**PRACTICAL NO:-10**

**AIM**:-**Linux kernel; learn linux kernel with respect to following points.**

****

**What is linux kernel?**

The Linux kernel is a monolithic Unix-like computer operating system kernel. The Linux family of operating systems is based on this kernel and deployed on both traditional computer systems such as personal computers and servers, usually in the form of Linux distributions,[9] and on various embedded devices such as routers, wireless access points, PBXes, set-top boxes, FTA receivers, smart TVs, PVRs, and NAS appliances. The Android operating system for tablet computers, smartphones, and smartwatches uses services provided by the Linux kernel to enable its functionality. While the adoption on desktop computers is low, Linux-based operating systems dominate nearly every other segment of computing, from mobile devices to mainframes. As of June 2017, 498 of the world's 500 most powerful supercomputers run Linux (The remaining two run AIX, a proprietary Unix operating system on IBM POWER7 hardware).[10]

**Operating Model:-**

The Linux kernel was conceived and created in 1991 by Linus Torvalds[11] for his personal computer and with no cross-platform intentions, but has since expanded to support a huge array of computer architectures, many more than other operating systems or kernels. Linux rapidly attracted developers and users who adopted it as the kernel for other free software projects, notably the GNU Operating System.[12] The Linux kernel has received contributions from nearly 12,000 programmers from more than 1,200 companies, including some of the largest software and hardware vendors.[13][14]

The Linux kernel API, the application programming interface (API) through which user programs interact with the kernel, is meant to be very stable and to not break userspace programs (some **programs,** such as those with GUIs, rely on other APIs as well). As part of the kernel's functionality, device drivers control the hardware; "mainlined" device drivers are also meant to be very stable.

**Licensing Model:-**

The Linux kernel is licensed explicitly only under version 2 of the GPL,[6] without offering the licensee the option to choose "any later version", which is a common GPL extension. There was considerable debate about how easily the license could be changed to use later GPL versions (including version 3), and whether this change is even desirable.[37] Torvaldshimself specifically indicated upon the release of version 2.4.0 that his own code is released only under version 2.[38] However, the terms of the GPL state that if no version is specified, then any version may be used,[verification needed] and Alan Cox pointed out that very few other Linux contributors had specified a particular version of the GPL.[39]

In September 2006, a survey of 29 key kernel programmers indicated that 28 preferred GPLv2 to the then-current GPLv3 draft. Torvalds commented, "I think a number of outsiders... believed that I personally was just the odd man out, because I've been so publicly not a huge fan of the GPLv3."[40] This group of high-profile kernel developers, including Linus Torvalds, Greg Kroah-Hartman and Andrew Morton, commented on mass media about their objections to the GPLv3.[41] They referred to clauses regarding DRM/tivoization, patents, "additional restrictions" and warned a Balkanisation of the "Open Source Universe" by the GPLv3.[41][42] Linus Torvalds, who decided not to adopt the GPLv3 for the Linux kernel, reiterated his criticism even years later.[43]

**How development works:-**

The current development model of the Linux kernel is such that Linus Torvalds makes the releases of new versions, also called the "vanilla" or "mainline" kernels, meaning that they contain the main, generic branch of development. This branch is officially released as a new version approximately every ten weeks, after Torvalds does an initial round of integrating major changes made by all other programmers, and several rounds of bug-fix pre-releases.

As of 2015, in the current development scheme, the main branch of development is not a traditional "stable" branch; instead, it incorporates all kinds of changes, including both the latest features, and security and bug fixes. For users who do not want to risk updating to new versions containing code that may not be well tested, a separate set of "stable" branches exist, one for each released version, which are meant for people who just want thesecurity and bug fixes, but not a whole new version. Thesebranches are maintained by the stable team (Greg Kroah-Hartman, Chris Wright, and others).

The development model for the 2.6 kernel series was significantly different compared to the 2.5 series. Before the 2.6 series, there was a stable branch (2.4) where only relatively minor and safe changes were merged, and an unstable branch (2.5), where bigger changes and cleanups were allowed. Both of these branches had been maintained by the same set of people, led by Torvalds. This meant that users would always have a well-tested 2.4 version with the latest security and bug fixes to use, though they would have to wait for the features which went into the 2.5 branch. The downside of this was that the "stable" kernel ended up so far behind that it no longer supported recent hardware and lacked needed features. In the late 2.5 kernel series, some maintainers elected to try backporting of their changes to the stable kernel series, which resulted in bugs being introduced into the 2.4 kernel series.