

## Introducing New Operating Systems for Computer

V. Basil Hans\*

### Abstract

*This study introduces new operating systems (OS) designed to meet the evolving demands of modern computing environments. As technology progresses, the limitations of traditional operating systems in terms of efficiency, security, and scalability have become evident. The proposed new OS models aim to address these challenges by incorporating advanced features such as optimized resource management, enhanced security protocols, and improved user interfaces. Additionally, these systems leverage emerging technologies like artificial intelligence, machine learning, and distributed computing to offer better performance, adaptability, and fault tolerance. Through comparative analysis with existing operating systems, we highlight the potential benefits and innovations brought by these new designs, envisioning a future where operating systems can seamlessly support a wide range of devices, applications, and networks. A new computer operating system is a software program designed to manage a computer's hardware and provide a user interface for interacting with it, essentially acting as the foundation for all other applications to run on a device, with recent advancements often focusing on improved performance, security, user experience, and compatibility with emerging technologies; popular examples include Microsoft Windows, Apple macOS, and Linux, each with its own strengths and target user base depending on needs like gaming, creative work, or server management. This study provides an in-depth look at the architecture, key features, and potential impact of these new OS models on both consumer and enterprise-level computing.*

**Keywords:** Apple macOS, Microsoft windows, Linux, operating systems (OS), open source, software, technology

### INTRODUCTION

An operating system (OS) is essential software that acts as an intermediary between application programs and the computer hardware, enhancing the computer's efficacy. Operating systems enable the computer hardware to comprehend software instructions, providing a user interface that protects the hardware and provides more useful functionalities to the users. Operating systems have been developed for various computing platforms; some of these, broadly distributed and utilized in businesses and homes, have gained widespread popularity. This study examines the development and functionality of modern operating systems, and considers the significance of new OS releases in terms of user

experience and improved system performance. Technological developments have significantly boosted the overall functionality of a modern operating system.

Operating systems serve as a vital interface mechanism between the hardware and corporate or individual end users, covering both professional applications and entertainment platforms. The explosive growth in size and functionality of computing platforms has cemented operating systems' core strengths, swiftly improving

**\*Author for Correspondence**  
V. Basil Hans  
E-mail: [vhans2011@gmail.com](mailto:vhans2011@gmail.com)

Research Professor, Department of Management & Commerce,  
Srinivas, University, Mangaluru, Karnataka, India

Received Date: February 26, 2025  
Accepted Date: February 27, 2025  
Published Date: March 18, 2025

**Citation:** V. Basil Hans. Introducing New Operating Systems for Computer. Journal of Operating Systems Development & Trends. 2025; 12(1): 37–47p.

information transfer, amassing data backups, and balancing resources to achieve a desired outcome. On top of this, a few lesser current developments in the academic community have either altered the fundamental structure of an operating system or made the current hardware it runs on obsolete [1]. A broad review of the historical context of operating systems, comparing both the academic discipline and commercial concern of new OS releases makes this subject both insightful and challenging. Furthermore, a well-researched argument with statistical data employs the necessary structure to examine the field's development. Conversely, newer acting paradigms, security concerns, and system resource limits highlight several emergent questions requiring further analysis or attention. Finally, these considerations are structured to provide the basis for a comprehensive comparison that draws a relevant conclusion on the topic.

## LITERATURE REVIEW

The exploration of new operating systems (OS) is an essential endeavor in the ever-evolving landscape of technology. The literature on this subject reveals a trajectory of innovation aimed at addressing the multifaceted challenges faced by users and developers alike. The early work by Farooq *et al.* emphasizes the critical role of open source software in democratizing access to technology [2]. Their analysis of Ubuntu as a high-quality alternative to Microsoft Windows highlights the necessity for cost-effective solutions that empower financially constrained individuals and organizations. This foundational understanding of open source technology sets the stage for further discussions on operating systems, particularly in the context of their adaptability and relevance in modern computing environments.

