Sri Lanka Institute of Information Technology

Assignment 1

Group 02

Evaluating the features of various OS

Secure Operating Systems – IE2032

B.Sc. (Hons) in Information Technology – Cyber Security

Group Details

Group Number: 02

Project Title: Evaluating the features of various OS.

	Student ID	Student Name	Email	Contact Number
1	IT22357762	Dewmini P.L.T	IT22357762@my.sliit.lk	0741688488
2	IT22315496	Anuradha D.P.G.C	IT22315496@my.sliit.lk	0713952609
3	IT22599186	Yashodini Jayasinghe	IT22599186@my.sliit.lk	0767096940
4	IT22002174	Dineth T.H.V	IT22002174@my.sliit.lk	0766604096

Term of references

A report submitted in fulfilment of the requirement for the module IE2032, Sri Lanka Institute of Information Technology

Contents

1 Abstract

The operating system assumes a crucial role in contemporary computer systems. The interface facilitates the interaction between the user and the computer system, including its physical components. Simultaneously, it assumes responsibility for the administration of processes, memory, files, input/output operations, and security. This research examines the role of resource management and extension capabilities on client, server, and mobile devices, with a focus on their utilization across various operating systems.

2 Introduction

Operating system is the fundamental component of a computer system which provides the interaction between the user and the hardware components which is underlying. To provide a seamless and effective user experience it provides a variety of services. It plays a crucial role as resource manager and as an extended machine.

OS serves as a resource manager allocating and deallocating the resources like as CPU, memory, input/output devices and other hardware components across multiple programs and processes. It ensures not to monopolize all available resources for one process.

The operating system (OS) serves as an augmented apparatus by providing a level of abstraction that conceals the complexities of the underlying technology, thereby establishing a uniform and user-friendly interface for individuals. The incorporation of the extended machine notion is crucial for facilitating efficient and successful user-computer interaction, irrespective of users' technical proficiency or knowledge with the underlying technology.

3 Resource management and Extended machine.

3.1 Resource management

"Resource management is the process of efficiently allocating and deallocating system resources, such as CPU time, memory, disk space, and I/O devices, to various processes and applications running on the system."[1]

Resource management can be considered as a crucial in operating system as it directly impacts performance, stability, and efficiency of the computer. This allows to manage resources effectively and efficiently preventing wastage and allowing to share resources. Optimal resource utilization is enabled within this and improve performance allocating resources based on priorities and requirements. Further, it manages the virtual memory which allows processes to use more than physical memory available by swapping data between RAM and disk storage.

Here are some specific tasks that the OS performs as a resource manager.

- Process management
- Memory management
- File management
- I/O management
- Security

Resource management allows multiple processes to run simultaneously by allocating CPU time memory to different processes by preventing deadlocks. Managing physical memory and virtual memory separately allocating memory to different processes, prevent memory fragmentation is also facilitated. Nevertheless, it manages I/O devices by buffering data and handling interruptions.

To improve that of service, there are several techniques that are followed by the OS.

- Following **scheduling algorithms** allows to decide what process to run next and the duration that should it run. This helps all processes to get fair CPU time without prioritizing any of process.
- **Paging** is used to manage virtual memory allowing OS to store data in memory than with physical memory alone.
- **Caching** is used to store frequently accessed data in memory to improve the performance of applications.
- OS use **buffering** that allows to store data that is being transferred between memory and I/O devices.

3.2 As an extended machine

An extended machine refers to a computer system that offers a level of abstraction, serving as an intermediary between application programs and the underlying hardware. The presence of this layer of abstraction serves to conceal the intricate details of the hardware, hence facilitating the task of programmers in developing application programs. In addition, the enlarged machine also offers. There exists a diverse range of system calls that facilitate the interaction between application programs and the underlying hardware components.

This was a concept that was first proposed by Gene Amdahl back in 1964. According to Amdahl, it was hard to write useful and dependable application programs because computer hardware was so complicated. He came up with a new plan that would hide the hardware's complexity and make it easier for programmers to use.

