

# Linux Core Concepts

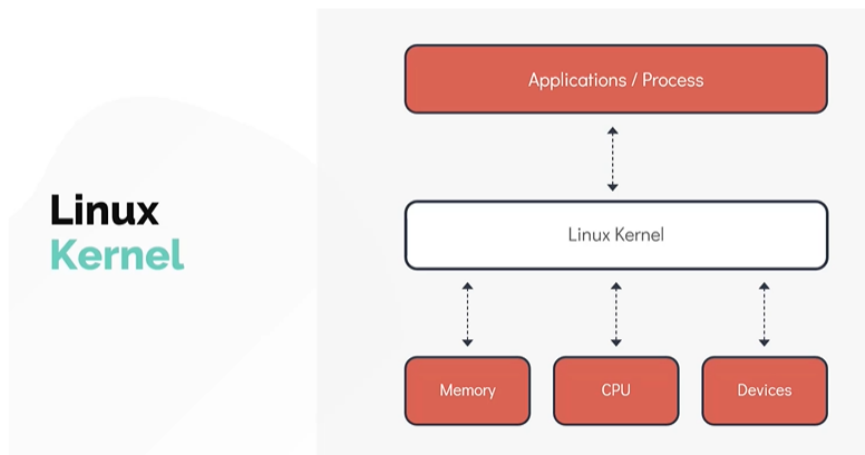
In this section, we will take a look at the core concepts of a linux operating system.

- We will start with introduction to the linux kernel.
- We will then learn about the kernel space and user space.

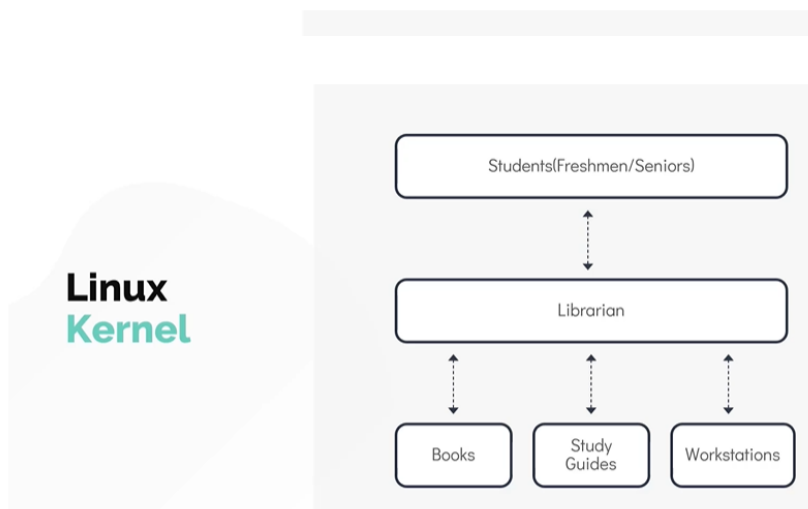
## Linux Kernel

If you have worked with any operating system, you have run into the term kernel.

- The Linux kernel is monolithic, this means that the kernel carries out CPU scheduling, memory management and several operations by itself.
- The Linux Kernel is also modular, which means it can extend its capabilities through the use of dynamically loaded kernel modules



To understand a kernel in simple terms, let us use an analogy of a **College Library**. Here the librarian is equal to Linux Kernel.



The Kernel is responsible for 4 major tasks

1. Memory Management
2. Process Management
3. Device Drivers
4. System calls and Security

## Linux Kernel Versions

let us know identify the ways to identify linux kernel versions

Use `uname` command to get the information about the kernel (by itself it doesn't provide much information except that the system uses the Linux Kernel).

```
$ uname
```

Use the `uname -r` or `uname` comamnd and option to print the kernel version

```
$ uname -r  
$ uname -a
```

## Kernel Versions

The diagram shows two terminal windows on the left. The top window shows the command `[~]$ uname` resulting in `Linux`. The bottom window shows the command `[~]$ uname -r` resulting in `4.15.0-72-generic`. To the right, a list explains the components of this version string: `4` is the Kernel Version, `15` is the Major version, `0` is the Minor Version, `72` is the patch release, and `Generic` is the Distro Specific Info. A link to <https://kernel.org> is also provided.

