

Operating Systems

Windows 10 S Case Study



Prepared by:

Simon Owens

Kunal Mukherjee

Abdulaziz AlShabibi

Date: April 20, 2018

Course Number: CS-470

Table of Contents

1. Introduction/History	2
2. Operating System Analysis	4
2.1. Processor Scheduling	4
2.2. Memory Management	5
2.3. File Management	8
2.4. I/O Scheduling	10
2.5. Power Management	10
2.6. Security	12
2.7. Application Programming Interface (API)	14
3. Unique Characteristics	15
4. Opinions on Usage	16
4.1. Simon Owens	16
4.2. Kunal Mukherjee	16
4.3. Abdulaziz AlShabibi	17
5. Conclusion	17
Appendix A: Sources and References	18

1. Introduction/History

The traditional definition of operating system is a system software that manages software resources like process scheduling, hardware resources like memory management, as well as provide common services (like daemons[1]) for computer programs. Recent years saw the growth of hand held devices (smart phones, tablet) as well as gaming devices (Xbox One). The issue of this advancement was that now, we needed specific operating system for specific platforms. For example, desktop OS for Microsoft was Windows 10, Xbox OS for Xbox One, Windows 10 mobile for handheld devices like Nokia Lumina. These Microsoft OS's were fundamentally different from one another and the consequences of that was the apps were not cross platform compatible, but they had one thing in common is that they all ran OS which are derivatives of Microsoft Windows 10. So, Microsoft decided to create a platform on top of its latest operating system, Windows 10, where is a homogeneous application architecture that will make is possible for the apps to be cross-platform compatible.

The platform, Windows UWP (Universal Windows Platform) is a platform homogeneous application architecture created by Microsoft. Figure 1 shows a comprehensive picture of what services are easily implemented in UWP and what devices support it. It provides a common application development and execution platform on every device, also it was first introduced in Windows 10 [2]. The purpose of this software platform is to help develop universal apps that run on different devices which run on Windows 10. Windows UWP is not a consumer related term, but rather a developer related term. For example, a consumer and would buy a laptop that runs windows 10 and not a laptop that UWP implemented in it. But, a developer wanting to create an app on an OS that has platform-homogeneous application architecture, so that modification is not needed in every platform for the app is to run on.



Figure 1: Microsoft UWP supported devices and the services it provide [3]

2. Operating System Analysis

2.1. Processor Scheduling

An executing program is called a process. A thread is the part of the process that can be scheduled for execution. A process spends certain amount of time executing instructions and making some I/O request [3]. Windows 10 is equipped with a process scheduler that is responsible for deciding whether the currently running process should continue running and, if not, which process should run next. Windows use a priority based, preemptive scheduling algorithm. The scheduler guarantee that the highest priority process runs first.

The part of the Windows kernel that handles scheduling is called the dispatcher. The dispatcher has a 32-level priority scheme to determine the order of thread execution. There are two classes of priorities: variable class and real-time class[4]. Variable class contains threads having priority from 1 to 15 and real-time class contains priority ranging from 16-31. Within each priority class there is a relative priority as well. Each process is started with a single thread, often called the primary thread, but can create additional threads from any of its threads. Memory manager has a special thread running

that has priority zero. Memory manager has the lowest priority as discussed later in the chapter, the memory manager needs to prefetch pages or fetch pages after hard page fault, and if it needs to wait for another thread to complete its job, the memory manager would be extremely slow in retrieving its pages.

The dispatcher uses a queue for scheduling priority and it goes through the queue from highest to lowest to find the process that is in ready state, if no thread is found the dispatcher execute a special thread called idle [5]. The initial priority of a thread is typically the base priority of the process that the thread belongs to. Table 1 gives a list of priority as well as the different types of thread and the associated priority with them. Windows 10 gives the user an extra layer of control over the processes, by giving the user the ability to configure processor scheduling so that it gives the best performance while using any program or background processes.

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

Table 1: *List of Priority and different types of thread the Windows system encompasses as well as their respective priority [2]*

Windows also support preemptive multitasking, which means it can simultaneously execute multiple thread form multiple processes. To facilitate grouping on processes, the concept of job was

introduced. The job allows group of processes to be managed as a unit. Sometimes, an app can schedule its own thread by using a mechanism called User-mode scheduling (UMS) without invoking the system scheduler [5]. UMS is more efficient than multi-threading for short duration job items that require few system calls.

