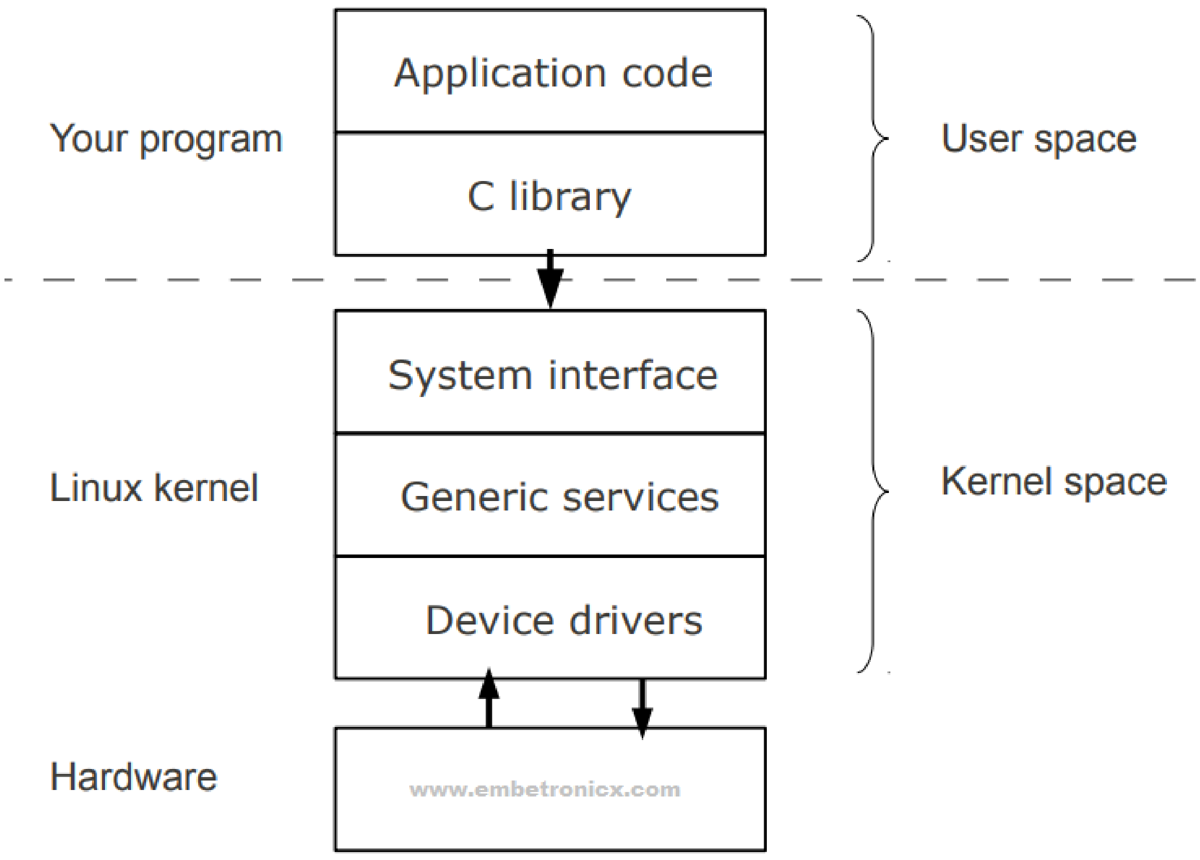
Kernel vs user space

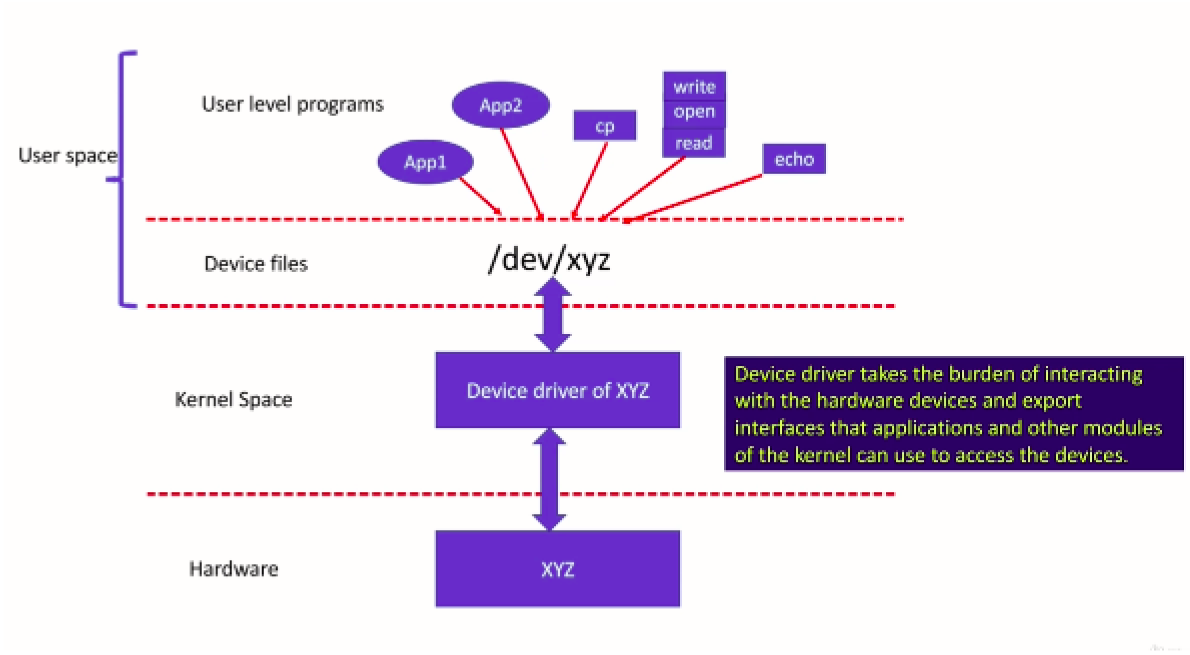


Linux Architecture

- **What is Linux?** Linux is primarily divided into ***User Space*** & ***Kernel Space***. These two components interact through a system call interface – which is a predefined & matured interface to Linux kernel for User Space applications.
  - ***Kernel Space:*** is where the kernel executes & provides its services.
  - ***User Space:*** is where the user applications are executeds.
- **What are Linux Kernel Modules?** Kernel modules are pieces of code that can be loaded and unloaded into the kernel upon demand. They extend functionality of the kernel without the need to reboot the system.
