International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

Computer Operating Systems: An Overview of Emerging Trends

Prof. Anirban Bhattacharya

M.A., M.Sc., M.C.A., M.Ed., MTech., P.G.D.B.M., C.C.N.P.*, O.C.P., M.C.I.T.P., C.E.H.*
Dept: C.S.E.; School: S.O.E.T.

Abstract

Computer operating systems (OS) and emerging trends are discussed in this article. Operating systems serve as interfaces between computer hardware and users. The purpose of the study is to determine the direction of OS for modern computing systems based on the emerging trends of OS. In order to accomplish this, the paper examines the concepts of OS, its underlying architectures, and its evolution. Furthermore, the papers discuss security issues with OS architectures and provide best practices for securing OS. A recent study found that IoT OS, Cloud OS, AI-powered OS, Blockchain OS, Hybrid OS, and Container OS are the current trends in OS. In the paper, the major OSes are compared for their strengths and weaknesses. A double-layer security approach is proposed, in which the OS is hardened with security policies and security protocols are embedded in the OS's hardware. A discovery was made regarding the operating system. It was discovered that every technology comes with its own unique architecture and OS. Such a system is worrisome for engineers to develop. As part of the design architecture, the paper proposes the development of a universal operating system for all architectures, which leaves room for further expansion while mitigating the impact of power consumption.

Key words: Operating System (OS); emerging trends; Internet of Things (IoT); cloud; Artificial Intelligence (AI), security

Introduction

The advent of modern computer technologies and their architecture have seen a sharp trend of computer operating systems being developed to run such technological architectures which provided the basis for components of the devices to communicate. This proves beyond every reasonable doubt that OS forms the foundation of every computer system in the world.

A computer operating system is defined as ‘software that controls the operation of a computer and directs the processing of programs (as by assigning storage space in memory and controlling input and output systems). The computer program provides a lucid mechanism through that the user will move with the pc and additionally the user from having to upset the problematical underlying hardware. User interfaces have evolved to a high level of complexity. This can be often in distinction to the primary days of desktop computing, once for the most part interaction mechanism was the statement. An Operating System (OS) is a computer program, which is, after being initially loaded by a boot program, capable of managing hardware and software resources of the target computer system, providing them a way to communicate (Figure 1) through API and ABI interfaces and system calls. Operating System is an essential part of computing, because without it the computer would be simply useless and unable to operate. Nowadays people tend to forget its importance, having no clue of what happens behind the fancy buttons they press to complete one action or another, which leads to the serious decrease of computer literacy among students. Linux was observed to have taken over almost all supercomputers' OS as can be seen from top500.com where supercomputers are rated for energy efficiency with their specifications. It is clear from the statistics that most scholars do not have much knowledge of current OS trends for modern technological architectures like Artificial Intelligence, the Internet of Things etc. Also, most literature gathered does not have a complete document detailing the current trends of OS put together to give readers a complete guide to OS issues and so this article seeks to close this gap. Readers will be well informed about the trend of OS and will serve as a blueprint for further research.

An operating system is a piece of software that controls how a computer's memory, processes, software, and hardware are used. Additionally, it enables interaction between users and computers. Because a user cannot operate a computer without an operating system, this element of our computers is absolutely essential. A computer system has a number of hardware and software components that aid in the processes or operation. Computer operating systems depend on a variety of components, including computer memory (RAM, ROM), storage systems (HDD, FDD, SDD), processors, and other input and output devices. A collection of software known as an operating system (O.S.) manages the hardware resources of a computer and offers fundamental computer program services. It is an interface system between the device hardware and the user. This is essential software that interfaces a computer user with computer hardware in such a way that it manages the running of all sorts of programs and guarantees the effective and easy usage of computer hardware and resources. All computer devices have an operating system in them from video game consoles, and mobile phones, to personal computers and web servers. There are several operating systems built around the world, some for private or school usage, some for-government use and users select operating systems based on professional guidance, context knowledge and need. Operating systems manage data and can be thought of as an index of logic rules to determine the structure of files. Operating systems serve as the essential building blocks of modern computing, providing communication between users, users' applications, and users' hardware. Operating systems have undergone significant change over time, responding to cutting-edge technologies and meeting new difficulties. Over the years, much research has been done on the design of new languages with more and more advanced features for programming. At the same time, a lot of other projects build new operating systems with some novel characteristics, but most of these operating systems are written in C or a similar language. Not much effort has been done to fully deploy advanced language features to build a good operating system. There are many reasons for functional languages are less popular than imperative languages. One of the most important reasons is that functional programming researchers place far more emphasis on construction of optimizing compilers than on applying the language for practical usage. In spite of this, there are quite few real-world applications developed in functional languages. We describe one more application built using an advanced functional language. Because functional languages have been thought too inefficient, they have not been used widely in systems programming, though they have many good features for system software. From what we have done, this academic paper examines the development of operating systems, and the difficulties that contemporary operating systems face, and identifies emerging trends in this dynamic area. Operating systems initially mostly focused on batch processing, in which a series of operations was carried out without user input. The role of operating systems has evolved over time, from sharing one device’s resources among many users in the mainframe era, to providing convenient user interface, storage, and networking abstractions in the personal computer era. As we transition to the ubiquitous computing era, operating systems must now manage a user’s information and computation across many computers and devices. By dividing processor time into brief time intervals, time-sharing systems like CTSS and Multics made it possible for numerous users to interact with a computer at once. Single-user operating systems started to appear at the same time as personal computers. Early personal computer use was made easier by Microsoft's Disk Operating System (DOS), which offered a command-line interface.

User-friendly computing was made accessible to the general public by the Macintosh Operating System (Mac OS), which had a graphical user interface. The emergence of networking and the requirement for multi-user capabilities led to the creation of UNIX and all of its variants. Because of its stability and scalability, UNIX has become more and more common in academic and research settings. The sharing of files and resources across networks was made possible by network operating system technology including Microsoft's Windows NT. The operating system serves as the fundamental software component that empowers all information systems, including web servers, client PCs, PDAs, and network devices (e.g., routers, and firewalls). The security of operating systems is a major concern in the realm of cyberspace and e-commerce. Issues related to operating system security are of utmost importance in the field of practical computer science; however, a complete and satisfactory resolution to the matter of computer security is still lacking. Numerous flaws or deficiencies in operating systems are responsible for numerous known vulnerabilities that have been identified so far. Any computer system's operation depends heavily on its operating system. The overall security of a computer system, including the security of all the software and applications running on that system, is significantly impacted by the security (or lack thereof) of an operating system. Any vulnerability in an operating system's security will endanger any applications that use it. Attacks or intrusions from one application into other apps may result from improper control and containment of individual applications' execution in an operating system (GIACP). This paper reviews the literature on the operating systems of computers. The literature review covers the emerging trends in computer operating systems and challenges and future directions. The paper also touches on the components of the computer operating system and security issues surrounding the OS.