2.2. Memory Management

Windows memory management is built around the concept of a working Set. A working set in general is the amount of memory that a process requires at a given time [6]. The working set of a process is the set of pages in the virtual address space of the process that are currently resident in physical memory (RAM) [7], or in other words, working set refers to the total physical memory(RAM) being used by the process. The functionality of working set is similar to TLB (translation lookaside buffer) as when a page is referenced, if it is not in the working set it causes a page fault. The usage of working set creates a unique characteristic of the system is that it has two kinds of page fault: hard and soft. There is a page fault handler that tries to resolve the page fault, and if it does, it adds the page to the current working set.

Hard page fault is like the traditional definition of page fault that the page was not found in working set and the page had to be retrieved from physical memory. Soft page fault occurs for several reasons but the overlying concept is that, the page is present in the memory in some form (another working set, stand-alone) because of some actions, so the memory manager does not need to go to the backing store, to retrieve the referenced page:

- The page is in memory but is in the working set of another process
- The page was referenced for the first time by the current process, but the page was referenced by another process earlier, so it is present in a different working set

- The page is in transition. Transition page remain cached in RAM for future use, as it might be referenced again or repurposed. Transition happens for mainly for two reasons:
 - The page has been removed from the working set of all process that were using it or it has not been repurposed. Repurposed means that the page is filled with zeros and given to another process.
 - The page is already present in the memory because of memory manager pre-fetch operation.

The running process has a minimum and maximum working set size that has an effect in paging behavior. The memory manager tries to maintain space for two working set for minimum working set size, so that memory manager pre-fetch operation can be perform. Pre-fetch operation is when the memory manager looks ahead in the instruction and pre-fetches the required pages. The memory manager does not let the working set get larger than the maximum memory working set size as it tries to main certain number of working sets in the memory.

One of the major problems that a memory manager faces is that sometimes the pages in a working set can get extremely large. Therefore, when a page is swapped from a working set due to the max working set size cap, it can slow the processor down if it must bring the page back from pagefile (storage) and more than likely, the page will be referenced again as the current process is still running. So, Microsoft has developed a special compression system, where the working set has a pointer to a modified list where the pages go when it has been trimmed and swapped out. A illustration is provided in figure 2.

Trimming is the procedure by which the memory manager determines which page gets moved to the modified list and it done by frequency analysis[8]. Modified list is a secondary list of page files backing up a list pages currently residing in a working set. The modified list then decides whether to

put a page back to the store and, should it also copy it to another list called the standby list. The page is compressed when it enters the standby list and it also makes the pages available for reuse in other working sets. The compression decreases the page size to about forty percent of its original size. The read from RAM is way faster as the page can be easily referenced from the standby list and read in faster as the page size has been decreased due to compression.