Building on this, Vichare delves into the historical evolution of operating systems, tracing their development from the late 70s and early 80s [1]. This period marked significant advancements in hardware technologies and the advent of personal computing, which catalyzed research into OS operations. Vichare discusses various innovative projects, such as Hydra and Amoeba, that tackled the complexities of multiprocessor and distributed systems. The challenges of programmability, monitoring, and debugging are critically examined, illustrating the ongoing need for OS architectures to evolve. The introduction of concepts like Exokernels and the SPIN approach further reflects the necessity for operating systems to adapt to the growing complexity of modern computing demands.

Most recently, Hè addresses the pressing need for personalized self-adapting operating systems in the cloud [3]. While traditional OS design has focused on speed, memory efficiency, and security, Hè argues for a paradigm shift that incorporates intelligence and personalized user experiences [3]. This perspective is particularly relevant in the context of personal devices that are increasingly becoming integral to users' lives. The integration of large language models (LLMs) into OS design represents a transformative approach to enhancing user interactions and software development. Hè's insights underscore the gap that exists in addressing personalized experiences at the OS level, highlighting the importance of evolving OS design to meet contemporary user expectations [3].

Together, these articles form a cohesive narrative that illustrates the evolution of operating systems from open source solutions to the necessity for personalization in modern computing. This literature review will further explore these themes, critically evaluating how the insights from these works contribute to the understanding and development of new operating systems in today's technological landscape.

The article "Critical Aspects of Modern Open Source Software Technology to Support Emerging Demands" by Farooq *et al.* provides a comprehensive analysis of the significance of open source software in the contemporary technological landscape [2]. The authors argue that software plays a crucial role in everyday life, handling various aspects of modern technology. This assertion is underscored by the increasing financial pressures that individuals and organizations face, which often limit their access to proprietary software solutions.

---

The authors present a comparative analysis between Ubuntu, an emerging open source operating system, and Microsoft Windows, a dominant proprietary system. This comparison is particularly relevant as it highlights the capabilities of open source software to meet user demands without the associated financial burden of commercial software. The authors emphasize that open source software not only provides similar functionalities as its proprietary counterparts but also offers the added advantage of flexibility and customization due to its accessible source code.

A critical evaluation of the material reveals that the authors effectively articulate the benefits of open source software in addressing the financial constraints faced by users. They argue that the availability of open source solutions can democratize access to technology, enabling a wider range of users to achieve their desired tasks without incurring significant costs. This perspective is particularly important in a world where technological advancements are often accompanied by rising costs, potentially excluding those with limited financial resources.

However, while the article successfully highlights the advantages of open source software, it could benefit from a more in-depth exploration of the challenges associated with its adoption. For instance, the authors could discuss potential issues such as the lack of official support, the learning curve associated with transitioning to a new operating system, and the security concerns that may arise from using less widely adopted software. Addressing these challenges would provide a more balanced view of the open source landscape and its implications for users.

The article "Intensional view of General Single Processor Operating Systems" by Vichare provides a comprehensive overview of the evolution of operating systems (OS) during a pivotal time in computing history, particularly in the late 1970s and early 1980s [1]. It highlights the significant innovations in hardware technologies and the emergence of personal computing, which laid the groundwork for contemporary operating systems.

One of the key insights from Vichare's work is the exploration of various projects, such as Hydra, Amoeba, and Accent, which focused on multiprocessor and distributed systems [1]. These projects tackled critical issues in programmability, monitoring and debugging, and multicore scheduling, which remain relevant in today's discussions about operating system design. The author emphasizes the challenges presented by distributed systems and the necessity of a clear formal intent to identify overarching formal issues. This perspective is crucial as it underscores the complexity involved in developing efficient and effective operating systems.

Furthermore, Vichare discusses the concept of reflectivity introduced by the Apertos project, motivated by the needs of mobile computing [1]. This idea of separating object-level and meta-level abstractions is particularly noteworthy, as it suggests a paradigm shift in how operating systems can be structured to enhance flexibility and adaptability. The discussion of Hypermedia by Nurnberg as a new computing paradigm also adds depth to the analysis, particularly in how it relates to the separation of data, structure, and behavior, which is a recurring theme in modern OS design.

