

LINUX

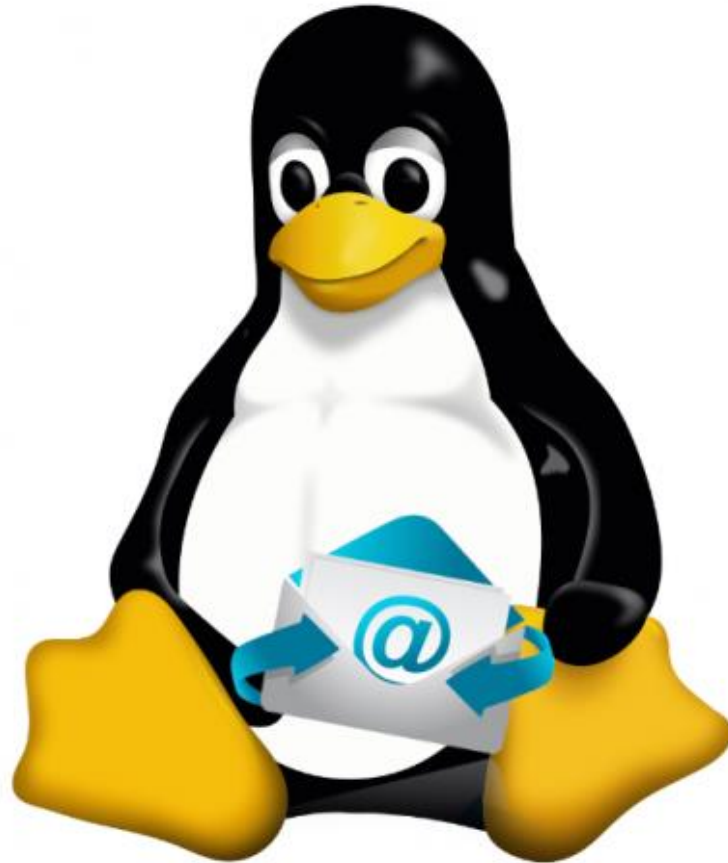
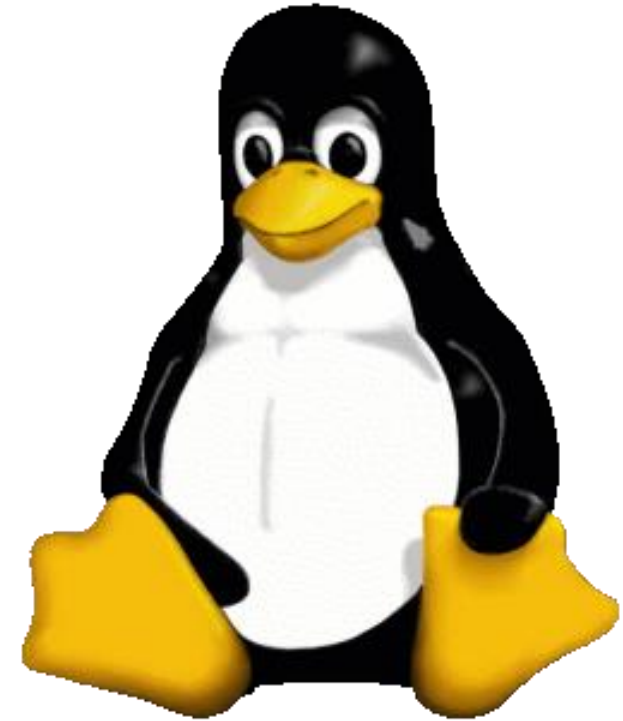


Image source: <https://www.itdev.co.uk/blog/get-your-patch-merged-journey-linux-kernel-%E2%80%93-part-3>

What is Linux Kernel?

The Linux kernel is a free and open-source, monolithic, modular, multitasking, Unix-like operating system kernel.

It was conceived and created in 1991 by Linus Torvalds for his i386-based PC, and it was soon adopted as the kernel for the GNU operating system, which was created as a free replacement for UNIX.