All current operating systems now use the idea of an "extended machine." Memory management, file management, and I/O management are just a few of the services that the OS offers to application programs. It is easier for programmers to write application programs with these services because they hide the complexity of the hardware underneath.

Having extended machine allows programmers to write applications as it abstracts away the complexity of the underlying hardware. Application programs that are developed for an expanded machine exhibit greater portability across various hardware platforms. The reason for this is that the expanded machine offers a uniform interface to application programs, irrespective of the underlying hardware. Taking advantage of the underlying hardware features, the extended machine can optimize the execution of application programs.

Here are some examples that can be considered as extended machines.

- Operating systems – Windows, Linux, macOS
- Virtual machines – VMware , VirtualBox
- Containers – Docker
- Cloud computing platform – Microsoft Azure , Amazon Web Service

4 How Client, Server and Mobile OS works as Resource manager and extended machine.

4.1 Support many devices simultaneously.

4.1.1 Client OS – Windows

The "support many devices simultaneously" feature refers to a system's ability to handle multiple devices or links at the same time without experiencing major performance drops or interruptions. In this case, the system is a client computer running the Windows operating system. Let's break down this feature in more detail:

- Concurrency
- Performance
- Resource Management
- User Experience
- Security
- Scalability

1. **Concurrency**: This characteristic suggests that the system has the ability to simultaneously manage and communicate with several devices or connections. All the devices kinds, including physical components, networked devices, Bluetooth devices, and USB peripherals, might fall under this category.

2. **Performance**: The system should be able to successfully spread system resources (CPU, memory, and network speed) to each device or link so that they can all work without any noticeable pauses or slowdowns. This is especially important when a lot of gadgets or programs are running at the same time.

3. **Resource Management**: Effective resource management is required by the system in order to support several devices at once. In order to ensure equitable and effective resource utilization, resources should be allocated dynamically according to the requirements of each device or connection.

4. **User Experience**: Supporting several devices at once should be simple to use and straightforward for users. Users should be able to interact with connected devices effectively and get clear information about them through the user interface.

5. **Security**: When the system supports several devices, security considerations must also be considered. It needs to be equipped with defenses against possible security risks and illegal access from any linked device.

6. **Scalability**: Scaling to accommodate several devices at once should be possible. Stated differently, it needs to possess the capability to manage a growing quantity of devices or connections without experiencing a noticeably diminished level of efficiency.

When the Windows techniques are concerned, It uses multiple to support the above-mentioned feature.

- Device drivers and manager
- Virtualization
- Plug and play (PnP)
- Interrupt Request Management (IRQ)
- I/O port and Memory management
- Multitasking and multithreading
- Device specific APIs

Multi-tasking refers to the capacity of an operating system to execute numerous processes concurrently. It's organized jobs specially code and data allowing CPU to execute always. This objective is accomplished by means of the distribution of processor resources and memory, so enabling each task to be allocated an appropriate amount of computational power and memory capacity. Through effective resource management, multitasking guarantees that all tasks are given sufficient attention and may be executed without significant delays or conflicts.

The Device Manager is an integral element of the Windows operating system, functioning as a centralized tool for the management of hardware devices that are connected to a computer. The software offers users a visual interface through which they may access, modify, and address issues related to hardware components.

The management of Interrupt Requests (IRQs) is a crucial component of the Windows operating system's handling of hardware devices, facilitating their efficient interaction with the Central Processing Unit (CPU) while minimizing conflicts. When a hardware component, such as a network card or a USB controller, necessitates communication with the central processing unit (CPU), it initiates an interrupt request.

When a hardware device is connected to a hardware device, windows is figuring out available IRQ to that device. Windows operating system has the capability to enable the distribution of Interrupt Requests (IRQs) among various devices. This implies that many devices have the potential to be allocated the same IRQ number, provided that they do not necessitate simultaneous servicing. As an illustration, it is possible for a keyboard and mouse to share the same Interrupt Request (IRQ) due to their infrequent simultaneous requirement of system attention.

