestions or improvements, feel free to create a pull request!

□ License

This project is licensed under the MIT License.