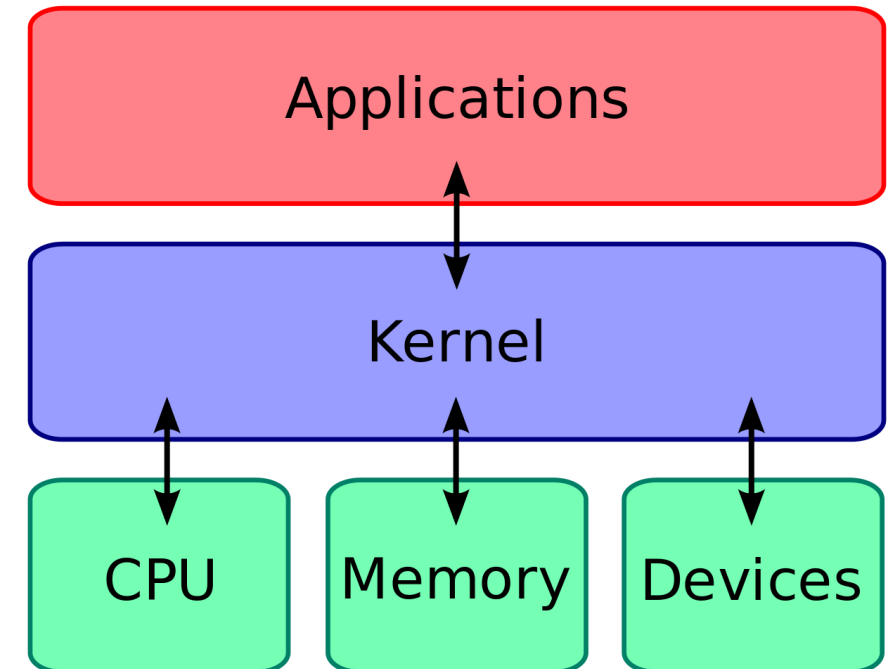
https://en.wikipedia.org/wiki/Linux_kernel

What is a Kernel?

- Kernel runs each processes and provides system services to processes, provides protected access to hardware to processes.

The kernal has following jobs:

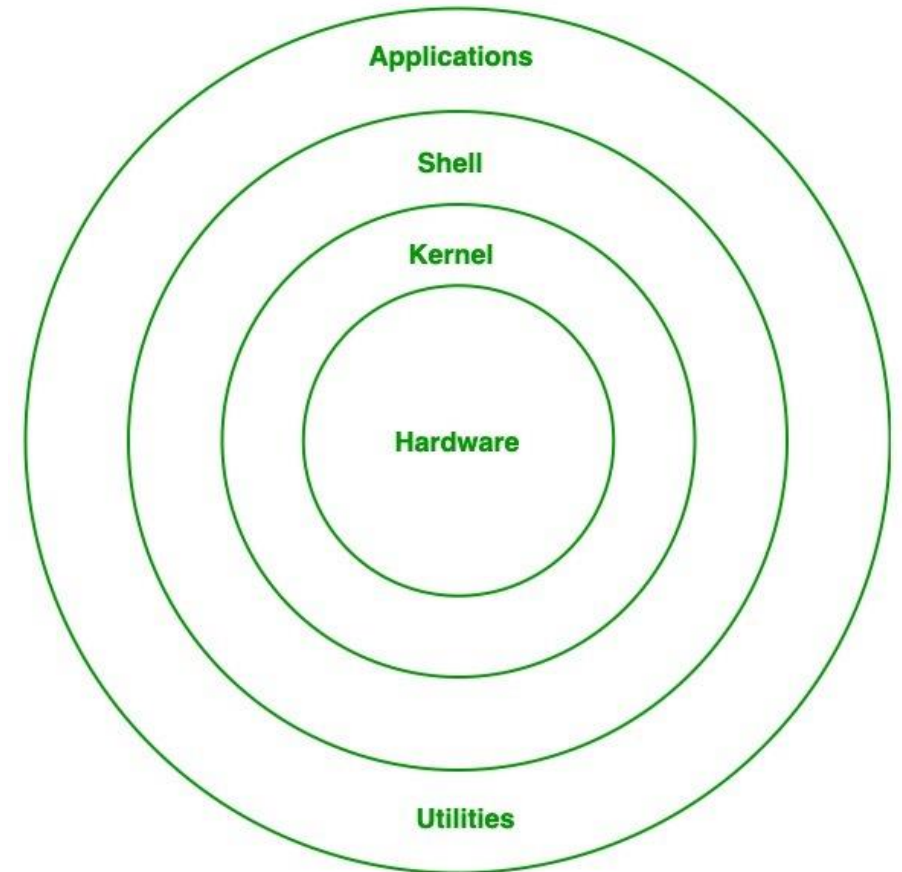
- **Memory management:** Keep track of how much memory is used to store what, and where
- **Process management:** Determine which processes can use the central processing unit (CPU), when, and for how long
- **Device drivers:** Act as mediator/interpreter between the hardware and processes
- **System calls and security:** Receive requests for service from the processes



