19115045 **Kunal Sachdeva** **3rd sem CSE**

Introduction to Unix Family
Assignment No.1

Ques: Explain interfaces to linux.

Answer:

The Linux kernel provides several interfaces to user-space applications that are used for different purposes and that have different properties by design. There are two types of application programming interface (API) in the Linux kernel that are not to be confused: the "kernel–user space" API and the "kernel internal" API.

The Linux API is the kernel–user space API, which allows programs in user space to access system resources and services of the Linux kernel. It is composed out of the System Call Interface of the Linux kernel and the subroutines in the GNU C Library (glibc).

The kernel can manage the system's hardware through what is referred to as interrupts. When the hardware wants to interface with the system, an interrupt is issued that interrupts the processor that in turn does the same to the kernel. To provide synchronization, the kernel can disable interrupts, be it a single one or all of them. In Linux, however, the interrupt handlers do not run in a process context, they instead run in an interrupt context not associated with any process.This particular interrupt context exists solely to let an interrupt handler quickly respond to an individual interrupt and then finally exit.

Interfaces:

System calls and Interrupts:

System Call Interface is the denomination for the entirety of all implemented and available system calls in a kernel. Various subsystems, such as e.g. the DRM define their own system calls and the entirety is called System Call Interface.

Applications pass information to the kernel through system calls. A library contains functions that the applications work with. The libraries then, through the system call interface, instruct the kernel to perform a task that the application wants.

Interrupts offer a way through which the Linux kernel manages the systems' hardware. If hardware has to communicate with a system, an interrupt on the processor does the trick, and this is passed on to the Linux kernel.

Linux kernel interfaces:

The Linux kernel offers various interfaces to the user space applications that perform a variety of tasks and have different properties. Two distinct Application Programming Interface (API) exist; the kernel-user space and the kernel internal. The Linux API is the kernel-userspace API; it gives access to programs in the user space into the system resources and services of the kernel. It is made up of the System Call Interface and the subroutines from the GNU C Library.

Linux ABI:

This refers to the kernel-user space ABI (Application Binary Interface). This is explained as the interface that exists between program modules. When comparing API and ABI, the difference is that ABI's are used to access external codes that are already compiled while API are structures for managing software. Defining an important ABI is majorly the work of Linux distributions than it is for the Linux kernel. A specific ABI should be defined for each instruction set, for example, x86-64. End-users of Linux products are interested in the ABIs rather than the API.

System Call Interface:

As earlier discussed, this plays a more prominent role in the kernel. It is a denomination of the whole part of all existing system calls.

The C standard library:

All the system calls of the kernel are within the GNU C Library whereas, the Linux API is comprised of the system call interface and the GNU C Library, also called glibc.

Portable Operating System Interface (POSIX):

POSIX is a collective term of standards for maintaining compatibility among the operating systems. It declares the API together with utility interfaces and command line shells. The Linux API, not only has the usable features defined by the POSIX but also has additional features in its kernel:

- Cgroups subsystem.
- The Direct Rendering Manager's system calls.
- A readahead feature.
- Getrandom call that is present in V 3.17.
- System calls such as futex, epoll, splice, dnotify, fanotify and inotify.