HISTORICAL EVOLUTION OF OPERATING SYSTEMS

When considering the evolution of the operating system as a whole, four separate generations can be distinguished [1]. Operating systems were not present on the early computers. Each program that ran on these early computers had to contain all the necessary code to operate on the computer, communicate with the linked hardware, and carry out the intended calculation. It is clear that a faulty code could cause the computer system to stop working since communication will break down and there are no error exceptions built into the programs. According to [2], even the simplest programs became extremely complex due to this circumstance. The proprietors of the central computers began to create system software as a solution to this issue, making it easier to create and run computer programs. As a result, the first operating systems were created [3]. During the first generation's existence (1945–1955), only electronic computers and equipment for performing basic mathematical operations were available. Over the years, operating systems have changed. Before assuming its initial form, it underwent a number of alterations. With the development of new technologies, these operational adjustments improve. Early computers lacked the modern operating systems we take for granted today, such as ENIAC and UNIVAC. For each task, users had to manually modify the devices. When users submitted work in batches to be handled sequentially, the idea of batch processing evolved. This procedure was automated with the creation of the first operating systems. With the advent of mainframe computers, more sophisticated operating systems that enabled time-sharing and multitasking were created. A substantial stride in this direction was made with the introduction of IBM's OS/360 in the 1960s. It utilized virtual memory and time-sharing techniques to enable the simultaneous operation of numerous programs. The creation of UNIX by Ken Thompson and Dennis Ritchie at Bell Labs in the late 1960s had a significant impact on the development of contemporary operating systems. Its emphasis on text manipulation, file system structure, and modular layout had an impact on later operating systems. A popular open-source operating system similar to UNIX is called Linux. With the release of systems like the Xerox Star and Apple Macintosh, graphical user interfaces (GUIs) began to grow in popularity in the 1980s. With Windows 95, Microsoft Windows, which was originally a GUI shell for MS-DOS, became a full-fledged operating system. This marked a shift towards user-friendly interfaces. The 21st century brought about modern operating systems like Windows 10, macOS, and various Linux distributions. This signaled a shift toward intuitive user interfaces. Modern operating systems including Windows 10, macOS, and numerous Linux variants were created in the twenty-first century. Personal computing has been revolutionized by the emergence of mobile operating systems like iOS and Android which were made specifically for smartphones and tablets. According to [4], virtualization and cloud computing have sparked the creation of specific operating systems that are intended to function in virtualized environments. These include infrastructure as a service provider and operating systems for massive data centre management.

CORE CONCEPTS OF OPERATING SYSTEMS

The role of operating systems in managing computer resources and facilitating user interactions is pivotal. This review provides a comprehensive understanding of the core concepts that form the basis of contemporary operating systems. Process management involves resource allocation and scheduling to ensure efficient CPU utilization and multitasking capabilities. Early works like "The Structure of the 'THE'-Multiprogramming System" [5] laid the foundation for process scheduling algorithms. Memory management encompasses strategies for efficient allocation and de-allocation of memory resources. Classic references such as A Survey and Critical Review" [6] explore various memory allocation techniques. File systems organize and store data hierarchically. "The UNIX Time-Sharing System" [7] outlines the hierarchical file system structure, influencing subsequent file system designs. Device management facilitates communication between software and hardware components. References like "The I/O System" [1] discuss the challenges of device management in early systems.

Taxonomy of Computer Operating Systems

This part of the article outlines the classifications of computer OS:

- ***Batch operating system***

An operating system of this type does not speak directly to the computer. An operator groups similar jobs with the same requirements into batches. Organizing jobs with similar needs into groups is the operator's responsibility. Examples of batch operating systems include payroll systems and bank statements. One advantage of the batch operating system is the difficulty in estimating or calculating the length of time required to complete any task. When a job is in the queue, the batch system processors are aware of how long it will take. Additionally, batch systems have relatively low downtime and can be shared by multiple users. The disadvantages of batch systems include the requirement that computer operators be familiar with them, the fact that they can occasionally be expensive, and the fact that if one operation fails, the others will have to wait for an arbitrary period.

- ***Multi-programming operating system***

Simple examples of multiprogramming operating systems include having numerous programs in main memory and having the ability to keep any one of them running. The main benefit of this is better resource management. Multi-programming has a number of advantages, including faster response times and increased system throughput. The inability of users to interact with system resources is a drawback.

- ***Multi-processing operating system***

A "multi-processing operating system" is one that makes use of multiple CPUs to execute resources. The throughput of the System is increased. It has several advantages, including an increase in system throughput and the possibility to use another CPU in the case that one fails. Its disadvantage is that it could be harder to understand and more complicated as a result of the numerous CPUs.

- ***Multi-tasking operating system***

Multitasking operating systems are described as those that support the round-robin scheduling technique. It has the ability to execute multiple programs simultaneously. Its advantages include effective memory management and the capacity to utilize a multi-tasking operating system while running numerous programs simultaneously. Its potential for overheating when running numerous large programs is one of its downsides.

- ***Time-sharing operating systems***

Enough time is allotted for each task to be done in order to guarantee smooth operation. Each user receives CPU time because there is only one system in operation. They go by the name of multitasking systems as well. The task may originate from a single user or a group of users. When a task's quantum time is up, the operating system moves on to the next one. Quantum is the amount of time each task has to finish itself.

The time-sharing operating system is called IBM VM/CMS. In 1972, IBM first unveiled VM/CMS. It still provides a virtual machine environment that allows numerous users to run their own instances of operating systems and applications. Time Sharing Option (TSO): TSO is a time-sharing operating system that was created by IBM in the 1960s for its System/360 mainframe computer. It allowed multiple users to execute independent programs on the same computer at the same time. A time-sharing operating system called Windows Terminal Services enables several users to connect remotely to a Windows server. Users get immediate access to shared resources like printers and network storage and can run their own programs.

Each task has an equal chance to succeed, software duplication is less likely, CPU idle time can be reduced, and time-sharing systems allow multiple users to share hardware resources like the CPU, memory, and peripherals, which lowers hardware costs and improves efficiency. Time-sharing reduces the amount of time users have to wait to use the computer by allowing users to work concurrently. In comparison to batch processing, time-sharing is an interactive environment that allows people to communicate with the computer in real-time. With this increased productivity, more work may be completed in less time. Reliability concerns, the requirement to safeguard the confidentiality and integrity of user applications and data, and time-sharing systems' complexity and need for sophisticated software to manage several users simultaneously are some of the disadvantages of time-sharing operating systems (OS). Due to the necessity for scheduling, context switching, and other overheads associated with supporting many users, time-sharing systems have a larger overhead than other operating systems. The risk of security breaches rises when multiple users share resources. The likelihood of mistakes and errors is increased by this complexity. Time-sharing systems require strict control of user access, authentication, and authorization to ensure the security of data and software.

