OPERATING SYSTEM

Lecture 12

Operating System Configurations: Windows, Linux, Android

Dr Victoria Mukami

INTRODUCTION

This lecture is the final in a series of 12. We do a review of three operating systems. The operating systems are Windows, Linux, and Android. We will review memory management, processor management, input/output management, file management and security management.

Learning objectives

By the end of this topic, you should be able to:

1. Understand the history of the Windows, Linux, and Android operating systems
2. Describe the various types of processors, memory, I/O, and file management for the three operating systems
3. Understand the type of security management for each of the three operating systems

OVERVIEW

We have come a long way since the first lecture. We have done an overview of various operating system terminologies including memory management and processor management. We also looked at the various types of I/O management, file management and security practices. This was done in a broad overview that was not operating system specific. We now concentrate on specific operating systems: Windows, Linux, and Android. We will look at each and begin with a history of each proceeded by a review of memory management, processor management, I/O management, file management and finally, security management.

WINDOWS OPERATING SYSTEM

The Windows operating system is one of the most popular operating systems by Microsoft. It features a graphical user interface (GUI) that is appealing to users and has an easy-to-use system. Most computers bought today will come pre-installed with a Windows operating system. The Windows operating system is versatile as it has been developed for computers of all sizes from desktops to servers, from laptops to mobile phones to embedded computers. We start with a brief history of the operating system.

History

The Windows O/S started with a command-based system known as MS-DOS. The first operating system was truly developed in 1995 with the inception of the Windows 95 O/S. Previous O/S featured a GUI that was placed on top of the MS-DOS system, making the MS-DOS the O/S [1]. This had several disadvantages including a lack of proper security or even worse, no multitasking. With the release of Win-95, a new era began, and Windows worked toward the current operating system we have today Win-11.

The Microsoft team specifically, the Windows team have five main design goals during the development of the O/S. These are extensibility, portability, reliability, compatibility, and performance [1].

Extensibility: this is the ability of any O/S to support new types of peripheral devices or new software technologies [1]. With the current world we live in, the developments happening within the hardware and software fields are immense and as such an O/S must adapt to changes in the same.

Portability: this is the ability of an O/S to operate on different platforms and more so to work with different processors [1]. Two main companies manufacture over 90% of the processors globally i.e., Intel and AMD. The Windows O/S is expected to work and integrate with any processor regardless of the version. With all the advancements especially in embedded hardware devices, the operating system needs to be able to work with a device regardless of the manufacturer.

Compatibility: this is the ability of an O/S to be able to read and write and execute instructions written in any previous O/S. That is, Windows 11 will be able to support programs written for Windows 10, 8 and so on.

Performance: this is the ability of an O/S to respond quickly to CPU-bound applications [1]. Keeping in mind how many applications are installed in an O/S, how many run concurrently and its impact on performance, it is key to have the O/S maintain performance.

Memory Management

As learnt previously, memory management focuses on how jobs and processes are allocated memory as a resource. Windows features the use of virtual memory (VM) when physical memory becomes full [1]. Mainly within the Windows environment, the O/S resides in high virtual memory while user data and code reside in low virtual memory [1]. The virtual memory manager within the Windows system relies on address space management and paging techniques. The upper part of the VM is only accessible by the O/S kernel. Paging policies within the VM system will dictate how and when paging will be done.

Processor Management

When we discussed the processor management schemes, we indicated that they were either non-preemptive or pre-emptive. Windows supports pre-emptive algorithms, multitasking and multithreading. Since windows support different types of processors and multiprocessors. a process can have as many threads as CPUs available [1].

I/O Management

This section devotes to I/O management. Accommodation is made to support all types of peripheral devices from input to output, storage, and communication devices. The Windows O/S allows for the recognition of different devices by installing what is known as drives that make the device work appropriately with the system. Windows O/S provides a device-independent model for I/O services [1].

File Management

While Windows O/S uses the NTFS (New Technology File System), it can support any file system available. File management with Windows supports long file names that include spaces and special characters [1]. The file system converts long names to standard file names to allow for compatibility with older Windows systems.

Security Management

Windows is one of the more popular O/S. It has a huge target for various intruders and attacks. Each Windows O/S will feature common security features such as user accounts that include biometric login options, and a memory protection system that wipes clean the memory before allowing for a switch of users. Device access follows

with a separation of user accounts and various levels of permission, as well as file security for different users.

LINUX OPERATING SYSTEM

Linux is based on the UNIX system. Linux differs from UNIX based on various features. Additionally, Linux is an open-source system that allows any developer to create their distribution also known as a distro. Linux is generally free, however, there are paid versions of Linux that offer advanced features.

History

Linux was a system developed by Linus Torvalds [1]. The O/S was first created to run on small desktop machines and had similar functions found within expensive commercial O/S [1]. The first Linux O/S was command-based like the Windows O/S. Newer Linux versions with a GUI have contributed to its widespread acceptance. The most common distribution of Linux is Ubuntu and more recently Kali Linux. RedHat is one of the commercial versions of Linux.

Linux is designed around three design goals: modularity, simplicity, and portability [1].

Modularity: means that the Linux Kernel is independent of the graphical interface. The system does not work as one unit like in the case of Windows.