Virtualization is a technological advancement that enables the concurrent execution of multiple operating systems on a single computing device. This functionality becomes advantageous in facilitating the concurrent operation of many versions of the Windows operating system on a single computer, as well as enabling the coexistence of Windows with other operating systems such as Linux or macOS. Windows has a diverse range of virtualization capabilities, including prominent technologies like Hyper-V and Windows Sandbox.

4.1.2 Server OS – Linux

Linux is out of the box and built to handle many devices simultaneously. It has serval techniques and technologies for doing that. In Linux, it is built on a modular kernel architecture. This allows device drivers to be loaded and unloaded dynamically. Because of that, the kernel can support many hardware devices and when the hardware is connected Linux can load drivers quickly. The open-source capability of Linux has a lot of community-driven devices.

So, it makes Linux support a wider range of hardware devices.

Supporting Multiprocessing and Multi-threading is a great feature of Linux. It takes on preemptive multitasking, allowing multiple processes to run simultaneously. Also, Linux supports multi-threading. because of that it can execute Multiple threats concurrently. Handling input. Output operating simultaneously for applications is an example of this.

In file system management Linux uses a virtual file system known as VFS. Applications and Users can interact with directories, but the users do not need how the files are stored. That is how the storage devices are being managed and supported by the Linux as a server OS.

For better support for the many devices simultaneously the OS need to have better resource management . When managing resources, the OS follows stranded procedures like control groups and process and memory Management. From group the in Linux the admins can manage the allocation of resource usage . With that each device can have fair resource usage. when considering about Process and Memory Management Linux server has

so many programs which will be converted as processes . So, OS is Handling process and maintain the stability as much as possible. When handling process OS create new processes and assigned using fork() ,exc() system calls. Also scheduling processes different process scheduling algorithms like priority. In memory management topic Linux is using virtual memory creations, allocating and de allocating memory spaces , swap spacing, memory protection like read-only memory pages features better achieve the management.

4.1.3 Mobile OS – Android

In the context of a mobile operating system such as Android, the term "Support many devices simultaneously" refers to the system's capability to effectively manage and execute several apps and processes on different Android-powered devices simultaneously.

Multitasking: Android users have the ability to run numerous applications at once. Apps may be switched between with ease, and many of them allow you to multitask by continuing to run in the background. For instance, you may get alerts while playing a game or listen to music while surfing the internet.

Multi-window Support: On devices with bigger displays, Android's multi-window capability lets you run numerous programs side by side. This feature allows users to work on many projects at once, which increases productivity.

Background Processes: Android effectively controls background processes, making sure that background-running applications don't use up too many system resources. This preserves both the device's general speed and battery life.

Device Compatibility: Android is made to work with a myriad of gadgets, ranging from smart watches and TVs to smartphones and tablets, and even certain appliances. Because of this compatibility, Android must be able to manage a range of screen sizes and hardware setups.

User Profiles: Two or more user profiles may be supported by Android on a single device. This is very helpful in scenarios when many users need to have separate customized experiences on the same hardware, such as shared devices.

Virtualization: Several Android instances may be operated concurrently on some Android devices thanks to virtualization technology. This is often seen in business settings, when a single device may have many distinct Android environments installed on it for various uses.

Resource Management: To effectively distribute CPU, memory, and other hardware resources across various applications and processes, Android uses resource management techniques. This guarantees that even with several programs open, the system will always be reliable and responsive.

Compatibility with Diverse Hardware: Because of Android's modular design, the operating system may operate concurrently on a large number of different devices and be tailored to a broad range of hardware configurations.

Android's "Support many devices simultaneously" feature is essential to the platform's broad acceptance and use across a variety of devices. It makes sure Android can maximize speed and battery life while effectively managing resources, running numerous applications at once, and offering a consistent user experience across different devices.

4.2 Allocate resources to different users.

4.2.1 Client OS – Windows

In a client system running Windows, the "Allocate resources to different users" feature describes the capacity to control and divide system resources, such CPU, memory, and network bandwidth, among many users who are using the same system at the same time.