The article also delves into approaches like the Factored OS (FOS), which addresses the increasing complexity of operating systems by breaking down components into smaller, more manageable units. This modular approach is essential for improving maintainability and scalability in OS development. Similarly, the Exokernel concept, which allows for application-specific customization by safely exposing physical resources, reflects a growing trend towards flexibility in OS architectures.

Additionally, Vichare mentions the SPIN approach, which leverages the Modula-3 programming language to create an extensible system [1]. This highlights the importance of programming language design in influencing operating system capabilities and extensibility. The article effectively argues that the structuring of operating systems is a common concern across various approaches, indicating a shared understanding of the challenges faced in OS development.

In "PerOS: Personalized Self-Adapting Operating Systems in the Cloud", Hè presents a compelling argument for the evolution of operating systems to meet the demands of modern users [3]. The article critiques the stagnation in OS design, which has largely focused on traditional metrics such as speed, memory efficiency, security, and scalability, while neglecting the increasingly important aspect of intelligence and personalized user experiences. Hè identifies a significant gap in existing OS frameworks, which are often ill-equipped to serve the unique needs of personal devices that have become integral to daily life [3].

The author proposes a novel operating system, referred to as `sysname{}`, which integrates large language model (LLM) capabilities to enhance user interaction and software development. This innovation is particularly relevant in the context of how personal devices function as "intimate companions" for users, suggesting a shift in the relationship between users and their operating systems. The introduction of LLMs into the OS framework represents a transformative approach, moving beyond conventional operating systems like Linux and iOS, which may not adequately address the personalized needs of users.

Hè emphasizes the importance of privacy and data management within this new paradigm [3]. The proposed features of `sysname{}`, including declarative interfaces, self-adaptive kernels, and secure data management, aim to ensure that user data is handled responsibly while providing a tailored experience. This focus on security and personalization is critical, as it addresses a growing concern among users regarding data privacy in an increasingly connected world.

The main research question posed in the article, how to develop intelligent, secure, and scalable OSES that deliver personalized experiences, serves as a foundation for the proposed innovations. Hè's work is a timely contribution to the field, as it challenges the status quo and encourages further exploration into the integration of advanced machine learning technologies within operating systems [3].

The literature on new operating systems (OS) reveals a dynamic landscape characterized by continuous innovation aimed at addressing the evolving needs of users and developers. The foundational work by Farooq *et al.* emphasizes the significance of open-source software, specifically Ubuntu, as a viable alternative to proprietary systems like Microsoft Windows [2]. Their analysis highlights the democratization of technology access through cost-effective solutions, which is crucial for users with limited financial resources. This exploration sets the groundwork for understanding the broader implications of OS development in a context where affordability and flexibility are paramount.

Building upon this foundation, Vichare traces the historical evolution of operating systems from the late 1970s and early 1980s, a period marked by significant advancements in hardware and personal computing [1]. He discusses various innovative projects, such as Hydra and Amoeba, which addressed the complexities of multiprocessor and distributed systems. Vichare's examination of programmability, monitoring, and debugging challenges underscores the necessity for OS architectures to adapt continually. Concepts like Exokernels and the SPIN approach reflect the ongoing evolution required to meet the growing demands of modern computing.

Most recently, Hè introduced the concept of personalized self-adapting operating systems, advocating for a paradigm shift that integrates intelligence and user personalization into OS design [3]. Hè critiques traditional OS metrics, which have predominantly focused on speed, memory efficiency, and security, arguing for the integration of large language models (LLMs) to enhance user experience [3]. This approach responds to the increasing significance of personal devices in users' lives, emphasizing the need for OS frameworks that prioritize personalized interactions and data privacy.

Together, these articles highlight a cohesive narrative on the evolution of operating systems, from the foundational role of open-source solutions to the pressing need for personalization in contemporary

---

computing environments. The insights presented illustrate a clear trajectory toward OS designs that not only meet traditional performance metrics but also adapt to the unique needs of users in an increasingly complex technological landscape. As the field continues to evolve, the integration of innovative concepts and user-centered approaches will be critical in shaping the future of operating systems.