Simplicity: this is the ability of the operating system to have simple code that is readable especially since the O/S is developed by numerous people.

Portability: This is the ability to work with numerous hardware devices like the Windows O/S.

Memory Management

Based on memory allocation, Linux will allocate a percentage of memory to the kernel. The rest is shared between the user's code, data, and any shared library data [1]. Linux uses the Least Recently Used algorithm to release pages especially when the Kernel requires pages. Page tables are used to manage the list of free and busy pages. Linux also uses virtual memory with demand paging implementing it. Linux also sometimes uses a modified version of the Least Frequently Used policy [1].

Processor Management

Process management within Linux uses several tables to coordinate the execution of processes. Each process within Linux has a descriptor that contains about 70 fields that describe various process attributes. These descriptors will be in ready or in execution state and contain fields like the next run and previously run [1]. Linux uses wait queues when processes need to synchronize with each other. To solve problems of mutual exclusion one of the conditions of deadlock, semaphores are used. Processes to be executed are normally selected based on a predefined criterion. FIFO in collaboration with priority scheduling is two of the algorithms heavily used within Linux.

I/O Management

Linux is device-independent [1]. This means that it ensures that portability between systems is improved. Like Windows, Linux uses device drivers that ensure data is transmitted between memory and the device [1]. Each device within Linux is identified by using a major device number and a minor device number like UNIX. The major device number is used to access the appropriate code for specific devices while the minor device number is used to access a specific device within similar devices [1]. Devices within Linux are allocated and deallocated using open and release respectively. Devices are either characterized as a character, block, or network. Character devices are accessed by a stream of bytes [1]. These include terminals, communication ports, monitors etc. Block devices are those that contain files that can be accessed directly such as the hard disk, solid state drive, optical drives etc. Network devices are those that send and receive packets of data or information such as routers, ethernet cards etc.

File Management

Linux files are arranged in an inverted tree structure [1]. The highest level is called the root with paths from the roots called branches and the leaves representing the individual files. File names within Linux are case sensitive where a name like a FileName, filename and FILENAME would be seen as three different files. We discussed the characters allowed for the file names during Lecture 10.

ANDROID OPERATING SYSTEM

The Android operating system was designed to run on mobile devices such as smartphones, wearable devices, and tablets. The Android 0/S was built on a Linux Foundation and has a very customizable user interface. Android is open-source software, however, only the key elements of its source code are shared [1].

History

The Android O/S was first developed by Andy Rubin before it was purchased by Google in 2005 [1]. When Google purchased it, the operating system was widely installed across millions of devices. Various versions of Android exist with the most recent one being Android 13 named Tiramisu. Whenever a new version of Android is released, the source code is released to manufacturers of various devices and software developers to allow for app installation and customized software. The design goals for android include enchanting me, simplifying my life and making me amazing. These design guides work to ensure that the applications used in Android are visually pleasing, easy to understand and easy to use.

Memory Management

Remember that Android is developed based on Linux O/S. Memory has a division between the top level and bottom level within the kernel. The top level will hold various applications such as the calendar, and browser while the bottom level will hold the Linux kernel that will hold various device drivers. Applications developed for Android are designed to use resources when require and to use minimal resources when dormant [1]. This means that if an App is open, it resides in memory even when it is hidden or not displayed. Android uses the LRU algorithm to monitor apps residing in memory and will kill apps that have not been used in a while especially when memory becomes scarce [1]. This means that users do not have to force close applications like they would on Windows.

Processor Management

There are four main objects that each app must have. These include manifest, activity, task and intent [1]. Manifest: contains all the information required before the app can begin processing. This file also contains all the permissions that an app requires. Activity: defines the interface screen that users interact with. Task: a collection of

activities that the users interact with. Intent: used to signal from one app to another especially when it needs to cooperate to accomplish something. Activities go through specific lifecycles and have similar states such as ready, running, waiting and finished state.

I/O Management

Android works on very many versatile devices therefore it is necessary for it to be accommodated by all [1]. Applications for Android are developed by quite many independent developers and in this regard must make sure that their application can run on a variety of devices from smartphones to wearable devices, smart TVs etc. Android ensures that it matches each app with the right device. Features like GPS, Bluetooth and others are normally turned off to ensure batteries last.

File Management

The same limitations and characteristics that are found within Linux are shared with Android. Any routine file control within Android is managed by Linux at the Kernel level. Each App has a User ID that allows it to manage and alter its files [1].

Security Management

The Android system has a multitiered security structure that protects a user's data, the system resources and application isolation. Android provides user-defined permissions as the person installing applications needs to give specific access to resources per application. The device security features passwords and biometrics.

SUMMARY

We have reviewed three operating systems. The operating systems were Windows, Linux, and Android. We started with a history for each of the operating systems. We then reviewed memory management, processor management, I/O management, file security and security management for the three operating systems.

DISCUSSION TOPIC

We have so far looked at the various management features used by the operating systems. Do a review of Ubuntu Linux and Kali Linux and find out what the differences are between the two in file management and security.

REFERENCES

[1] McHoes, A., & Flynn, I., Understanding Operating Systems. Boston: Cengage Learning, 2018

[2] Stallings, W., Operating Systems: Internals and Design Principles. Harlow: Pearson Education Limited, 2018.