- User Isolation
- Resource Management
- User Authentication
- Session Management
- Session Management
- Fairness
- Security
- Logging and Monitoring

1. **User Isolation**: Windows-based client systems are often used in multi-user settings, including home computers or shared workstations in offices. The system may establish separate user profiles and assign resources to each user on an individual basis thanks to this functionality. It makes sure that no user's actions disrupt or negatively affect the efficiency of another user's sessions.

2. **Resource Management**: The resource allocation process has to be intelligently managed and prioritized by the system. It should equitably divide up memory, CPU time, and other system resources between the running user sessions. This implies that other users' performance shouldn't be hampered if one user is using resource-intensive programs.

3. **User Authentication**: The system needs user authentication procedures in order to distribute resources to various users in an efficient manner. Every user has to sign in using their unique credentials, and the system need to distribute resources according to the rights and privileges linked to their user accounts.

4. **Session Management**: Multiple user sessions should be supported simultaneously by the system. Individuals have the ability to transition between their sessions, and the system need to effectively handle these transitions while guaranteeing that every user's information and programs stay distinct and safe.

5**. Fairness**: Fair and equitable resource distribution is necessary to prevent any one user from monopolizing system resources at the expense of other users. This avoids situations in which the actions of one user make the experience of other users slow.

6. **Security**: Giving tools to users, security needs to be thought about. Only the information and tools that users are allowed to use should be available to them. Access rules, rights, and security procedures must be in place to protect user data and the stability of the system.

7. **Logging and Monitoring**: Every user should have access to tools for tracking their resource utilization inside the system. This may assist administrators in identifying and resolving a resource allocation-related performance bottlenecks or security concerns.

4.2.2 Server OS – Linux

In a server it has so much resource comparing to client computer. Also, the server computer contains so many users. So allocating resources among these different users is very hard and complex. If the server OS cannot handle the resource allocating better the server fails to contain its reliability and stability.

Scheduling process maintain cgroups for resource contorting , having user and group for management of permissions for usage of different resources , supports disk quotas which allows admit to manages disk space of different users and user groups.

Having kernel parameters is the best option for optimizing the server in Linux. With this it can manage files behaviorist, network behaviorist such as number of connections in a server. Monitoring resources is great option that that handling many users in an environment like server. Because monitoring resource usage can prevent deadlocks from occurring and detect them. In Linux environment there are popular tools like top,htop . btop to monitor the resource allocating and running process with its pid. And their names, users who are running them.

Linux operating system as a server is a better resource allocator and one of best exiting machine . The above fact also gives a great proves it better.

4.2.3 Mobile OS – Android

The ability of a mobile operating system (OS), such as Android, to manage and distribute system resources, like CPU processing power, memory (RAM), and network bandwidth, among several users or applications running on the device is referred to as the "feature of allocating resources to different users." This functionality is crucial for maintaining a seamless and effective user experience while avoiding resource monopolization and performance problems caused by a single user or program.

The following explains how Android's resource allocation functionality operates:

User Profiles: On a single Android smartphone, users may establish several user profiles. Like distinct environments, each user profile has its own collection of programs, preferences, and data. This division aids in resource isolation for various users.

Resource Management: The resource allocation mechanism of Android uses a priority-based strategy. The OS determines which applications or processes to prioritize while they are running concurrently, considering user activity, app significance, and background duties.

CPU and Memory Allocation: The scheduler for Android makes sure that background services and running applications share CPU processing time equally. Additionally, it controls memory allocation, moving data in and out of RAM to make ensuring that programs have the memory resources they need to run well.

Background Process Limitations: To stop applications from using up too many system resources while they are not in the forefront, Android restricts the amount of CPU time and background network use that they may use.

Foreground and Background Tasks: Android can tell the difference between background and foreground processes. Resources are allocated with a greater priority to foreground programs. More severe resource limitations apply to background processes in order to prevent them from degrading user experience.

Resource Monitoring: Android keeps an eye on how many system resources are being used, and it may take appropriate action if a process or app uses too many resources. For instance, the OS may throttle or even kill a program that uses too much CPU or memory in order to free up resources for other processes.

User Control: Selecting which programs to launch or dismiss gives users the option to directly allocate resources. They also have control over permissions unique to each program, which might impact resource access.