- ***Distributed operating system***

These operating systems are a new advancement in computer technology that is swiftly gaining traction everywhere. Many independent, networked computers communicate with one another using a shared communication network. The CPU and memory are separate for every independent system. These are referred to as distributed or loosely coupled systems. The CPUs in these systems come in a variety of sizes and capacities. Working with these operating systems has the main benefit that remote access, which is enabled within the connected devices, is always possible for one user to access files or software that aren't actually on his system but are instead present on another system connected to this network. One of the advantages of a distributed operating system is that since each system runs independently of the others, its failure won't affect how the network communicates with the other systems. Electronic mail speeds up the exchange of data, and because resources are shared, computation is exceptionally fast and dependable. These systems are easily scalable since more systems may be added to the network with little trouble, reducing the burden on the host machine. Faster data processing is a drawback of distributed operating systems. If the primary network fails, all communication will stop. The distributed system development language is still in development. These kinds of systems are uncommon because of their exorbitant cost. Not only that, but the underlying software is incredibly complicated and is yet not fully understood.

- ***Network operating system***

The management of data, users, groups, security, applications, and other networking duties is made possible by these server-based systems. These operating systems offer shared access to files, printers, security, software, and other networking features via a small private network. Another essential characteristic of these systems is that they are known as Tightly Coupled Systems since every user of Network Operating Systems is completely aware of the underlying configuration, of every other user on the network, of their unique connections, etc. Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, BSD, and others. A network operating system's benefits include centralized servers that are incredibly stable, servers that manage security issues, simple system integration of new technologies and hardware updates, and remote server access from a number of places and systems. The drawbacks to write about them. It needs specific device drivers and an interrupt signal to react as quickly as possible to disturbances. Because these systems are not very adept at switching jobs, setting thread priorities is not advised.

- ***Mobile Operating System***

The mobile operating system runs extra application software on tablets and mobile devices. They are similar to Linux and Windows in terms of software, but they are lighter and simpler. The operating systems used by smartphones most frequently are iPhone OS, Windows, Android, Symbian OS, Harmony OS, Palm OS, and Web OS. The iPhone OS ID originated from BSD and The Next Step, both of which are related to UNIX. The operating systems for a manufacturer's goods are then chosen after that. One of its benefits is how user-friendly it is. It takes the least amount of RAM and stores data on a flash drive. The mobile operating system can be used with touchscreen and touchpad mobile devices. It can also control Wi-Fi connectivity. It handles a variety of applications effectively, consumes less power, and prevents energy loss. Data backup and restore can be quite useful for the mobile operating system. It provides clients with good customization options and functions reliably across a range of devices. They start up faster as a result. Certain mobile operating systems have issues with storage needs and battery life. Updates must be constantly made in order to maintain effectiveness. The system's functionality is hampered by the several apps that are still running in the background. Operating systems for mobile devices are far less adaptable. The security of these operating systems is continuously at risk due to unauthorized applications and the process of getting rid of bugs is challenging.

Emerging Trends in Operating Systems

Emerging trends reflect the ongoing evolution of operating systems to meet the demands of modern computing environments. Operating system developers are focused on enhancing security, performance, and adaptability to address the challenges of the digital age. [8] believe that an area gaining ground is serverless architecture which abstracts server management, enabling developers to focus on code without worrying about underlying infrastructure. Platforms like AWS Lambda and Azure manage resource allocation based on the actual execution load. As IoT devices and applications grow, edge computing is becoming essential. Operating systems are adapting to manage processing at the edge, reducing latency and improving real-time data analysis. With the proliferation of Internet of Things (IoT) devices, operating systems designed specifically for resource-constrained environments are emerging. These OS are optimized for low power consumption and efficient communication. As cyber security threats also evolve, operating systems are integrating advanced security features. This includes hardware-based authentication, encryption, secure boot processes, and enhanced user privacy controls.

Operating systems are adapting to manage applications seamlessly across different cloud platforms and on-premises infrastructure, allowing for hybrid and multi-cloud deployments.

MAJOR TRENDS IN OPERATING SYSTEM

This section discusses the major trends of OS and this includes the following points.

Internet of Things OS

IoT OS's constant advancement by specialists and scientists is significant to give a platform that bolsters the most recent conventions standard for the future canny IoT. The Internet of Things is a system of various computing devices that are interconnected to each other and these are specified with unique identifiers (UI). This system has the ability to transfer data from one device to another with the help of this unique ID of a device. This data transfer does not require any human-to-human or human-to-computer interaction. The devices included in IoT can be any computing device, mechanical device or digital device. This system allows us to transfer data with reliability and accuracy over the network in minimal time possible. The operating systems developed for the IOT systems do not require large kilobytes of RAM and high-power consumption. Instead, these OSs consume low power and require a few kilobytes of RAM in order to perform their operation [9]. Different IOT standards have been introduced by various organizations such as ITU-T, IETF, IEEE, ITSE, ISO, IEC, 3GPP, and M2M to help us provide different protocols for communication [10]. Many IoT operating systems are available that provide us with a complete internet protocol networking stack with HTTP, UDP and TCP [11]. An exploratory survey of embedded devices has been carried out with a concentration of basic features such as memory, processing, size and interface [12]. The most suitable paradigm for the fastest-growing IoT is fog computing. It brings the closer connection between storage and networking with edge devices.

Parameters for Choosing IoT Operating Systems

An operating system is a type of software that is responsible for performing all the major tasks of the system and for controlling all the hardware resources attached to the system. The IoT operating system used in IoT systems must meet specific criteria. These criteria depend on some parameters, the detail of them is given below.

Architecture

Architecture is the core part of the system. It is the hardware of a system. The kernel is the most important part of the operating system. The arrangement of the kernel is built upon the type of operating system structure. The size of application programs and OS services is impacted by the structure of the operating system. Some popular types of architectures are monolithic, microkernel, modular and layered. Monolithic deals with a single large process that runs in a single address space and it does not have a specified structure. Its cost is low but it requires separate services. Microkernel architecture is simpler and has a separate process called server. It is best for many embedded OSs. Modular architecture is better than monolithic as the fault in a single module does not cause system failure as a whole.

