Linux Kernel Architecture

From:

Pooja Dhamija

121034

**Kernel architecture overview**

There are mainly two types of spaces:

1. User Space
2. Kernel Space

User Space is the space where user processes run which is basically application based and user space is converted to kernel space by a system call.

**Anatomy of linux kernel**

These are the modules which are in the kernel space.

In the class we have learned in detail of process management and memory management.

**File system**

It is basically used to store information on the disk and update and retrieve the information from it.

It is accessed through system calls like open, read, write etc

There are 4 types of files:

1. Ordinary Files – contain information entered by the user or any type of application
2. Directory files – Used to manage the catalogue of the file system.
3. Special files – it is used to access the peripheral devices
4. FIFO files – used for pipes

File system structure

1. Boot Block – information needed to boot the system
2. Super Block- contains the file system specifications like

Size, Maximum no of files, free blocks, free inodes

1. Inode list- Each file has a structure with a unique i-number. It basically contains the information to access the file.
2. Block list

**Virtual file system**

It provides switching fabric between users and file systems.

In the figure, the top layer is the system calls like open, read, write etc. Below are the upper layer abstractions that define how upper layer functions are implemented. They are basically the plug ins

Below is the buffer cache which is used for fast access to the physical devices. Below are the device drivers which implement the interface for the particular physical device. Device drivers interface to user applications via an entry in the /dev or /sys directories.

3 types of device drivers:

1. Character devices- access as stream of bytes like keyboard, mouse

2. Block devices – access in the multiples of block

3. Network devices – created in the linux kernel

**Networking**

At the top are the BSD sockets which is the general interface.

Below is the INET socket layer that manages the end to end communication for IP based protocols like TCP and UDP.

UDP is the connectionless protocol and unreliable. The socket is the datagram socket.

TCP is the reliable end to end protocol.

Below is the IP which helps to understand how to route incoming IP packets to either TCP or UDP layer.

Underneath this are the network devices like Ethernet, PPP.

 The ARP protocol sits between the IP layer and the protocols that support ARPing for addresses.