Operating Systems

Kernel

Kernels are interfaces between computer Hardware and the greater software, including the operating system. Depending on the type of kernel, many features are included, revolving around low-level operating system structure and communication (Thakur, 2020). Kernels come in three flavours, which describes the overall functionality and scale of the kernel:

- Monolithic Kernels, which involve a larger kernel which controls much of the functionality required by a kernel. This includes CPU Scheduling, Memory Management and I/O & Device Management (TechDifferences, 2020).
- Microkernels differ from monolithic kernels by being much smaller, making them more efficient for storage and processing, as well as being much simpler to produce. These kernels provide basic functionality, such as Scheduling and Memory Management, while relying on user space services for I/O & Device Management, File System integration and process management (TechDifferences, 2020).
- Hybrid Kernels are a mix of the two, with varying functionality blending together the benefits of the modularity and simplicity of a microkernel with the functionality and efficiency of a monolithic kernel (Thakur, 2020).

CPU Scheduling

Modern CPUs rely on queue of instructions for operation, called a schedule. This schedule dictates which instructions are handled by the processor and in which order, with interrupts allowing for high priority instructions. Various different scheduling algorithms are used, with different priorities and implications for each of them (Studytonight, 2020).

- Round Robin scheduling assigns a fixed amount of time to each process, using a circular queue for every current process until completion. This scheduling method has a delay for each process, but equally assigns time to each process in the short term (Studytonight, 2020).
- First Come First Serve scheduling completes each process as it arrives, using a simple queue where new tasks are put at the back of the queue. This algorithm can have a higher delay than round robin, and functions badly with long processes (Studytonight, 2020).
- Multilevel Feedback Queue scheduling is a different kind of scheduling system, utilising many other scheduling algorithms in the same system, with many queues each with separate algorithms. This algorithm works by having methods for tasks to change queue, revolving around priority or age. And example MFQ scheduler could have three queues, two with round robin scheduling with different time steps, and another with first-come-first-serve scheduling (Studytonight, 2020).

Memory Management

Memory management controls integration between CPU cache memory, RAM, and secondary memory. This involves moving instructions and data to and from each different area of storage depending on priority and quantity. This system is usually included within the kernel, and allows for processes to have easy management for and access to memory (Techopedia.com, 2020). Memory managers also have features that help to optimise memory usage and access, such as compaction, paging, and segmentation:

- Compaction involves removing empty space between chunks of assigned memory, creating a large chunk of free memory. This allows for less complicated memory assignment, minimising fragmentation (Tutorialspoint, 2020).
- Paging involves breaking down memory into pages, which are blocks of memory between 512 and 8192 bytes. Paging allows for memory addresses to be smaller, reducing the complexity of addressing sections of memory, as well as simplifying address assignment. This process utilises address translation, by having addresses for data within a page, as well as an offset for which page is being referenced (Tutorialspoint, 2020).
- Segmentation is a similar process to paging, where multiple addresses are combined into segments. It differs from paging by having variable-sized segments, rather than fixed sized pages. This can simplify memory access and assignment, with address translation using a segment map to identify each segment (Tutorialspoint, 2020).

Drivers

While systems with a monolithic kernel rely on the kernel to integrate I/O and device management, other types of kernels rely on Device Drivers to connect and manage devices. Drivers perform the function of integrating peripherals and other inputs and outputs with the general operating system, simplifying user applications by providing a standard method for certain I/O devices' functionality (Fisher, 2020).

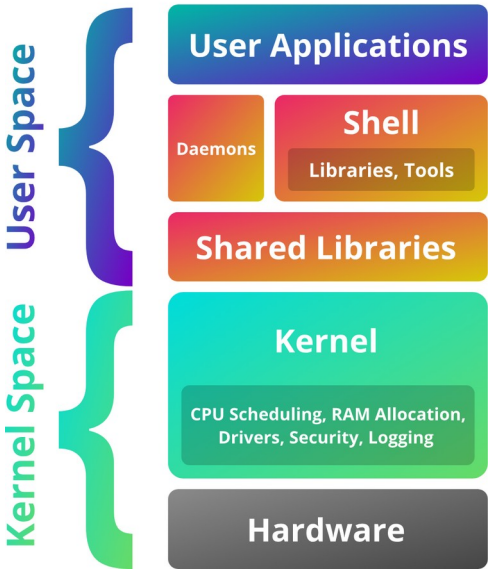
Drivers are modular, allowing for reduced system complexity by only requiring the specific drivers for each peripheral and protocol necessary. Modern operating systems can download drivers dependant on what they is connected, simplifying usage and installation (Fisher, 2020).

Hardware Abstraction Layer

Kernels provide a translation between software and hardware, while the Hardware Abstraction Layer (HAL) provides a simplified and standardised abstraction of the hardware in a computer. This means is that the HAL conceals differences in the physical hardware, simplifying the usage and development of a kernel. Without a HAL, kernels would require the ability to communicate with many different types of hardware architectures, which can be very complicated. HALs also provide direct access to hardware devices for device drivers (Techopedia.com, 2020).