Generally speaking, Android's resource allocation function makes sure that the mobile operating system can divide and manage a device's limited resources across many users and apps. This keeps the system stable, guards against crashes, and guarantees a snappy and pleasurable user experience—even while using the device to multitask or run resource-intensive applications.

4.3 The level of security it provides.

4.3.1 Client OS – Windows

In a client system running Windows, the term "The level of security it provides" describes the system's defense against different security risks, vulnerabilities, and unauthorized access. It also refers to the system's capacity to safeguard user data and programs. Over time, Windows' security capabilities have developed to provide strong defense against a variety of attacks. This security feature in a Windows client system is explained as follows:

- User Authentication
- Access Controls
- Firewall
- Antivirus and Antimalware
- Windows Updates
- BitLocker Encryption
- Secure Boot

User Authentication: Windows uses passwords and usernames as well as more sophisticated techniques like hardware tokens and biometrics (such fingerprint or face recognition) to enforce user authentication (e.g., smart cards). This guarantees that the system can only be accessed by authorized users.

Access Controls: Windows controls permissions for files, directories, and system resources using access control methods. Granular access permissions may be defined by administrators to grant or restrict access to files and folders for certain users or groups. This feature makes sure that private information is shielded from unwanted access.

Firewall: A built-in firewall in Windows may be set up to filter both incoming and outgoing network traffic. By doing this, harmful network traffic is blocked and unauthorized network access is prevented from the system.

Antivirus and Antimalware: Windows also comes with its own built-in security solution known as Windows Defender, in addition to being compatible with a variety of third-party antivirus and antimalware programs. These programs are helpful in locating malware, viruses, and other types of malicious software as well as removing them from your computer.

Windows Updates: Microsoft fixes operating system vulnerabilities by releasing security updates and patches on a regular basis. Updates that happen automatically make sure the system is always safe from known security risks.

BitLocker Encryption: BitLocker, a feature of Windows, allows for complete disc encryption. This function encrypts data on the system's hard disc to prevent unauthorized access without the right password or decryption key.

Secure Boot: A feature called Secure Boot aids in preventing unauthorized operating system code and firmware from executing when the machine is booting up. It lowers the possibility of malware infiltrating the boot process by making sure the system boots up with reliable software components.

With a plethora of features and tools intended to protect the system and user data, a Windows client system offers a significant amount of security overall. To guarantee the best degree of security and remain up to date on new threats, users and administrators must regularly maintain and adjust these security elements.

4.3.2 Server OS – Linux

For a server, security is the next important thing. Managing better security prevents the server from unauthorized access and unavailability. Linux has several features that protect servers from breaching data. Linux has a feature to control access of the users that provides restricted permissions for user accounts and files. As an example, admins can set a file or directory read-only or write-only admins can manage who can edit, read, or execute relevant files or programs.

Also, Linux supports multiple authentication methods. Because of that admins can use not only username and password for the server but also smart card digital certificates. These provide an extra layer of protection to the system. Because of that Linux can correctly verify the users before granting access and resources to them.

The User permission and the access contorting structure of the Linux is at very high level. In Linux it follows privilege-based access controlling. The root is highest prevailing in the Linux environment. With these privileges the user can do anything on the system even the kernel of the os. When using Linux as server os administrator need better manage the permission for preventing the privilege escalation. Which allows a regular user to get the root privileges.

And modifies system needed files. Sudo is a method to authenticate the user to get the root privileges. Sudo is very power command in the Linux command system. But it gives a temporary time period.

The open-source nature of the Linux kernel allows any users to see the underground of the operating system. The ends users who are deeply concerned about their privacy much more choose Linux as their server os form this transparency the developers and bug hunter can easily find security vulnerabilities.

Linux provides update from time to time and make the operating system up-to-date.

The kernel of Linux supports major security modules like app armor, SE Linux which provides access control, sand boxing and exploit mitigation.

allows it through the Sudo commands. Auditing and logging in formations are major security points of the operating system. With w and who commands Linux provides the logging information details with the ps commands Linux shows the procuresses that running by or running on the user.