Programming model

The programming model decides how a program is supposed to be modelled by a programmer. Traditional programming models are event-driven and multi-threaded models. Event-driven models are based on triggering an external event such as an interrupt for performing the job in an IoT system. A multi-threaded model is a more consistent and reliable model for IoT devices. It provides inter-process communication by providing a multi-threaded environment. Assembly language is considered to be the finest language for interface although high-level languages can also be considered for easy rendering but they may have restrictions at various platforms.

Scheduling policy

It determines the performance of a system. It depends upon energy efficiency, latency, performance, turnaround time, waiting time, response time and real-time capabilities. Common types of schedulers are preemptive and cooperative. A preemptive scheduler assigns CPU time for each job whereas in cooperative scheduling different tasks take different times by CPU. For IOT systems, the schedulers must be efficient and able to perform multitasking. The measure of memory of the executive’s prerequisite depends on the sort of use and the fundamental stage support. The dissemination of memory might be static or dynamic. Memory circulation is simpler through static technique, yet unique methodology can give adaptability in runtime memory obtaining. The availability of the web is an essential condition for IoT gadgets. It should be workable for the IoT associations to speak with low force utilization. Operating system underpins different conventions of availability. The IoT stack must be adaptable to be arranged to fulfil the necessities of an expansive range of IoT applications with negligible changes. In IOT plans, support for IPV6 is mandatory to have unmistakable personalities in enormous systems. Energy productivity is pivotal for battery-controlled IoT frameworks and ought to be viewed when building up an IoT operating system. Most IoT frameworks are asset-bound in nature. Consequently, battery or other compelled vitality sources are utilized to work it. Situations for IoT usage are different, troublesome and at times far off. Humongous IoT arrange size requires the IOT operating system to work the IOT gear for a long time to be power proficient.

Cloud Operating System

To comprehend the essence of a cloud software package, it is of utmost importance to initially grasp the concept of cloud computing. In its most basic form, cloud computing entails the utilization of software and hardware hosted on external servers to support an organization's systems and computers. The sole requirement is the installation of an application that serves as an intermediary between the service and the user [13]. This approach offers significant advantages in terms of power efficiency, as it diminishes energy consumption and processing time by transferring most of the tasks to the server while handling user data. This document is stored on a server located at a considerable distance from the current location and diligently records every keystroke as well as any revisions made. The origination of cloud computing can be traced back to the 1950s when the academic and corporate domains employed computers that could be accessed through terminals, which were subsequently connected to a central computer. This pivotal advancement laid the foundation for the progression of cloud-based computing [14]. However, given the improved access to enhanced, expeditious, and more reliable internet connections, cloud operating systems are presently in the process of emerging.

How does OS work in the cloud

Most computers are equipped with an operating system (OS) that operates from the computer's local storage, whereas a cloud OS or internet OS functions from a distant server located in a foreign country, far from the user. The resultant display that appears on the computer screen is merely a rudimentary interface, which could be as basic as a web browser [15]. Furthermore, all the data is stored on this remote server. The hardware requirements for running these systems are quite minimal, and even a modest computer is capable of executing various tasks seamlessly, ranging from word processing to video editing. An outdated computer or one without the newest hardware can be revitalized using Cloud OS [16]. The only negative of cloud OS is the requirement for a steady and continuous internet connection, which can be problematic as the number of users grows and there is a growing need for quicker and more dependable online connections. Distributed cloud operating systems use cloud computing technologies to connect data and applications dispersed over numerous geographical locations. In the context of information technology (IT), the utilization of shared resources across several systems positioned in diverse locations is referred to as "distributed”. The implementation of a distributed cloud model enhances service communication on a worldwide scale while enabling more targeted communication in certain geographic areas [17]. To guarantee the decrease of latency and the improvement of efficiency in their cloud services, cloud providers use a distributed method. Furthermore, two additional distributed cloud examples that operate independently of cloud providers are public resource computing and voluntary cloud. According to [18], cloud computing has the potential to be a high-performing and economical technology. Additionally, it is anticipated that in the upcoming years, both the use of cloud computing and the associated operating systems that serve the cloud will increase dramatically. In the context of their research, the authors also looked at the foundational elements of cloud computing, such as the security issues that could develop when an operating system is solely built for the cloud and run from a centralized server. Numerous security issues are crucial in the context of cloud computing. Notably, the concerns of data integrity and privacy stand out as two of the most important security factors. Data becomes vulnerable to unauthorized access both during storage and transmission because of the fundamental nature of information being stored openly within the cloud and the lack of precise knowledge regarding the location of the data. The traditional IT ecosystems have undergone considerable change and, in this context, IT resources and infrastructure are progressively being made available online as standardized and virtualized cloud services. A significant change in how IT services are conceived of, created, implemented, scaled, updated, maintained, and acquired is being brought about by the rise of cloud computing [19]. The deployment of modern enterprise systems, the conceptualization of cloud enterprise systems, the benefits and challenges of adopting cloud enterprise systems, and the use of cloud enterprise systems within small and medium-sized enterprises are some examples of pertinent topics that are expanded upon in this study. According to [20], a variety of readily available, virtualized resources form the foundation of cloud computing, this includes the creation of operating systems in the cloud. Hardware, development platforms, and services are just a few examples of the resources that are commonly housed in centralized data centres and are dynamically reconfigured to maximize their use. Cloud service providers guarantee performance through individualized Service Level Agreements (SLAs), which are offered on a pay-as-you-go basis. The IT industry and society as a whole have undergone a substantial change as a result of this conversion of computer power into a service. There are numerous technological and societal advantages to cloud computing. The computing power comes from massively standardized and centralized data centres that house many servers and achieve significant economies of scale. From a commercial standpoint, cloud computing can provide on-demand computer capability with little initial outlay and continuous maintenance expenses. [21] gives a thorough overview of Chrome OS, a Google-developed operating system. This operating system exemplifies the idea of being incredibly straightforward and internet-focused. The rise in popularity of notebooks compact laptop computers made to facilitate internet access had a significant impact on the development of Chrome OS. These laptops are renowned for being inexpensive and for having hardware restrictions. An operating system called Chrome OS was developed by Google and is only compatible with specific hardware. It stands out from other operating systems because of its distinctive architecture. Chrome OS is specifically designed to work in unison with web applications.