Linux Architecture

Linux is a family of operating systems, originating from an open-source, Unix-like kernel developed by Linus Torvalds in 1991 (The Info Cave, 2020). The Linux kernel is a monolithic kernel (How-To Geek, 2020), differing from the Windows or MacOS kernels, which are both Hybrid Kernels (BitFeed.co, 2020) (Pediaa.com, 2020). This kernel architecture has a larger kernel space as a result of it's design, with drivers being integrated in kernel space rather than user space. The distinction between kernel space and user space revolves around the memory allocation that modules receive, with kernel space being better protected than user space (Linfo, 2020).



This diagram shows the architecture of the Linux operating system, showing the kernel space, the user space, and the modules within each (Cumulus Networks engineering blog, 2020) The user space contains the Shell, which is the command line interface that's present in Linux, usually Bash (Bourne-Again Shell) (GNU Project, 2020), with libraries and tools being executed through shell scripts and binaries. As well as this, Daemons or Services are run alongside the shell, accessible through the command line (Boelen, 2020).

Overall, the complexity of the Linux Kernel (being the largest open-source codebase available online) (The Register, 2020) is offset by being completely free and open source, with many different versions of kernel available depending on the distribution of Linux chosen, such as Debian or Arch.

Bibliography

Dinesh Thakur. 2020. What is Kernel (Operating System)? - Definition. [ONLINE] Available at: http://ecomputernotes.com/fundamental/disk-operating-system/kernel. [Accessed 16 March 2020].

TechDifferences. 2020. Difference Between Monolithic Kernel and Microkernel(Tabular Form). [ONLINE] Available at: https://techdifferences.net/difference-between-monolithic-kernel-and-microkernel/. [Accessed 16 March 2020].

Studytonight. 2020. CPU Scheduling in Operating System | Studytonight. [ONLINE] Available at: https://www.studytonight.com/operating-system/cpu-scheduling. [Accessed 16 March 2020].

Studytonight. 2020. Round Robin Scheduling | Studytonight. [ONLINE] Available at: https://www.studytonight.com/operating-system/cpu-scheduling. [Accessed 16 March 2020].

Studytonight. 2020. First Come First Serve Scheduling | Studytonight. [ONLINE] Available at: https://www.studytonight.com/operating-system/cpu-scheduling. [Accessed 16 March 2020].

Studytonight. 2020. Multilevel Feedback Queue Scheduling | Studytonight. [ONLINE] Available at: https://www.studytonight.com/operating-system/cpu-scheduling. [Accessed 16 March 2020].

Techopedia.com. 2020. What is Memory Management? - Definition from Techopedia. [ONLINE] Available at: https://www.techopedia.com/definition/8302/memory-management. [Accessed 16 March 2020].

Tutorialspoint. 2020. Operating System - Memory Management - Tutorialspoint. [ONLINE] Available at: https://www.tutorialspoint.com/operating_system/os_memory_management.htm. [Accessed 16 March 2020].

Tim Fisher. 2020. What Is a Device Driver? [ONLINE] Available at: https://www.lifewire.com/what-is-a-device-driver-2625796. [Accessed 16 March 2020].

Techopedia.com. 2020. What is a Hardware Abstraction Layer (HAL)? - Definition from Techopedia. [ONLINE] Available at: https://www.techopedia.com/definition/4288/hardware-abstraction-layer-hal. [Accessed 16 March 2020].

The Info Cave. 2020. What Is Linux: An Overview of the Linux Operating System - The Info Cave. [ONLINE] Available at: https://theinfocave.com/what-is-linux-an-overview-of-the-linux-operating-system/. [Accessed 16 March 2020].

How-To Geek. 2020. What is the Linux Kernel and What Does It Do? . [ONLINE] Available at: https://www.howtogeek.com/howto/31632/what-is-the-linux-kernel-and-what-does-it-do/. [Accessed 18 March 2020].

BitFeed.co. 2020. XNU: the kernel hybrid for excellence! | BitFeed.co. [ONLINE] Available at: https://www.bitfeed.co/page/xnu-the-kernel-hybrid-for-excellence. [Accessed 18 March 2020].

Pediaa.com. 2020. What is the Difference Between Windows Kernel and Linux Kernel - Pediaa.com. [ONLINE] Available at: https://pediaa.com/what-is-the-difference-between-windows-kernel-and-linux-kernel/. [Accessed 18 March 2020].

Linfo. 2020. Kernel Space Definition. [ONLINE] Available at: http://www.linfo.org/kernel_space.html. [Accessed 18 March 2020].

Cumulus Networks engineering blog. 2020. Understanding the architecture of the modern Linux operating system - Cumulus Networks engineering blog. [ONLINE] Available at: https://cumulusnetworks.com/blog/linux-architecture/. [Accessed 18 March 2020].

GNU Project. 2020. Bash - GNU Project - Free Software Foundation. [ONLINE] Available at: https://www.gnu.org/software/bash/. [Accessed 18 March 2020].

Michael Boelen. 2020. Understanding what runs on your Linux system (and why) - Linux Audit. [ONLINE] Available at: https://linux-audit.com/running-processes-and-daemons-on-linux-systems/. [Accessed 18 March 2020].

The Register. 2020. Linux in 2020: 27.8 million lines of code in the kernel, 1.3 million in systemd • The Register. [ONLINE] Available at: https://www.theregister.co.uk/2020/01/06/linux_2020_kernel_systemd_code/. [Accessed 18 March 2020].