By default, Linux comes with a strong security foundation.

Linux is out of the box and built to handle many devices simultaneously. It has serval techniques and technologies for doing that. In Linux, it is built on a modular kernel.

Architecture. This allows device drivers to be loaded and unloaded dynamically. Because of that, the kernel can support many hardware devices and when the hardware is connected Linux can load drivers quickly. The open-source capability of Linux has a lot of community-driven devices.

So, it makes Linux support a wider range of hardware devices.

Supporting Multiprocessing and Multi-threading is a great feature of Linux. It takes on preemptive multitasking, allowing multiple processes to run simultaneously. Also, Linux supports multi-threading. because of that it can execute Multiple threats concurrently. Handling input. Output operating simultaneously for applications is an example of this.

In file system management Linux uses a virtual file system known as VFS. Applications and Users can interact with directories, but the users do not need how the files are stored. That is how the storage devices are being managed and supported by Linux as a server os.

For better support for the many devices simultaneously the os need to have better resource management . When managing resources, the os follows stranded procedures like control groups and process and memory Management. From cgroup the in Linux the admins can manage the allocation of resource usage . With that each device can have fair resource usage. when considering about Process and Memory Management Linux server has

so many programs which will be converted as processes . So, OS is Handling process and maintain the stability as much as possible. When handling process os create new processes and assigned using fork() ,exc() system calls. Also scheduling processes different process scheduling algorithms like priority. In memory management topic Linux is using virtual memory creations, allocating and de allocating memory spaces , swap spacing, memory protection like read-only memory pages features better achieve

In a server it has so much resource comparing to client computer. Also, the server computer contains so many users. So allocating resources among these different users are much harder and complex. If the server os cannot handle the resource allocating better the server fails to contain its reliability and stability.

Scheduling process maintain cgroups for resource contorting , having user and group for management of permissions for usage of different resources , supports disk quotas which allows admit to manages disk space of different users and user groups.

Having kernel parameters is the best option for optimizing the server in Linux. With this it can manage files behaviorist, network behaviorist such as number of connections in a server. Monitoring resources is a great option for handling many users in an environment like server. Because monitoring resource usage can prevent deadlocks from occurring and detect them. In Linux environment there are popular tools like top,htop . btop to monitor the resource allocating and running process with its pid. And their names, users who are running them.

Linux operating system as a server is a better resource allocator and one of best exiting machine. The above fact also gives great proof it better.

4.3.3 Mobile OS – Android

An Android mobile operating system's (OS) security level is a crucial factor that directly affects the security and privacy of the user's data as well as the device itself. Google's Android operating system has improved its security measures over time. The Android operating system's security aspects and concerns are explained as follows:

App permissions: On Android, users may provide or withhold certain rights for individual applications. This feature lowers the possibility of unwanted access to sensitive data by ensuring that programs can only access the data and features they need to function.

Regular Security Updates: To address vulnerabilities and exploits, Google publishes security updates and patches on a regular basis. These upgrades are essential for maintaining the OS's security and are sent to devices through Over-the-Air (OTA) updates.

Google Play Protect: Google Play Protect is an integrated security tool that checks applications and the device for possible risks and viruses on Android devices. It also alerts users about potentially dangerous applications before to installation.

Secure Boot procedure: When an Android device boots up, a secure boot procedure is used to ensure that the software and firmware are intact. This lessens the chance of illegal system alterations.

File-Level Encryption: user data saved on the device is safe even if it ends up in the wrong hands thanks to Android's support for file-level encryption. Unauthorized users find it very difficult to access personal data without the right credentials because to encryption.

Biometric Authentication: Android is compatible with biometric authentication techniques, which include face and fingerprint recognition. These techniques provide an additional degree of protection when unlocking devices and authorizing transactions.

Android Enterprise: This version of Android comes with security features designed specifically for business and corporate usage. One such feature is the ability to containerize work-related applications and data to keep them apart from personal data.