How virtualization OS works in the cloud:

On a single physical host, virtual computers can be created and managed thanks to virtualization operating systems. They enable the simultaneous use of several operating systems on the same hardware. The way virtualization OS in the cloud functions is by adding a virtual layer of abstraction between the operating system and the physical hardware. This enables the use of numerous operating systems, each running in a separate environment on the same physical hardware. The virtualization operating systems VMware vSphere and Microsoft Hyper-V are two examples. A virtualization program known as a hypervisor is placed on the real hardware in order to generate a virtual machine (VM). A lower level of virtualization management is handled by the hypervisor operating system, which offers a platform for running many OSs on a single host. Each virtual machine (VM) is given a unique virtual environment by the hypervisor, complete with its own CPU, memory, storage, and network resources. As though it were running on its physical hardware, the VM can then install its own operating system and programs. The VM can be started, halted, and resumed whenever necessary once it has been built. Additionally, the VM can be moved to a different physical server without affecting its functionality. Virtualization is therefore perfect for cloud computing, where resources may be controlled and allocated in a flexible manner. There are essentially two different types of hypervisors: Type 1 hypervisors run on hardware directly, while Type 2 hypervisors run on top of a host operating system. Examples of Type 1 hypervisors include Xen and VMware vSphere, while VirtualBox is a Type 2 hypervisor.

Artificial Intelligence (AI) Operating System

The desire to increase productivity on portable devices and the quick development of artificial intelligence technologies over the previous five years have laid the groundwork for the move to an innovative computational framework. This framework is conceptualized as a contextual card system. We argue that Windows Metro got closer to encapsulating a true next-generation Operating System than earlier projects like Google Now and Windows Metro. The cards in Metro's User Interface were little more than representations of already-existing folders, offering no advantages above a more conventional display. It did, however, come with the drawback of changing a user's workflow. Furthermore, there were no developer tools or predictive or contextual APIs that made use of the new design. In essence, it offered a fascinating design concept but lacked the requisite artificial intelligence to provide a cutting-edge experience.

Characteristics of an artificial intelligence-first operating system

The subsequent rational progression is an Operating System that prioritizes Artificial Intelligence. This objective presents a formidable task that necessitates various stages, yet the pivotal technological advancements that delineate a subsequent-generation OS encompass the following: Contextual cards and disappearing applications. The primary means of interaction with a novel operating system would entail the implementation of a card-based system, wherein each card possesses the ability to embody a microformat containing dynamic data, contextual intention, and visualization of said data. These cards would be arranged in a hierarchical order based on their significance to the user and their intent, which is determined by the analysis of historical data patterns within a specific context. The fundamental shift in this paradigm lies in the elimination of applications in the conventional sense, thereby allowing a card to represent various elements such as the user's upcoming meeting, inclusive of relevant individuals, locations, and associated documents. This representation is contingent upon the user's intention.

Extensible semantic framework

Navigation applications possess knowledge of the notion of location, while email applications possess knowledge of files or contacts, and so on. Presently, the representation of semantic data is specific to each application and encompasses only a limited number of dimensions, such as time and space, as described in the operating system. However, in an operating system centred on Artificial Intelligence, the paradigm is inverted, wherein the operating system possesses an internal representation of knowledge and applications derive their data structure from it. This paradigm facilitates seamless communication between diverse applications without necessitating the manual integration of APIs or reliance on third-party services. Moreover, it enables the operating system to invoke multiple applications collaboratively to address a problem. Based on this idea, the operating system can create a plan and employ different programs to carry out each phase of the plan when the user expresses their wants, such as "going out for dinner.

Additionally, the operating system might automatically search online in a "feature store" for a suitable functionality if none are built into the device, as the user's intents, expressed in the framework's language, define the necessary functionalities.

User modelling

The prevailing trend that is currently occurring is the transition towards an operating system that is designed according to the user's data. In the past, there has always been a certain level of personalization in operating systems, whether it be through scripts or colour preferences. However, the next advancement involves the system analyzing the user's data, comprehending it, categorizing it, establishing connections within it, and subsequently generating responses based on this information. The central aspect of this process is user modelling, which is supported by a collection of predictive application programming interfaces (APIs) that must be made accessible to developers. These APIs include functionalities such as determining what the next action should be, which itself encompasses a range of possible options depending on whether the subsequent action involves travel, communication, or reading.

This Table 1 discusses the strengths and weaknesses of the major OS trending in a tabular form and proposes future works to mitigate its weaknesses.

The major trending OS seen in Table 1 has security issues that need to be looked at urgently and addressed to prevent hackers from penetrating the system. It is however worth noting that the OS that has better security protocols is the cloud OS since professionals who are experts in the field are dedicated to providing robust security protocols since services are metered. On the other hand, in the AI-OS and IoT OS, users are mostly responsible for updating security issues of the OS, as such they may not have the expertise to properly secure their data. In terms of cost and complexity, the IoT OS is light and consumes less power and memory space thereby being fast in terms of performance as compared to the other OS. The AI-OS requires no human reasoning and assistance as compared to the other OS, thus eliminating the common sense of decision-making and judgment. Based on its self-learning abilities, the AI-driven OS may learn bad experiences that may defeat the purpose and functionality of the OS. In terms of availability, the AI OS and the IoT OS are more readily available as compared to the cloud OS since the cloud OS constantly needs an internet connection. Access to the user's data for computation is very important because the OS is the gateway to other applications on the computer system. Clients must be able to access their data at any point in time and it can clearly be seen that the cloud OS is disadvantaged as compared to the AI OS and IoT OS. With issues of data privacy, the IoT OS does not expose the privacy of the client's data as compared to the Cloud OS and AI-OS since these two have issues of privacy preservation in the sense that the Cloud OS, data is controlled by the service providers. Also, with the Al-OS, the self-learning ability of AI can understand users' data and trigger access or sending to unauthorized parties. Lastly, in terms of complex computations, predictions and performance, the AI OS is better off than the cloud OS and IoT. The AI OS learns quickly and has high reasoning abilities due to its complex engineering process and can make a very accurate decision as compared to Cloud OS and IoT OS but it lacks common sense.