https://blog.digilentinc.com/wp-content/uploads/2015/05/1280px-Kernel_Layout.svg.png

Linux Architecture

- Hardware Layer
- Kernel
- Shell
- Utilities



<https://media.geeksforgeeks.org/wp-content/uploads/20200105215737/Untitled-Diagram-215-1.jpg>

History of Linux

UNIX: 1969 Thompson & Ritchie AT&T Bell Labs.

BSD: 1978 Berkeley Software Distribution.

Commercial Vendors: Sun, HP, IBM, SGI, DEC.

GNU: 1984 Richard Stallman, FSF.

POSIX: 1986 IEEE Portable Operating System unIX.

Minix: 1987 Andy Tannenbaum.

SVR4: 1989 AT&T and Sun.

Linux: 1991 Linus Torvalds Intel 386 (i386).

Open Source: GPL..

Linux Features

- UNIX-like kernel.
- Features:
 - Open source.
 - Preemptive multitasking.
 - Portable
 - Security

Linux Distributions

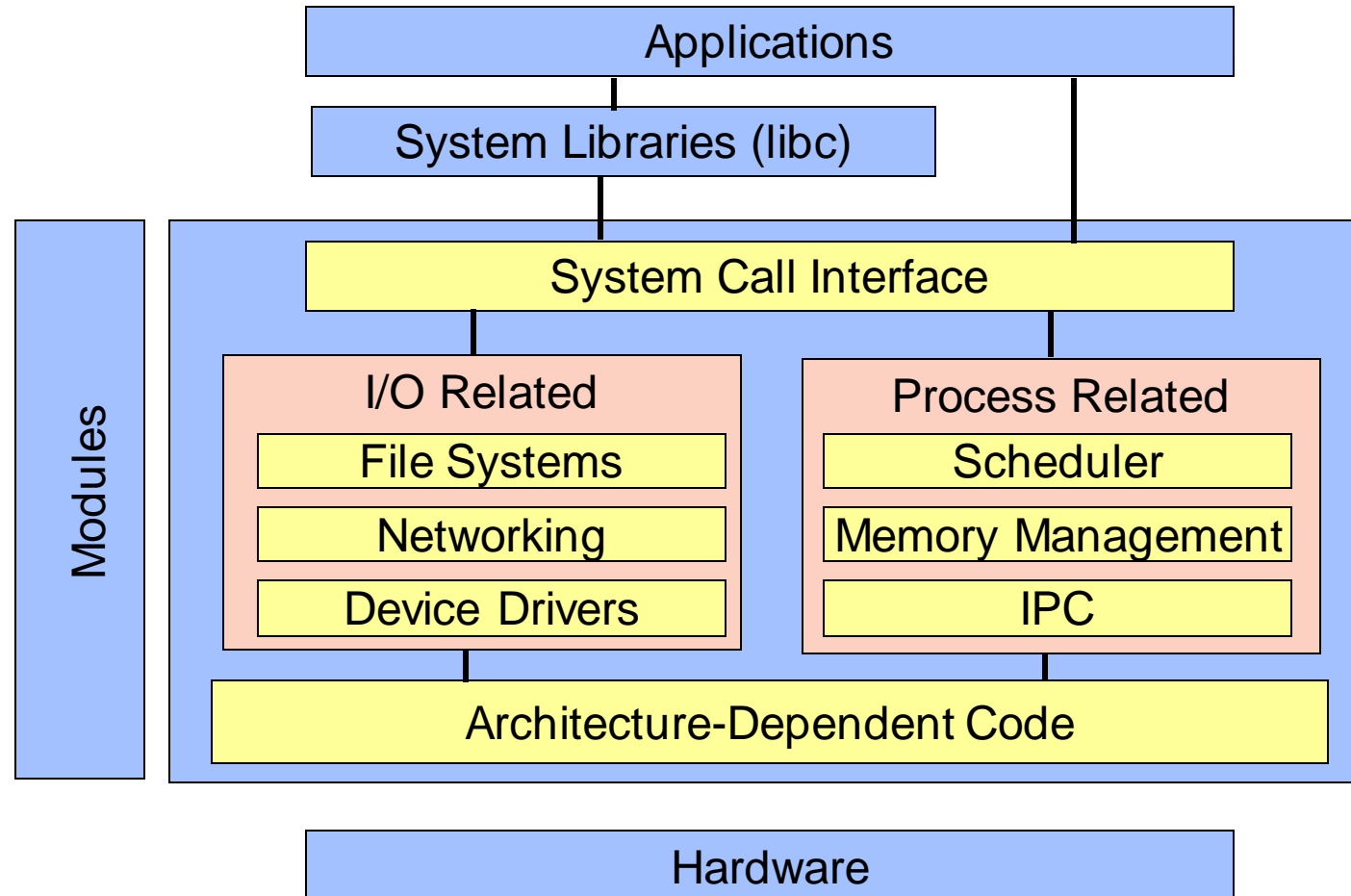
- Ubuntu
- Debian
- Fedora
- Kali Linux
- Arch Linux
- Raspberry Pi OS