## **HISTORICAL OVERVIEW OF OPERATING SYSTEMS**

In the late 1950s and 1960s, computers began to study the automation of routine jobs, such as accounting. An early operating system (OS) of this era was developed by General Motors for IBM 701 [1]. Computers of this vintage operated one job at a time on a single CPU. Programs and data on cards or tape were processed in batch mode. Once the card batch finished, a list of results was printed and, if the next program was not on site, the system was idle. The time-sharing operating system was first developed in the mid-1960s in an effort to increase CPU utilization. A time-sharing system allowed many users to interact directly with the computer. Also, a set of networked system (server) became common in universities and research institutes. This application is the first form of distributed system. After quitting the project in 1969, two of the original MULTICS developers, Brian Kernighan and Denis Ritchie, manipulated an economy model of an early 3B20, and designed an OS without hardbound in its V7 form. Since Unix was implemented almost entirely in a high-level language, the transition took only 2 years.

Time-sharing systems loosely inspired the idea of Unix. Time-sharing systems were large and expensive and required mainframe-class resources to manage many users at the same time. In the early 1970s, it was unreasonable to build a dedicated computer for an experiment. The FORTRAN IV development environment was inadequate and was not portable among different vendor systems. Only the compiled program could be executed, no possibility of dynamic binding. Despite these restrictions, the first version of Unix was written in this modest environment, eventually propelling Unix's worldwide success. Unlike other systems of the time, Unix was not designed as an extension to a particular hardware architecture, but rather as an assembly of small and understandable components. This particular design allowed Unix to be compiled in any environment where a semi-compatible C compiler was available.

## **KEY COMPONENTS OF MODERN OPERATING SYSTEMS**

The operating system (OS) on a computer is the most important software since it is responsible for monitoring a vast array of different processes that interact and share the various components of a computer system [4]. A user interacts with the OS on a daily basis; however, each version has three core components which are the most essential to a system. Each of these components plays a critical role in resource, memory, and IO management which ultimately maintains the balance and stability of the operating system. On a day to day basis, however, a user only interacts with the components which can be seen and heard such as the user interface, and the desired applications on the system. One integral part of the operating system which cannot interact with the user is strictly the kernel.

Second only to what is held in memory, the kernel manages all of a computer system's resources based on its pre-programmed policies and procedures. One of the kernel's primary tasks is handling the complex communication between the computer hardware and software, one request at a time. The foremost, and almost only goal of any operating system's kernel is to make sure requests such as read, write, and execute are not attempting to interact with unauthorized regions in memory, or hardware, and make sure such requests are carried out as efficiently as possible. In addition to these significant tasks, the kernel also undertakes interrupt handling and process management. The Resources: Inter Process Communication, Memory Management, and Scheduling, provide an overview at a high level how the kernel manages these resources. Within memory management, it will be discussed how the kernel interacts with the page table, and then in step 2 to IO, the TLB. The scheduling policy that a kernel uses can greatly affect system performance. The design of the kernel is usually fine-tuned based on a computer system's expected workloads, and so, most designs beyond this dove into a very detailed and often unnecessarily complex design.

### **Kernel**

An operating system is a software component of a computer system that is responsible for the management and coordination of activities and the sharing of the resources of a computer. The operating system acts as a host for application programs that are run on the machine. As a host, one of the purposes of an operating system is to handle the details of the operation of the hardware. This relieves application programs from having to manage these details and makes it easier to write applications. Almost all computers, including hand-held computers, desktop computers, supercomputers, and even many cellular phones, use an operating system of some type. Some of the oldest models may however use an embedded operating system, that may be contained on a compact disk or other data storage device. Browsing the Web, checking email, word processing, gaming, and the like are common uses of desktop and laptop computers along with order processing, inventory checks, and record keeping by a company. The components of an operating system all exist in order to make the different parts of a computer or a networked group of computers work together. The operating system's job is to create an efficient exchange between these different systems. The operating system has to manage all of these different components smartly, so that each of the other machines can do its job as efficiently as possible. With so many different worker machines operating at the same time, it is important that a big boss, or, a kernel is present to take care of everything. This kernel is the core of any good OS, and it is able to make all parts of a computer work together in the most productive way possible. The kernel is the first part loaded into an operating computer, and it stays protected in the memory from ordinary processes, which are the other vital applications in a computer, such as the Web browsers, document editors, and games played.