Security Of Operating Systems

Most modern computer systems offer the capability to concurrently execute multiple applications on a single physical computing hardware, which may consist of multiple processing units. In these situations where

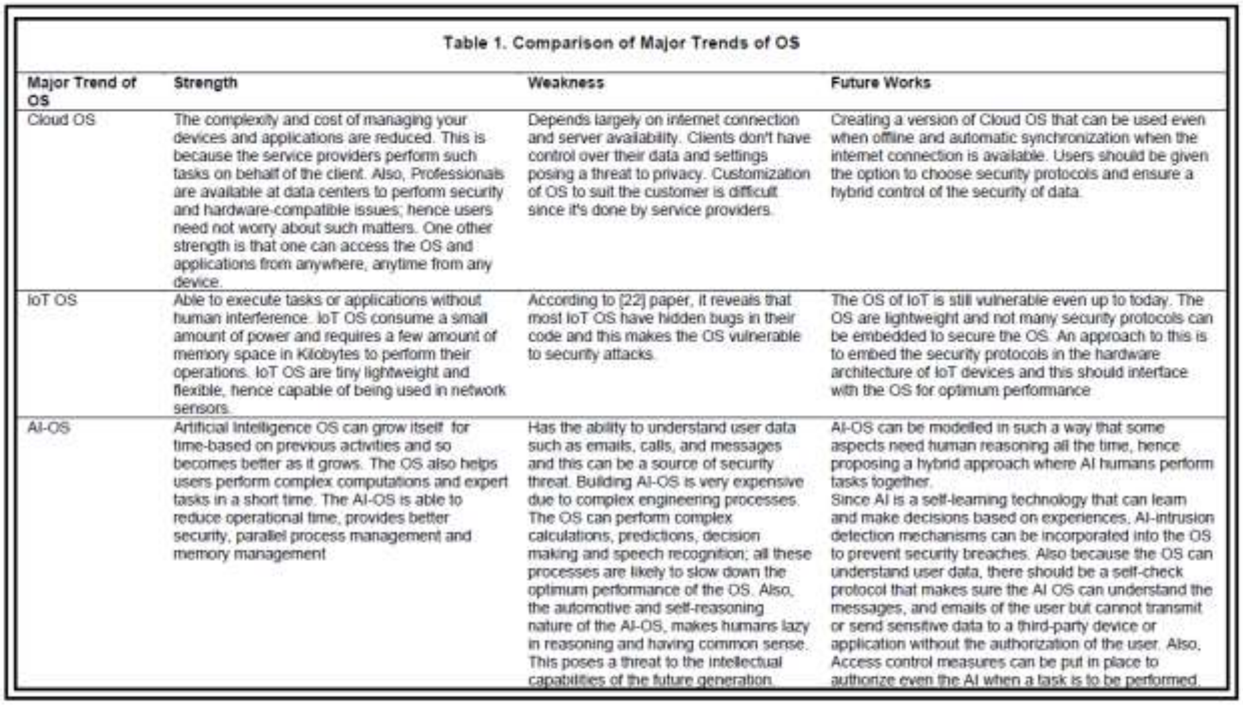

Table 1. Comparison of Major Trends of OS

Major Trend of OS	Strength	Weakness	Future Works
Cloud OS	The complexity and cost of managing your devices and applications are reduced. This is because the service providers perform such tasks on behalf of the client. Also, Professionals are available at data centers to perform security and hardware-compatible issues; hence users need not worry about such matters. One other strength is that one can access the OS and applications from anywhere, anytime from any device.	Depends largely on internet connection and server availability. Clients don't have control over their data and settings posing a threat to privacy. Customization of OS to suit the customer is difficult since it's done by service providers.	Creating a version of Cloud OS that can be used even when offline and automatic synchronization when the internet connection is available. Users should be given the option to choose security protocols and ensure a hybrid control of the security of data.
IoT OS	Able to execute tasks or applications without human interference. IoT OS consume a small amount of power and requires a few amount of memory space in Kilobytes to perform their operations. IoT OS are tiny lightweight and flexible, hence capable of being used in network sensors.	According to [22] paper, it reveals that most IoT OS have hidden bugs in their code and this makes the OS vulnerable to security attacks.	The OS of IoT is still vulnerable even up to today. The OS are lightweight and not many security protocols can be embedded to secure the OS. An approach to this is to embed the security protocols in the hardware architecture of IoT devices and this should interface with the OS for optimum performance
AI-OS	Artificial Intelligence OS can grow itself for time-based on previous activities and so becomes better as it grows. The OS also helps users perform complex computations and expert tasks in a short time. The AI-OS is able to reduce operational time, provides better security, parallel process management and memory management	Has the ability to understand user data such as emails, calls, and messages and this can be a source of security threat. Building AI-OS is very expensive due to complex engineering processes. The OS can perform complex calculations, predictions, decision making and speech recognition; all these processes are likely to slow down the optimum performance of the OS. Also, the automotive and self-reasoning nature of the AI-OS, makes humans lazy in reasoning and having common sense. This poses a threat to the intellectual capabilities of the future generation.	AI-OS can be modelled in such a way that some aspects need human reasoning all the time, hence proposing a hybrid approach where AI humans perform tasks together. Since AI is a self-learning technology that can learn and make decisions based on experiences, AI-intrusion detection mechanisms can be incorporated into the OS to prevent security breaches. Also because the OS can understand user data, there should be a self-check protocol that makes sure the AI OS can understand the messages, and emails of the user but cannot transmit or send sensitive data to a third-party device or application without the authorization of the user. Also, Access control measures can be put in place to authorize even the AI when a task is to be performed.

multiple tasks are being performed simultaneously and resources are being shared, such as the central processing unit (CPU), storage, memory, and others, the operating system provides guidance. To ensure the preservation of each application's integrity and protect against potential interference and attacks from other tasks, contemporary operating systems incorporate various abstract constructs, such as processes (or tasks) and their associated Task Control Blocks (TCBs), virtual memory spaces, files, ports, and inter-process communication (IPC). The majority of commercial operating systems, such as MS Windows and UNIX, make access decisions primarily based on user identity and ownership. Unfortunately, they do not consider factors such as the user's role, program reliability, data importance or integrity, and other security-related aspects. This approach fails to restrict data flows effectively or implement a comprehensive system-wide security policy. Consequently, once an application is compromised, breaching the security of the entire system becomes relatively straightforward due to this inherent flaw. Several potential attack scenarios could arise from a corrupted program. The utilization of system resources without proper protection is considered illegitimate. To illustrate, a malicious program known as a worm can execute an attack by sending emails to all recipients listed in a user's address book, once it gains control of a user account. The subversion of protection mechanisms implemented within an application occurs through the manipulation of the underlying system. For instance, an attacker could take over the web server that is hosting the website and alter the virtual directory using Microsoft IIS. By using privileges improperly, it is possible to gain direct access to secured system resources. On a typical Unix operating system, for instance, the compromised "send mail" program may be running with root capabilities, giving the attacker superuser rights and unrestricted access to all system resources. Another tactic used is the transmission of inaccurate and misleading information about security-related choices. For instance, remote attackers can quickly access files on the remote file server by creating a fake file handle linked to Sun's NFS. Because a program running under a certain user's name will by default inherit all privileges connected with that user, it is not possible to guard against malicious code in an application just by applying the existing methods offered by most commercial operating systems.

OS Security Best Practices

Operating system security plays a crucial role in protecting the system and data from a range of cyber threats, including viruses, worms, malware, and ransomware. Furthermore, it ensures the protection of data integrity by preventing unauthorized modifications, theft, or deletion. An indispensable aspect of operating system security is the regular update of the system, as patched systems are less vulnerable to attacks. This applies regardless of whether antivirus software is installed or updated. To fortify the operating system, several security practices can be employed.