  - The basic way is to add the code to the kernel src tree & recompile the kernel.

    - A more efficient way to do this by adding code to the kernel while it is running (this process is called loading the module)
- **The kernel modules's purposes:**
    - Device drivers
    - Filesystem drivers
    - System calls

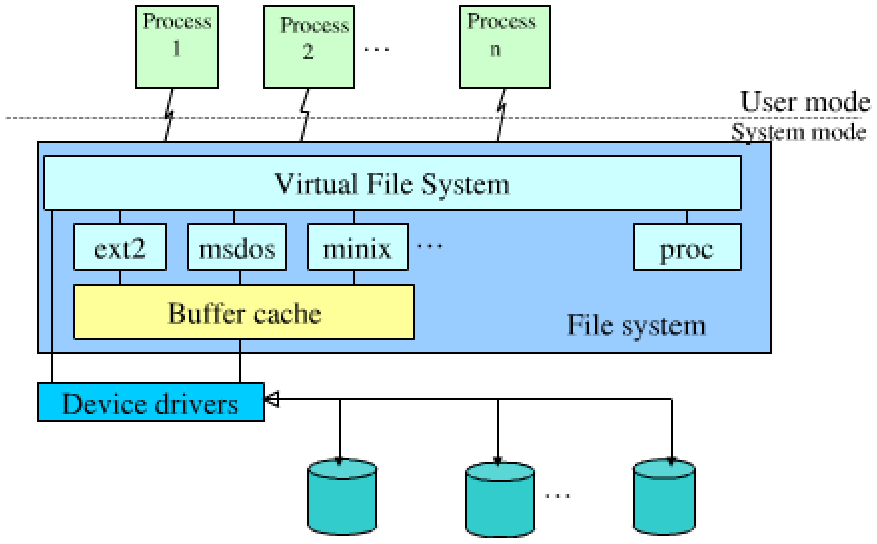
1. ***Device drivers:***


User space
User level programs
App2
App1
cp
write
open
read
echo
Device files
/dev/xyz
Kernel Space
Device driver of XYZ
Device driver takes the burden of interacting with the hardware devices and export interfaces that applications and other modules of the kernel can use to access the devices.
Hardware
XYZ