Third-Party Security applications: To further improve the security of their smartphone, users may install third-party security applications from reputable developers. Features like firewall defense, malware detection, and anti-phishing techniques could be included in these programs.

All things considered, the security offered by Android OS is constantly rising thanks to Google's dedication to patching holes and adding new security features. On the other hand, user actions that include updating the operating system and applications, using robust authentication techniques, exercising caution when giving app permissions, and downloading apps from unknown sources all contribute to an Android device's true security.

4.4 Hides the implementation details.

4.4.1 Client OS – Windows

In a client system running Windows, the term "Hides the implementation details" describes the system's capacity to hide or abstract the underlying technological complexity and subtleties from the end users. This is done to make the system intuitive and user-friendly so that users don't have to worry about how the system operates within.

Let's investigate this feature in further depth:

- Simplified User Interface
- Abstraction of Hardware
- Application-Level Abstraction
- Automatic Updates and Maintenance
- Security Abstraction
- Error Handling and Troubleshooting

Simplified User Interface - The most common method for achieving this capability is via a simple and intuitive graphical user interface. It is common knowledge that the complexity of the underlying hardware and operating system is hidden by the user- friendly interfaces of Windows operating systems. Users operate via windows, menus, and icons rather than working with low-level system settings or command lines.

Abstraction of Hardware: Windows systems abstract the specifics of hardware, enabling users to connect a variety of peripherals (such as external drives, printers, keyboards, and mouse) without having to manually setup drivers or comprehend technical requirements. Resource management, driver installation, and hardware identification are usually done in the background by the system.

Application-Level Abstraction: Applications may be used without the user having to understand how they interact with hardware or connect with the operating system. Windows offers an abstraction layer that lets programs operate without change on a variety of hardware setups.

Automatic Updates and Maintenance: A lot of the time, Windows handles maintenance and updates automatically, saving users from having to learn complex things like patch management or system optimization. This contributes to the system's long-term security and functionality.

Security Abstraction: including firewall and antivirus defense, often operate in the background without needing users to actively control them. This is known as **security abstraction. Although many security processes are automated, Windows offers user-friendly interfaces for adjusting security settings as required.

Error Handling and Troubleshooting: When mistakes happen, Windows often includes easy- to-understand error messages along with tools or troubleshooting guides to assist users in resolving problems without the need for advanced technical expertise.

To put it simply, the goal of a Windows client system's "Hides the implementation details" function is to make computers more user centric. In order to lower the learning curve and increase the system's accessibility for a wider variety of users, it abstracts away the underlying technological intricacies and enables users to engage with the system and applications in a simple and intuitive way.

4.4.2 Server OS – Linux

Operating system is the intermediate tor for user and hardware . Os is the major actor dealing with hardware. For that Os have a deep concept .but for the Users perfective they don't need that . So, Os should be made with abstraction. Thus, concepts help users to interact with the Linux as a server without understand hard concepts.

So there have two main abstractions out there

- Hardware abstraction: Linux uses hardware abstraction for hide the hardware device s from application and provides interface for interact with the hardware devices. Because of that developers can easily manage the usage of the hardware components.
- Kernel abstraction: kernel is the intermediary between software and hardware. So, kernel has a way complex interface. Reducing the complexity abstraction is the solution .from this the software gets a simple and easy way to interact with the hardware devices. Use of system call and API are examples.

Files are the main interface for how users interact with the operating system . So, implementation of the file system is a bit complex. Users need to understand the storage media architectures and write code for each, and all devices attach to the system . Because of that Linux abstract the files and storage devices into a interface called VFS. Means virtual File System . this allows users and apps to access files & media without knowing the underline of the system and architectures.

Network is the major thing in a server. For sharing files navigate internet users require networks. But The implementation of the network is same as hard as above. Thanks to the socket programming with java and

python languages make easy the implementation of networking. Most applications use Socket programming to communicate. Because of that, the user doesn't need to make codes for each network and doesn't understand the internal architecture of the network. Use of the network interfaces makes users easier interact with the network. Ethernet and Wi-Fi are major examples for modern network interfaces. having the interface made easy for Developers.