Operating system hardening

One such custom is system hardening, which involves implementing security measures and patches specifically designed for the system, including Windows, Apple OS, and Linux. By adopting this plan, the danger of cyber-attacks can be notably mitigated. However, for operating system hardening to effectively mitigate threats, it is crucial to establish a robust data backup process. This ensures the availability of a backup copy of both the data and the operating system, facilitating prompt restoration in the event of an attack. An important aspect of maintaining operating system security lies in the development and administration of security policies. These policies serve to uphold the integrity of the operating system. For instance, users who do not obey the organization's password policy will be prompted to change and update their passwords. Additionally, the strength of these passwords is evaluated through attempts to crack them, ensuring their effectiveness in preventing unauthorized access.

Authentication

Authentication is an integral practice of security, wherein software endeavours to identify and validate users in relation to the data or programs they seek to access. The operating system encompasses a range of controls that are utilized to authenticate users, ensuring that they exclusively run authorized programs to which they possess access rights. Various techniques are employed to authenticate users, including the utilization of security keys, usernames and passwords, biometrics, and multi-factor authentication. Security keys, often supplied through physical dongles, are generated to afford users enhanced security. Users are required to insert their security key into the designated slot of the machine to log in, thereby mitigating any potential security threats. As part of the authentication process, users are assigned a unique username and a corresponding password, both of which are registered within the operating system. Biometric signatures, such as fingerprint or retina scans, are utilized to authenticate users, thereby ensuring their identity.

User accounts

With regard to user accounts, it is advisable to maintain a restricted number of accounts on the server. The proliferation of accounts can complicate the system and heighten its susceptibility to vulnerabilities. Additionally, the administration of a surplus of accounts consumes valuable time for administrators. Consequently, only a limited number of trusted individuals should be granted access to the server, as this facilitates ease of maintenance.

System patches

The implementation of regular system patch maintenance is instrumental in cultivating a secure environment. It is prudent to consistently employ the most recent iterations of patches that are recommended by the operating system's vendor. These patches may encompass updates for the operating system itself as well as supplementary application patches.

One time password

In situations where users are attempting to log into the system, the utilization of unique or one-time passwords can be advantageous. These passwords are designed to be used solely on a single occasion and must be entered within a designated time frame. Examples of such one-time passwords include random numbers, secret keys, and network passwords.

Future Directions

Future operating systems are likely to integrate advanced security mechanisms, such as hardware-rooted security, secure enclaves, and AI-driven threat detection, to provide robust protection against evolving cyber threats. As quantum computing matures, there will be a need for operating systems to manage quantum processors, support quantum programming languages, and address new challenges in quantum algorithms and error correction. As technology becomes more integrated into our lives, operating systems face increasing threats from cyberattacks, data breaches, and privacy infringements. Developing robust security measures and maintaining user privacy are ongoing challenges. With the proliferation of resource-intensive applications, operating systems also need to efficiently manage hardware resources like CPU, memory, and storage to ensure fair allocation and optimal performance. Modern processors increasingly rely on multiple cores for performance. Operating systems must manage concurrency, parallelism, and synchronization efficiently to fully utilize these capabilities. Artificial intelligence and automation will likely play a significant role in future operating systems, optimizing resource management, automating system maintenance tasks, and providing personalized user experiences. Future operating systems will need to support cutting-edge innovations in Augmented Reality (AR), Virtual Reality (VR), and blockchain while also delivering seamless integration and improved performance for these novel paradigms. The study of technology in the fields of semiconductors, artificial intelligence, detection, and cognition has significantly increased recently. This development has resulted in a significant increase in the need for more complex, quick, and efficient computer devices, which has forced an adjustment in operating system development. The operating system is therefore anticipated to improve further in terms of user interface, human-machine interaction, artificial intelligence, and cutting-edge hardware technology. This development aims to make a variety of jobs easier, including database management, computing, communication, documentation, and information processing. Due to its reliance on human commands, the user interface for the next generation of operating systems is of utmost importance. The use of Command Line Interfaces (CLIs) in the past has been replaced by Graphical User Interfaces (GUIs) in modern systems. Along with the Graphical User Interface, the next generation of operating systems may potentially include Audio User Interfaces (AUIs) and User Gesture Interfaces (UGIs). To improve the accuracy of input identification, it is possible to integrate sensors with visual and auditory interfaces into the human-machine interface in addition to the standard interface systems. In order to communicate with the local or worldwide community or input/output devices, machine-machine communication and machine- input/output communication are also expected to progress. As they allow the system to adapt to user activities even when faced with incomplete, fabricated, or loud inputs, artificial intelligence (AI) components are projected to become an essential component of operating systems. Through auditory or visual interfaces, an intelligence component could communicate with the user and then carry out the necessary operations on their behalf. Furthermore, the creation of next-generation operating systems depends heavily on sophisticated hardware technologies. This technology may encompass an Artificially Intelligent System on a Chip (AI SOC) or an AI-based sensor controller combined with a multi-task processor to increase the computation speed.

Discussion And Some Findings

As seen from the paper, the main essence of OS is to serve as an interface between the hardware devices and the user. The evolution of computer OS started from batch operating systems to the new trends of OS. OS is developed for hardware architecture to power the device. The authors think that developing hardware with its OS is very expensive because of the rapid changes in the modern computer environment. Devices may become obsolete in a few years and discarded leading to e-waste which will harm the environment. The authors see the need to develop a universal OS for device categories to avoid the re-occurring cost of building OS all the time for new devices or technologies. There are a number of challenges associated with OS and these pose threats to the users and their devices. The major challenge of OS is the issue of security. A colossal cost is incurred to find a solution to secure the system. The Evolution trend of OS makes it prone to security attacks since changes are made to the hardware and device drivers all the time. To curtail this, there should be s standardized security framework for every hardware device in such a way that when this requirement is not met, the product is not passed for use by industry players. Also, the new trends of OS as found in the literature suggest that any emerging technology comes out with its OS. It is therefore seen that the OS largely depend on the Technology that is trending in the world. The current trends of OS in the world are IoT OS which is developed to power IoT devices, cloud OS which is designed to power hardware devices at cloud data centers, AI-OS is designed to power AI hardware devices, blockchain OS, Containerization OS, and Quantum OS which runs on different types of quantum computer hardware. Every new trend in technology has its own unique hardware devices which will require different OS such as device drivers to make it work. This also comes with a cost component since unique hardware and OS must be developed to implement the new Technology.