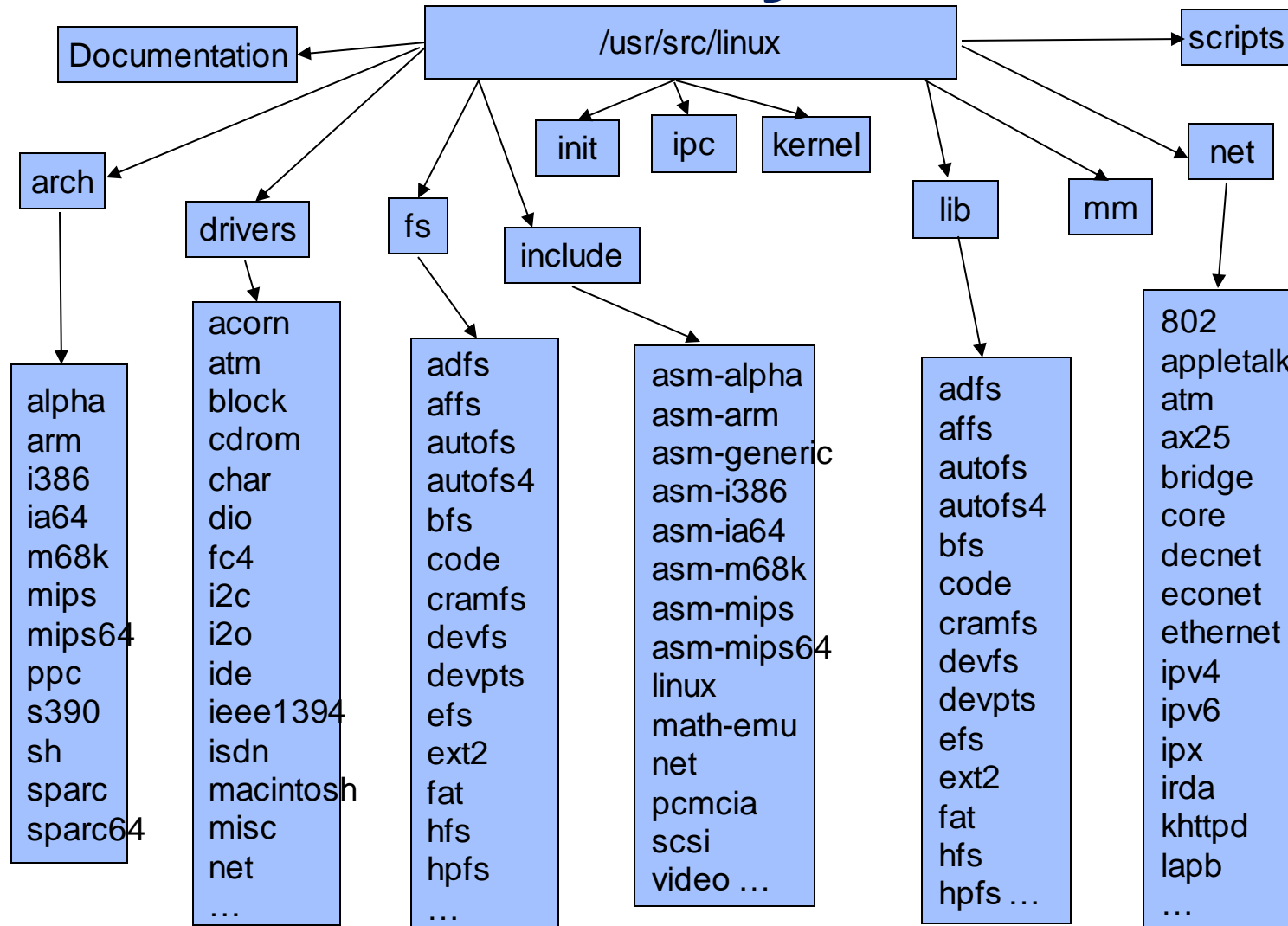
Kernel Design Goals

- Performance: efficiency, speed.
 - Utilize resources to capacity with low overhead.
- Stability: robustness, resilience.
 - Uptime, graceful degradation.
- Capability: features, flexibility, compatibility.
- Security, protection.
 - Protect users from each other & system from bad users.
- Portability.
- Extensibility.

Kernel Design



Linux Source Tree Layout



/misc	For miscellaneous purposes.
/mnt	Standard mount point for external file systems, e.g. a CD-ROM or a digital camera.
/net	Standard mount point for entire remote file systems
/opt	Typically contains extra and third party software.
/proc	A virtual file system containing information about system resources. More information about the meaning of the files in proc is obtained by entering the command <code>man proc</code> in a terminal window. The file <code>proc.txt</code> discusses the virtual file system in detail.
/root	The administrative user's home directory. Mind the difference between <code>/</code> , the root directory and <code>/root</code> , the home directory of the root user.
/sbin	Programs for use by the system and the system administrator.
/tmp	Temporary space for use by the system, cleaned upon reboot, so don't use this for saving any work!
/usr	Programs, libraries, documentation etc. for all user-related programs.

Directory	Content
/bin	Common programs, shared by the system, the system administrator and the users.
/boot	The startup files and the kernel, vmlinuz. In some recent distributions also grub data. Grub is the GRand Unified Boot loader.
/dev	Contains references to all the CPU peripheral hardware, which are represented as files with special properties.
/etc	Most important system configuration files are in /etc, this directory contains data similar to those in the Control Panel in Windows
/home	Home directories of the common users.
/initrd	(on some distributions) Information for booting. Do not remove!
/lib	Library files, includes files for all kinds of programs needed by the system and the users.
/lost+found	Every partition has a lost+found in its upper directory. Files that were saved during failures are here.

ls command

ls -a (all) Lists all the files (including .* files)	ls -S (size) Lists the biggest files first
ls -l (long) Long listing (type, date, size, owner, permissions)	ls -r (reverse) Reverses the sort order
ls -t (time) Lists the most recent files first	ls -ltr (options can be combined) Long listing, most recent files at the end

Lists the files in the current directory, in alphanumeric order, except files starting with the “.” character

cd and pwd commands

cd <dir>

Changes the current directory to <dir>.

cd -

Gets back to the previous current directory.

pwd

Displays the current directory ("working directory").

cp command

`cp <source_file> <target_file>`

Copies the source file to the target.