### **File System**

The entire operating system is designed for the robust handling of files, the unit of stored information. This encompasses the organization, structure, and management of data within the operating system. The clear, efficient storage and retrieval of these files, and moreover, the easy manipulation of these files. File systems enable the efficient storage and retrieval from storage devices of files created by users, operating systems, and applications. From the beginning, databases, document and spreadsheet applications, and large emails (with attachments) bombard the user's hard drive, eventually filling it to some capacity. Because of this, the organization of the data structures on the machine's storage remains a highly crucial consideration for its performance and later, even the ability to access the stored data. There are many architectures for file systems, some are highly efficient albeit of high complexity for additional features including recovery from some file corruption. Simpler and ancient architectures include FAT16, FAT32, ext4, or xfs. Most flash drives come with a FAT file system. All systems must be robust and maintain integrity; various checks may be implemented to ensure such integrity and attempts at repair or failover when these checks fail. Moreover, systems must be able to specify access restrictions on files for privacy and data security purposes. Hence mechanisms are developed to enforce file access (writing, reading, execution) and address the ever-present concern of bypassing today's security paradigm. A brief outline of common permissions and security models is given here. Furthermore, with the rise of cloud and virtualized context, file systems today must cope with new global challenges, as effectively sharing files are becoming popular. Contemporary challenges include global data access and virtual machine hard drive file porting, both forfeiting the tradition of local access of regular files.

### **Device Drivers**

Device drivers are a crucial part of the architecture of a modern operating system. A device driver is a software component that allows the operating system to communicate with a hardware device. The operating system uses device drivers to control the particular components of the hardware, sending commands to them and retrieving information as needed. The hardware, however, needs a command in a lesser abstract degree. With this respect, the driver works as a translator, bridging the gap between the operating system's generic, high degree of abstraction commands related to the piece of hardware it controls, and the hardware's more specific instructions [5]. Drivers are crucial for the normal functioning of a hardware device. Without a driver, it will be as if the hardware did not exist at all, as the operating system will not be able to understand it. One of the most crucial points about drivers is its

degree of compatibility with the hardware it controls, and its performance. The first is due to the software part of the written driver, and the second is due to the way it sends the commands to the hardware. There are two types of drivers that can be found in a system: kernel-mode drivers and user-mode drivers, or, in a more simplistic manner, normal drivers and services. The first are able to run in Ring 0, accessing both user and kernel space, while the second can only run in user mode in Ring 3, being more limited with respect to the system's internal functions. An operating system has a modular or layered design, with different parts being abstracted from each other. It is possible to build an operating system where the drivers operate in user mode, with an isolated address space, maintaining all the rest of the kernel isolated from the driver execution, to avoid crashes because of a faulty driver. Requirements regarding hardware and software platforms include stability and performance. When dealing with so many different devices and manufacturers, the multitude of software drivers can shape the user's experiences drastically. The proliferation of different drivers can also slow down the performance of OSs, as each one has to use different interfaces to perform the same tasks. Regarding the development of device drivers, it is mostly done by following the hardware specifications provided by the equipment manufacturers, which often results in drivers being written for a specific system. This practice is leaving a lot of legacy systems behind, as many tests and maintenance can be required to keep support for a driver that works in legacy system versions in future ones. This trend led to advancements in automation of the driver development process. A lot of development and research is also being done in the user of open-source driver development to help reduce the time and costs associated with writing software drivers.