4	= Kernel Version
15	= Major version
0	= Minor Version
72	= patch release
Generic	= Distro Specific Info

<https://kernel.org>

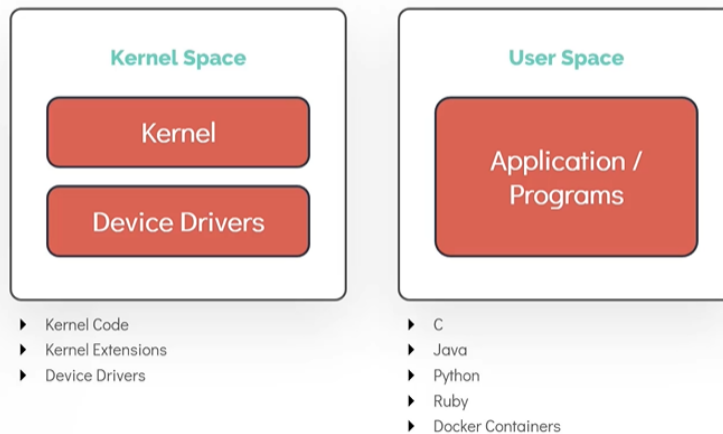
## Kernel and User Space

One of the important functions of the linux kernel is the **Memory Management**. We will now see how memory is separated within the linux kernel

Memory is divided into two areas.

1. Kernel Space
  - i. Kernel Code
  - ii. kernel Extensions
  - iii. Device Drivers
2. User Space
  - i. C
  - ii. Java
  - iii. Python
  - iv. Ruby e.t.c
  - v. Docker Containers

## Kernel And User Space

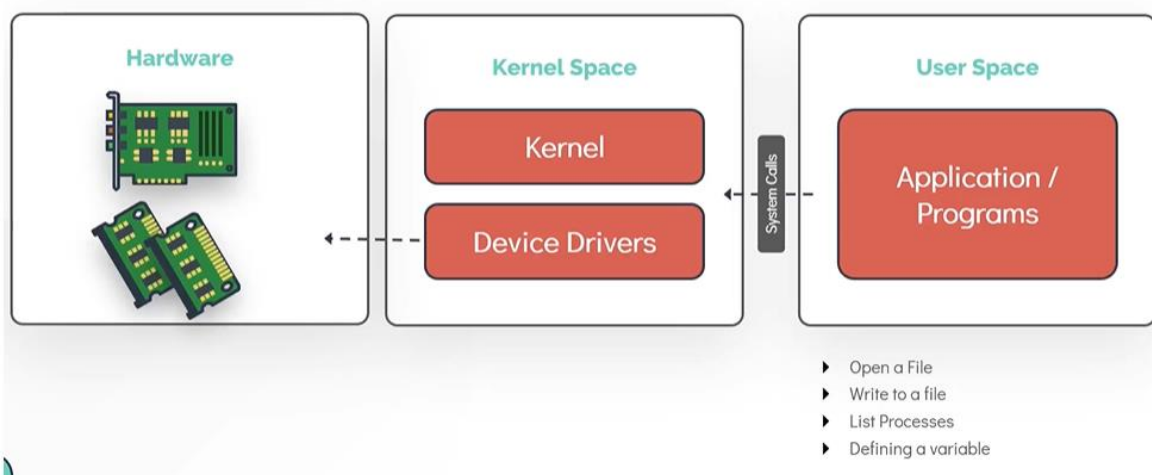


Let us know see how programs running in the **User Space** work

All user programs function by manipulating data that is stored in memory and on disk. User programs get access to data by making special request to the kernel called **system calls**

- Examples include, allocating memory by using variables or opening a file.

## Kernel And User Space



- For example, opening a file such as the `/etc/os-release` to see the operating system installed, results in a **system call**

# Kernel And User Space