`cp file1 file2 file3 ... dir`

Copies the files to the target directory (last argument).

`cp -i` (interactive)

Asks for user confirmation if the target file already exists

`cp -r <source_dir> <target_dir>` (recursive)

Copies the whole directory.

mv and rm commands

`mv <old_name> <new_name>` (move)
Renames the given file or directory.

`mv -i` (interactive)
If the new file already exists, asks for user confirm

`rm file1 file2 file3 ...` (remove)
Removes the given files.

`rm -i` (interactive)
Always ask for user confirm.

`rm -r dir1 dir2 dir3` (recursive)
Removes the given directories with all their contents.

Creating and removing directories

`mkdir dir1 dir2 dir3 ...` (make dir)

Creates directories with the given names.

`rmdir dir1 dir2 dir3 ...` (remove dir)

Removes the given directories

Safe: only works when directories are empty.

Alternative: `rm -r` (doesn't need empty directories).

File access rights

Use `ls -l` to check file access rights

3 types of access rights:

- Read access (r)
- Write access (w)
- Execute rights (x)

3 types of access levels

User (u): for the owner of the file

Group (g): each file also has a “group” attribute, corresponding to a given list of users

Others (o): for all other users

Access right examples

`-rw-r--r--`

Readable and writable for file owner, only readable for others

`-rw-r-----`

Readable and writable for file owner, only readable for users belonging to the file group.

`drwx-----`

Directory only accessible by its owner

`-----r-x`

File executable by others but neither by your friends nor by yourself. Nice protections for a trap...

chmod: changing permissions

chmod <permissions> <files>

2 formats for permissions:

Octal format (abc):

$a, b, c = r*4 + w*2 + x*1$ (r, w, x: booleans)

Example: chmod 644 <file>

(rw for u, r for g and o)

symbolic format. Easy to understand by examples:

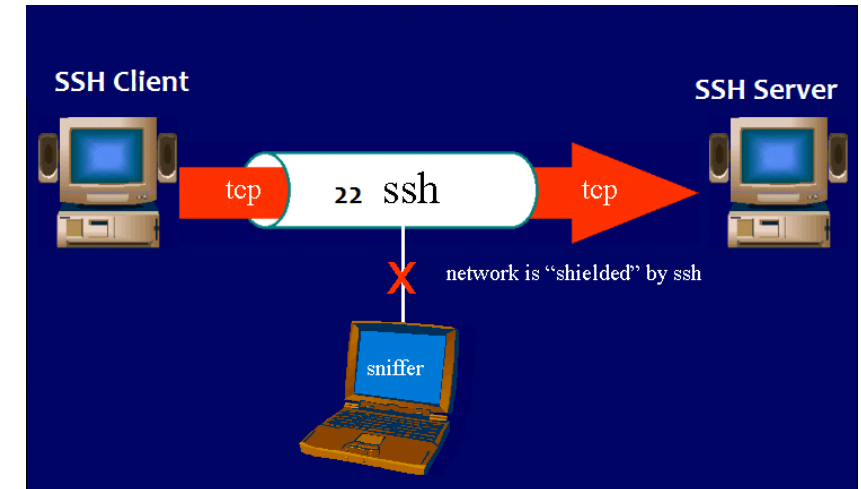
chmod go+r: add read permissions to group and others.

chmod u-w: remove write permissions from user.

chmod a-x: (a: all) remove execute permission from all.