### **POPULAR OPERATING SYSTEMS IN USE TODAY**

This study will cover the four most popular operating systems today. They are Microsoft Windows, Apple's macOS, Ubuntu's Linux, followed by Open Source Windows alternatives. Some people who have only used these later might be interested in seeing how their experiences compare to more mainstream systems. Questions to investigate must include: Why are these systems so popular? What problems do they solve and what still needs to be addressed? How do these systems differ and appeal to different markets, both from the point of view of user experience and underlying hardware? Consider how efforts broadening other markets to these systems and what can be learned from their progress?

The most popular operating systems today are all heavily commercialized: Microsoft Windows, Apple's macOS, and Google's Android/iOS. All of these have a strong focus on consumer products and have had a major impact on society. Yet, the fourth most popular OS today is Ubuntu's Linux, an Open Source alternative with widespread use in the developer and enthusiast communities. Despite their widespread adoption today, these platforms have seen a lot of changes throughout the years as developers compete for more capabilities. In the past, many different platforms existed and some of the less successful historical efforts can offer valuable lessons in how markets are won and lost. Only a few decades ago, UNIX was one of the most popular OS with a lot of different incompatible versions and schisms, much like the plethora of Android modifications which exist today. Even more niche operating systems have seen modern rebrands as new groups argue that their approaches are the future of computing.

### **Windows**

Windows is a series of operating systems developed by Microsoft, originally introduced in 1985, and since then it has become synonymous with personal computing [6]. The latest and most recent version of Windows is Windows 10, which was officially released in 2015. Since its introduction Windows has continued to evolve at a rapid rate. The first milestone in its development came in 1995 with Windows 95. This was followed some five years later with the release of Windows XP in 2001, which has been one of the most popular releases of Windows to date. This was followed by Windows Vista in 2006, which was infamous for its poor performance. In 2009 Windows 7 was released which brought improvements in performance. Later, the system was replaced by Windows 8 in 2012, which introduced a modern user interface. Microsoft quickly updated the system and released a refined version of the operating system with fixes and improvements in 2013. A year later Microsoft released Windows 8.1, which further refined the operating system adding back new features. As of October 2015, Microsoft's

newest version of the Windows operating system is Windows 10 [7]. Windows 10 has been designed for a personalized experience and is the most versatile and powerful version of Windows yet. Through the Start menu, the system directs the user to the user's most used applications and stands out for its performance and battery life optimization. Windows 10's curated applications come from the Microsoft Store and third-party platforms, being certified and verified apps by the Norton secured program.

Windows stands out for being designed to be both user-friendly and powerful with the user being able to move between the desktop, compressed screen and applications across multiple platforms. All-in-one security is included with built-in protections and ongoing updates. A focus on user security ensures viruses are detected and removed already built-in the system, meaning users can feel productive and secure, whilst the system is actively protecting the user's data. Windows Defender is incorporated into the system for full security confidence and offers a firewall for the protection of apps, games and movies all secure for the user. The File Explorer provides an easy way to search and locate documents, photos and music. The flexibility of the system means that everyone can work in a way that is productive, and multiple things can be done at once with the support of four applications being used simultaneously. Despite its popularity, the system is not without issues for users with its fair share of bugs, and compatibility problems. Windows is in use throughout the world, with companies both big and small using the system. With cloud based online productivity tools that work on any device and an approach to corporate IT that reduces complexity, the system remains adaptable to developing technology and the needs of global industry.

### **MacOS**

Since it has been introduced by Microsoft in 1985, Microsoft Windows is by far the most used operating system (OS) worldwide, running approximately 75% of desktop systems. Macintosh Operating System (shortly macOS) is a graphically interfaced OS developed by Apple. macOS editing is favored for aesthetics and delicate design intuition. As it is solely operated on Apple-built personal computers, consistency in hardware and software is appreciated. Apple's exquisite integration of system software with hardware components renders the unique feature of macOS, which is imperceptible for Windows due to its variable hardware selection and OS compatibility. The unique design of apps hides the complexities of the functional interaction with the system. The central core can feature the arrangement of application windows on the desktop with proximity to each other or to the screen's borders, which Mac users find useful. Nevertheless, system software compatibility issues of alternative OS lead to empathy towards Windows systems by less avid users. By 2023, this alternative "artistic" OS reaches its peak proportion among professional services and creatives, investing in the multimedia industry [3]. Windows PC share around 75% of the market worldwide is more cost-effective selection than the alternative, permanent-majority OS. Many key software solutions are built within the Windows framework and thus represent a key selection factor for professional and academic services. Major virus infections target due to user prevalence carries out comprehensive protection activities. Microsoft antivirus and malware removing software are integrated within the OS streamlined maintenance scheduling. It can be expanded with user customization and 3rd party software. It allows for data and document sharing within a local network. Random-access memory (RAM) management features high memory demanding key applications. Default system applications like Photo Viewer and Media Player are more elementary than the macOS counterparts. No split-view option has been developed natively within the productivity applications. Basic consumer support (Cortana) is available [8]. Apps are more vulnerable to compatibility issues trailing OS updates. Understanding the OS functionality through "Control Panel" is more laborious than macOS user settings.