Managing process and memory is the main work of operating system. As a intermediary OS need to understand about usage of ram and CPU . Os need to make not only a one process make CPU busy always. For that CPU need to have a schedule . Linux in hand allows application run process without understanding the underlying CPU scheduling means multitasking mechanisms.

Taking about the memory abstraction Os use virtual memory concept which allows programs to use independent memory address for Physical RAM location. This abstraction simplifies the memory and process management for end users. Having a user interface make os convenient to use but making interface is harder. Operating systems are maintaining two major user interfaces one is GUI and other is CLI the GUI is called graphical user interface is more resource heavily and more user-friendly interface other hand the CLI called Command-line interface is less resource usage and hard to use for ordinary users. Building a GUI for the operating system is very hard and abstract most of the operating system internals. Heavily resource usage of this interface because it is not majorly use in server the side.

API also named as Application Programming Interfaces are the major interface for application to communication with system calls. When developing applications, the developers need to know about system call to provide a function which requires usage of the hardware components and requires kernel level access . But the complex interface of the system calls makes it hard to understand its behavior for application developers. So having an intermediate tor for doing that work makes its easy.

In Linux it provides High level API from Standard Libraries like glib c. These libraries abstract the lower-level system operations. In Linux it allows for Third party APIS. With this Linux can reduce complexity of the system call and abstract complex functions from the end users. As an example, if developers want to build application to web services the developers need to have deep knowledge about web services and its implementation. So, Developers need to write codes for every device out there . but the APIS makes that easy. Thanks to APIS Developers can make programs easy and simple which makes the software cost less.

The abstract usage of Linux makes it easy to use. With the abstraction Linux is easy to maintain , develops as a server side os.

4.4.3 Mobile OS – Android

You brought up a feature that "Hides the implementation details," which is a basic feature of the Android OS and, more generally, of mobile operating systems in general. Let's examine this feature's meaning in relation to Android:

Abstraction Layer: Like other mobile operating systems, Android offers developers a high degree of abstraction. Application developers are not given access to low-level hardware and system data,

such as the CPU, memory, and sensors. Several software layers, including the kernel, libraries, and APIs (Application Programming Interfaces), work together to provide this abstraction.

API Abstraction: To communicate with the hardware and system services, developers may utilize a set of APIs that Android offers. The underlying OS and hardware functionality is abstracted by these APIs. For instance, a developer may utilize the Camera API to access and manage the camera on a smartphone rather than figuring out the intricacies of the camera hardware.

Uniform Interface: Android provides application developers with a standard interface. This implies that programmers don't have to worry about the nuances of various Android device models and hardware variants while writing code for Android applications. The operating system manages the hardware-specific information, so applications can function reliably on different Android devices.

Hardware Independence: Hardware independence is made possible by Android's abstraction of hardware information. Android applications are compatible with a broad variety of devices and hardware setups. A wide range of Android smartphones, tablets, wearables, and other gadgets are made possible by this capability.

Simplifies Development: Android makes development easier by concealing implementation details. Instead of becoming mired down in low-level hardware programming, developers may concentrate on creating cutting-edge and intuitive apps. App development is now more efficient and accessible because to this abstraction.

To put it simply, Android's feature "Hiding the implementation details" refers to its capacity to abstract away the intricacies of system-level and hardware-level processes, giving app developers a clear and consistent interface. This abstraction makes it easier to create apps, encourages hardware independence, and guarantees a uniform user experience on different Android devices.

5 References

[1]“Computing Science Courses | Jacksonville University in Jacksonville, Fla.,” *Computing Science Courses | Jacksonville University in Jacksonville, Fla.* Available: https://www.ju.edu/computingscience/course-descriptions.php?cv=1.

[2] "Operating Systems," ResearchGate, 2017. [Online]. Available: https://www.researchgate.net/publication/313309097_Operating_Systems.

[3] R. Arpaci-Dusseau and A. Arpaci-Dusseau, "Operating Systems: Three Easy Pieces," 2023.

[4] T. Anderson and M. Dahlin, "Operating Systems: Principles and Practice 2nd Edition," 2nd ed.