Conclusion

An operating system facilitates the execution of tasks and processes on a machine. In this scholarly discourse, we have undertaken an investigation into conventional and modern operating systems. Based on this examination, we put forward the significant constituents of the forthcoming generation of operating systems, commonly referred to as advanced operating systems. While these constituents are either derived from the software interface or the machine interface, the topics of communication and artificial intelligence necessitate further deliberation in the context of communication and artificial intelligence. Cloud operating systems possess the capability to concurrently manage multiple virtual machines. Furthermore, they aid in the administration of the cloud environment and the configuration of the dashboard to fulfil the requirements. Additionally, the consolidation of all essential applications for both business and personal purposes in a single location saves time and enhances productivity. Notably, certain cloud operating systems even incorporate a business solution to assist in the selection of a data-centric strategy and the integration of results-oriented solutions. Before beginning the use of a cloud OS, it is advisable to thoroughly investigate and authenticate the insights due to the distinct features present in each operating system. From the contemporary trends in Operating Systems gleaned from the literature, it is patently evident that each new technology is accompanied by a new operating system to facilitate the functioning of said architectural technology. This has financial implications as novel technologies continue to emerge incessantly. Consequently, this scholarly paper advocates for the design of a universal OS that can encompass various OS architectures while allowing for the future expansion of the new architectural trends in OS. Given the perpetual rise in security alerts, it is imperative to explore a more effective approach for addressing the root causes of vulnerabilities in operating systems. The execution of applications from a highly secure operating system can serve as a frontier in combating many of the prevailing real-world cyber threats. Although it may not be possible to completely eradicate all the perils of the current cyberspace, and the security of individual applications may still be susceptible to their vulnerabilities, the implementation of a secure operating system can help control the damage and impact across multiple applications. The security of the operating system is a major concern, and this scholarly article suggests the adoption of a dual-layer security approach wherein the OS is fortified with security policies and security protocols are embedded in the hardware architecture of the OS to enhance computational speed. The technical implication of this research is that new operating system designs are needed to accommodate the changing operating system trends. For instance, new architectures are needed for containerization in order to manage and coordinate containers. To satisfy the specialized requirements of IoT devices, IoT integration necessitates novel designs. And new architectures are needed for AI to support its workloads. It is also clear that new operating system features will be necessary to keep up with the developments in operating systems. For instance, new functionalities are needed for IoT integration to facilitate data gathering and sharing with IoT devices. Both enhanced security and AI require new features to accommodate AI workloads and defend against evolving threats. Lastly, new technologies and development tools will be needed for the creation of new operating systems and operating system functionalities. For instance, new development tools and technologies are needed for the creation of containerized apps. Additionally, new development tools and technologies are needed to provide operating system functionalities powered by AI.

CITED REFERENCES

1. Casseau E, Dobias P, Sinnen O, Rodrigues GS, Kastensmidt F, Savino A, et al, A. Operating Systems under test: an overview of the significance of the operating system in the resiliency of the computing continuum. 2021 IEEE 39th VLSI Test Symposium (VTS):1–10.
2. Qingquan J, Yinming G, Rui Z, Qiaozhen L. Research on the Evolution Law of -computer Interaction Function in Computer Operating System and Control Mode. 2020 Management Science Informatization and Economic Innovation Development Conference (MSIEID):300–303.
3. Tomilin AN. Viktor Ivannikov’s four generations of operating systems. 2020 Fifth International Conference “History of Computing in the Russia, Former Soviet Union and Council for Mutual Economic Assistance Countries” (SORUCOM):47–48.
4. Goyal S, Chaudhary A, Kumar A. Cloud Operating Systems: Analysis and Problems. 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO):1–5.
5. Chen L, Zhang Y, Tian B, Ai Y, Cao D, Wang FY. Parallel Driving OS: A Ubiquitous Operating System for Autonomous Driving in CPSS. IEEE Transactions on Intelligent Vehicles. 2022 ;7(4): 886–895.
6. Rong X. Design and Implementation of Operating System in Distributed Computer System Based on Virtual Machine. International Conference on Advance in Ambient Computing and Intelligence (ICAACI). 2020;94–97.
7. Zeng Y, Wang R, Cheng Y, Xie G. Design of Test Framework Based on Lightweight Operating System. International Conference on Computer Network, Electronic and Automation (ICCNEA). 2021;96–100.

8. Sha A, Anvesh K, Naidu GN, Dasari, R. Recent Trends and Opportunities in Domain Specific Operating Systems. 2022 3rd International Conference for Emerging Technology (INCET):1–5.

9. Borgohain T, Kumar U, Sanyal S. Survey of Operating Systems for the IoT Environment. International Journal of Advanced Networking and Applications. 2015;6(5):2479-2483.

10. Musaddiq A, Zikria Y, Hahm O, Yu H, Bashir A, Kim SW. A Survey on Resource Management in IoT Operating Systems. 2018;1-1. 11. Zikria YB, Kim SW, Hahm O, Afzal MK, Aalsalem MY. Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution. Sensors. 2019;19(8):1793.

11. Gaur P, Tahiliani M. Operating Systems for IoT Devices: A Critical Survey. 2015:33-36.

12. Pianese F, Bosch P, Duminuco A, Janssens N, Stathopoulos T, Steiner M. Toward a cloud operating system. IEEE IFIP Network Operations and Management Symposium Workshops. 2010;335-342.

13. Xiong G, Ji T, Zhang X, Zhu F & Liu W. Cloud operating system for industrial application. 2015:43-48.

14. International Conference on Service Operations and Logistics, And Informatics (SOLI) (pp.43-48). IEEE.

15. Garcia S. Cloud Operating System (cloud OS); 2021. Wasabi. Accessed 6th September 2023. Available: https://wasabi.com/glossary/cloud-operatingsystem-cloud-os-definition

16. Oza N, Münch J, Garbajosa J, Yague A, Ortega E. Identifying Potential Risks and Benefits of Using Cloud in Distributed Software Development. 2013 ;7983. DOI:10.1007/978-3-642-39259-7_19.

17. Sujeet KS, Ali HA, Srikrishna MG, Mohammed HA. Predicting motivators of cloud computing adoption: A developing country perspective. Computers in Human Behavior. 2016; 62:61-69.

18. Marston S, Li Z, Bandyopadhyay S, Zhang J, Ghalsasi A. Cloud computing — The business perspective. Decision Support Systems. 2011; 51:176-189.

19. Yang H, Tate M. A Descriptive Literature Review and Classification of Cloud Computing Research. Communications of 2012; 31:35-60.

20. Okechukwu OG. Security Evaluation of Google Chrome Operating System. IOSR Journal of Computer Engineering. 2014; 16(6):64–67.