### **Linux**

Linux is a powerful open-source operating system with high flexibility and security. There are many different Linux distributions, each tailored to meet different user needs. Ubuntu is an excellent distribution of Linux for novices to work with. The interface and design of Ubuntu are user friendly. At the same time, Fedora and Debian are comparable distributions of Linux. With Fedora being more geared towards advanced users and administrators [2].

Each of the distributions of Linux is able to manage software and libraries differently, since each version of Linux has a particular configuration. There are even distributions of Linux that are completely tailored, virtually having to be built from scratch. Linux is different than most proprietary operating systems. One of the biggest differentiators between Linux and most other operating systems is command line interaction. Much can be said about the importance of command-line interaction, but it would be easy to write a dozen essays about command lines and not come close to their significance within the Linux environment. The Linux operating system is highly customizable and tweakable. Also, Linux is dominant in server environments due to its robustness [9].

Security in Linux is managed by user permissions and system integrity. Permissions are set per user and can range from full access (super user, root, sudo) to no access at all. Just as important are system level integrity features. Operating system level integrity features prevent many of the viruses and malware that are so pervasive against Microsoft's operating systems. Be prepared, however, to spend some time managing and reconciling permissions in a Linux system, since this is a very time-consuming task. Over the last 10 years, there has been a significant shift within the Linux user community. Linux has gone from a gritty, hacker-centric operating system to being a serious competitive environment to Windows and Mac. Networking function is much deeper within the Linux operating system than Windows, and the required libraries and back-end structures are better maintained by open source groups. As such, it is natural that Linux is the chosen environment for cloud computing and the Internet of Things. Today in a modern business, Linux is as common as Windows for cloud computing and data analytics and is virtually dominant in the growing IoT environment.

## RESULTS

As of February 2025, several new operating systems (OS) have been introduced or are scheduled for release, each offering unique features and enhancements. Here is an overview of some notable developments:

### Android 16 ("Baklava")

Google has announced Android 16, codenamed "Baklava", with a release planned for the second quarter of 2025. This version marks a shift from the traditional fall release schedule, aiming to better align with device launch timelines. Android 16 will incorporate advanced AI features, including integration with Google's Gemini AI, enhancing user experience across supported devices.

### Android XR

In December 2024, Google unveiled Android XR, an extended reality (XR) operating system designed for immersive experiences. Developed in collaboration with Samsung, Android XR is set to launch in 2025 on Samsung's Moohan headset and Google's Project Astra smartglasses. The OS is heavily integrated with Gemini, Google's generative AI-powered chatbot, aiming to provide seamless and intelligent interactions in XR environments.

### BMW Operating System X

BMW introduced the Operating System X, focusing on voice commands and intuitive interaction for tasks such as music control, navigation, and vehicle settings. This system is part of BMW's Neue Klasse lineup, reflecting a shift towards more advanced infotainment technology. The first models equipped with the new system are expected to debut later in 2025.

### VisionOS 3

Apple is anticipated to release visionOS 3 in 2025, the next iteration of its operating system for the Vision Pro headset. While the exact release date is unconfirmed, it is expected to follow Apple's pattern of announcements at the Worldwide Developers Conference in June and launch in September. visionOS 3 is rumored to include new features such as Apple Intelligence, integrating AI capabilities into the Vision Pro headset.