The interaction between device drivers and hardware layer in Linux

- A device driver is designed for a specific piece of hardware.
- Without having to know any details of how the hardware works

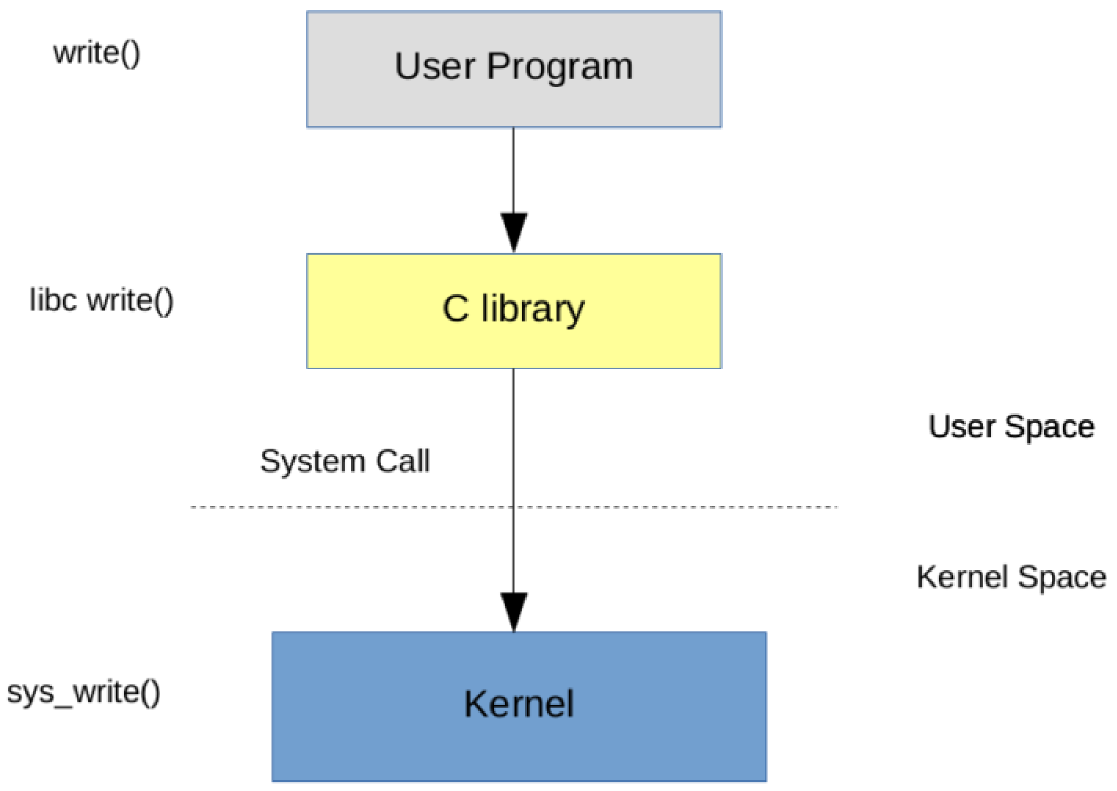
2. ***Filesystem drivers:***



???

- A filesystem driver interprets the contents of a filesystem (ex: disk drive, …) as files and directories.
- For example, there's a filesystem driver for the EXT2 filesystem type used almost universally on Linux disk drive

3. System calls:



The calling from User space to Kernel

- No LKM option
- Userspace programs use system calls to get services from the kernel

- **The difference between kernel modules and user programs?**
    - Separate address spaces
    - Higher execution privileges
    - Not execute sequetially
    - A different header files
- **The difference between kernel modules and kernel drivers?**
    - ***A kernel module*** is a bit of compiled code that can be inserted into the kernel at run-time, such as with "**insmod**" or "**modeprobe**"
    - ***A driver*** is a bit o code that runs in the kernel to talk to some hardware device