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 encrypted form between the host and the client.
- It transfers inputs from the client to the host and relays back the output. `ssh` runs at TCP/IP port 22



`ssh user_name@host(IP/Domain_name)`

Example: `ssh root@192.168.1.1`

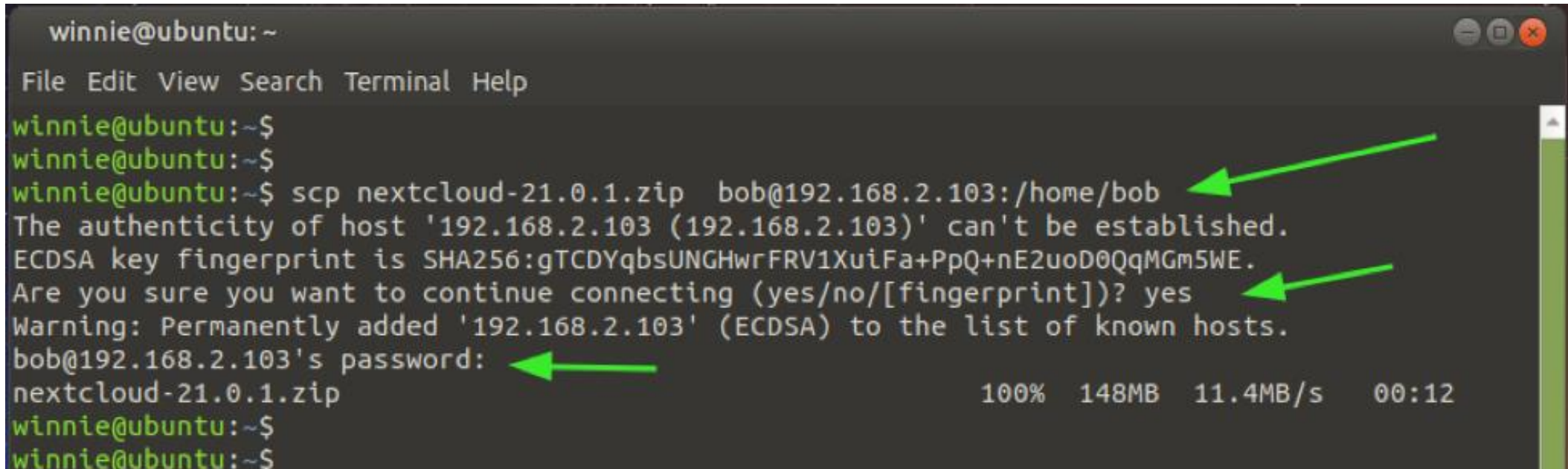
SCP

- The scp command allows you to copy files over ssh connections.
- This is pretty useful if you want to transport files between computers
- Syntax

scp examplefile yourusername@yourserver:/home/yourusername/

- Example

scp file1.pdf root@192.168.1.1:/root/Desktop



```
winnie@ubuntu: ~  
File Edit View Search Terminal Help  
winnie@ubuntu:~$  
winnie@ubuntu:~$  
winnie@ubuntu:~$ scp nextcloud-21.0.1.zip bob@192.168.2.103:/home/bob  
The authenticity of host '192.168.2.103 (192.168.2.103)' can't be established.  
ECDSA key fingerprint is SHA256:gTCDYqbsUNGHwrFRV1XuiFa+PpQ+nE2uoD0QqMGm5WE.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.2.103' (ECDSA) to the list of known hosts.  
bob@192.168.2.103's password:  
nextcloud-21.0.1.zip 100% 148MB 11.4MB/s 00:12  
winnie@ubuntu:~$  
winnie@ubuntu:~$
```

REFERENCES

1. https://en.wikipedia.org/wiki/Linux_kernel
2. https://www.tutorialspoint.com/operating_system/os_linux.html
3. <https://buildmedia.readthedocs.org/media/pdf/lym/latest/lym.pdf>
4. <https://phoenixnap.com/kb/linux-commands-cheat-sheet>
5. <https://www.guru99.com/file-permissions.html>
6. <https://www.hostinger.in/tutorials/linux-commands>

THANK YOU