### HarmonyOS for Laptops

Huawei plans to expand its HarmonyOS beyond smartphones to include laptops, with a rollout expected in mid-2025. This move aims to provide a unified operating system experience across Huawei devices, enhancing interoperability and user convenience.

These developments reflect the industry's focus on integrating advanced AI capabilities, enhancing user experience, and expanding the reach of operating systems across various device categories.

### CHALLENGES AND FUTURE TRENDS IN OPERATING SYSTEM DEVELOPMENT

Developing and evolving functioning operating system, within the rapidly changing computing environment, present a formidable challenge. Operating systems represent one of the most vital, yet complex, software found in systems. As such, the technology invested in operating systems is changing over time. It is thus most necessary to empirically understand the research and technology trends within operating systems. The ability to adapt to changing technology trends comes from watching and predicting such trends. In response to such necessity, empirical studies are utilized to analyze the evolution and trends within operating system research and technologies [10]. There is an interest in understanding to what extent technology trends can be controlled, such as whether research focus provides only impacts, or it is indeed related strongly to technology trends.

Developing efficient monitoring refers to examining and evaluating operations states, and its potential for dynamically changing system operations. Attempting at developing metrics that reflect the efficiency of resources utilization for the system, also need to be resolved. The altering of system contexts and its effect on metric impacts are also examined, then developing means to deal with them. A model is developed for detecting and isolating definition changes. Seven schematic and two operational contexts changes, and any subsequent root causes is likewise involved. A collaborative control strategy is then suggested towards resolving such system contexts changes.

### CONCLUSION

In conclusion, the introduction of new operating systems in 2025 marks a significant shift toward integrating advanced AI capabilities, enhancing interoperability, and expanding the functionality of devices. Notable examples include *Android 16*, which will integrate AI features to improve user experiences across a variety of devices, and *Android XR*, designed for immersive AR/VR environments. *BMW's Operating System X* and *visionOS 3* from Apple show how operating systems are evolving to manage complex user interactions in both automotive and XR settings. Moreover, *HarmonyOS's* expansion to laptops further strengthens the trend of creating seamless ecosystems across multiple device categories.

These developments suggest a future where operating systems are more interconnected, intelligent, and capable of providing richer, more personalized experiences for users across a broad range of technologies.

### REFERENCES

1. Vichare A. Intensional view of General Single Processor Operating Systems. arXiv preprint arXiv:1308.1199. 2013 Aug 6.
2. Farooq U, Iqbal MA, Shabbir U, Nazir S. Critical Aspects of Modern Open Source Software Technology to Support Emerging Demands. arXiv preprint arXiv:1109.0246. 2011 Sep 1.
3. Hè H. PerOS: Personalized Self-Adapting Operating Systems in the Cloud. arXiv preprint arXiv:2404.00057. 2024 Mar 26.
4. Vichare A. What is an OS? arXiv.org. 2025. Available from: <https://arxiv.org/abs/1112.4451>
5. Pessolani PA, Weisz G, Bardus M, Hein C. Enhancing MINIX 3. X input/output performance. In XIV Congreso Argentino de Ciencias de la Computación 2008.

6. Wilson PJ. A forensic comparison: Windows 7 and Windows 8. New York: Rochester Institute of Technology; 2013.
7. Kulshrestha AA, Song G, Zhu T. The Inner Workings of Windows Security. arXiv preprint arXiv:2312.15150. 2023 Dec 23.
8. Petsios T, Tang A, Mitropoulos D, Stolfo S, Keromytis AD, Jana S. Tug-of-War: Observations on Unified Content Handling. arXiv preprint arXiv:1708.09334. 2017 Aug 29.
9. Ovadia S. Linux for Academics, Part II: The Advantages of Free and Open-Source Software. Behav Soc Sci Libr. 2014 Jan 1; 33(1): 47–51.
10. Peng Y. Characterizing the evolution of operating systems. New Jersey, United States: New Jersey Institute of Technology; 2005.