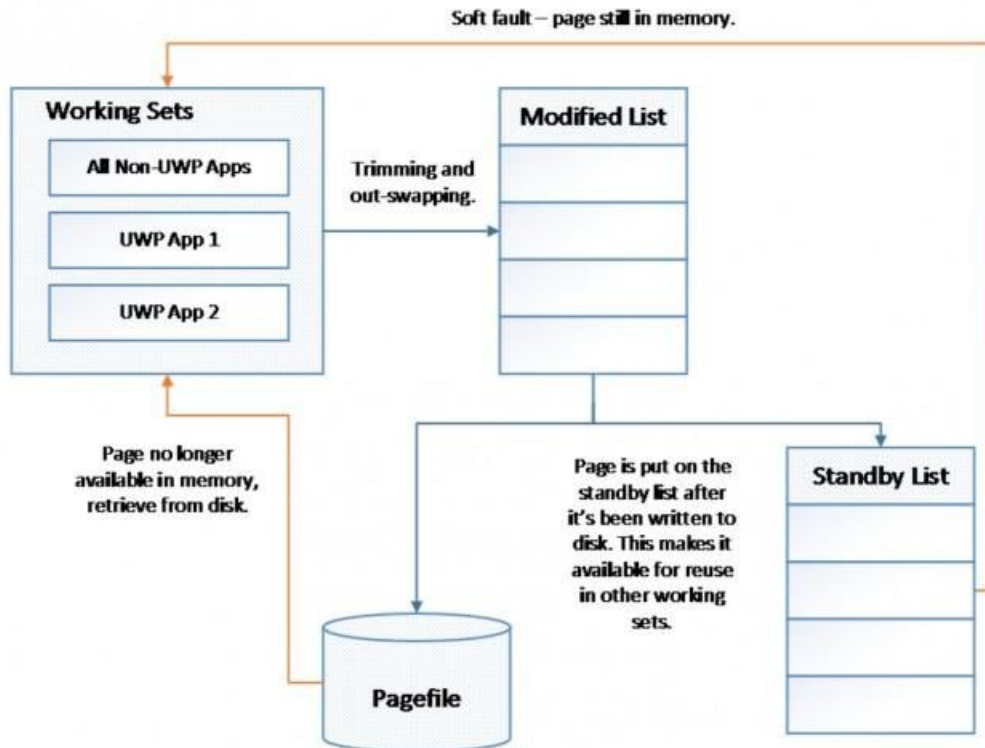


Figure 2: Memory management illustration [6]

The “all or nothing” approach was initially added in Windows 10 by introducing the layer of modified list as well as the standby list between working set and the storage disk or pagefile [8]. This concept of standby and modified list connects to Windows UWP as earlier memory manager scheme was implemented if one of the process of UWP apps was suspended by the process scheduler, the entire page file was swapped out of memory as Microsoft decided not to exhaust it resources on process that are not on the foreground, something that is usually seen in smartphone OS management and surfaces.

The old scheme makes sense for UWP apps running on smartphones where resources are limited but it made the UWP apps running on desktop OS and modern surfaces a nightmare to operate on as it took extremely long time to swap in the entire working set every time the process gets suspended.

2.3. File Management

Windows 10 S uses the New Technology File System (NTFS). This filesystem provides a combination of performance, reliability, and compatibility with other formats. It's goals are to be able to read, write, and search on large disks quickly and reliably. This isn't exactly necessary for most Surface tablets as their disk drive space should be smaller. NTFS can support partitions up to approximately 256 TB so space is generally never an issue [17]. Built with security and multiple users in mind, it has access control and multiple types of configurable permissions for various file operations. There are various file types that aid in functionality and organization of the operating system. The file system format is indexed. This indexing is done by B-Trees. Each of these files have the following attributes:

General

- Name
- File type
- Opens with

Security

- Groups or users
- Permissions: full control, modify, read & execute, read, write

Location

- Location, size, size on disk

Time

- Created, modified, accessed

There is a defined structure that these files are laid out in. At the root of the system, there are the following folders:

- Windows

- Users
- Program Files
- PerfLogs

The system contains system files here but they are hidden because most users do not need to alter or view them. NTFS also supports multiple data streams. This allows multiple different data attributes of a file to each be represented by a stream. The NTFS format supports file compression and encryption. Files are automatically compressed upon closing and are decompressed when attempting to read. File encryption is supported in NTFS version 3.0 and above. Users can choose to encrypt files they have permissions to with access lists. When users request the encrypted folder or file, it is sent to them with a challenge, if the user has the private key to these files, the operating system will decrypt the challenge, and send the result back to the system automatically. The system will then decrypt the contents and serve them to the user. This decryption is done in the kernel so that the key is not left in any accessible memory pages[17]. Users that do not have access rights will receive an “access denied” notification until they can successfully complete the challenge. There are no passwords to remember or that could be brute forced because this is all designed with access control lists.

2.4. I/O Scheduling

I/O scheduling or may called disk scheduling is a method to schedule I/O requests are coming from different processes. There is only one I/O request can be served at a time. That means I/O manager should schedule the coming I/O requests, so the disk controller could serve as much as possible requests in the least time possible. The importance of I/O scheduling algorithms is that the requests are dealing with the hard disk drive which is one of the slowest parts of the computer. Thus, we need to schedule the requests in the most efficient way to serve them in the least time possible.

There are multiple factors affecting the access time of the disk such as seek time, rotational latency, and transfer time. Seek time is the time needed for the disk arm to find the track of targeted data. The rotational latency is the time that the sector needs to rotate into a position in which it will be under read-write head. The transfer time is the time needed for the data to be transferred. The sum of the of seek time, rotational latency, and transfer time is the disk access time. All I/O scheduling algorithms try to minimize the access time by minimizing one or more of these three times.

In Windows 10 S, The I/O system is packet driven. Most I/O requests are represented by an I/O request packet (IRP.) An IRP is a data structure that contains information completely describing an I/O request [19]. For each I/O operation, the I/O manager creates an IRP to represent that operation. The I/O manager passes a pointer of the correct driver to the IRP. On the other hand, the driver will get the IRP then will perform that IRP operation. After the I/O operation has been completed, the I/O manager will receive the IRP from the driver.

2.5. Power Management

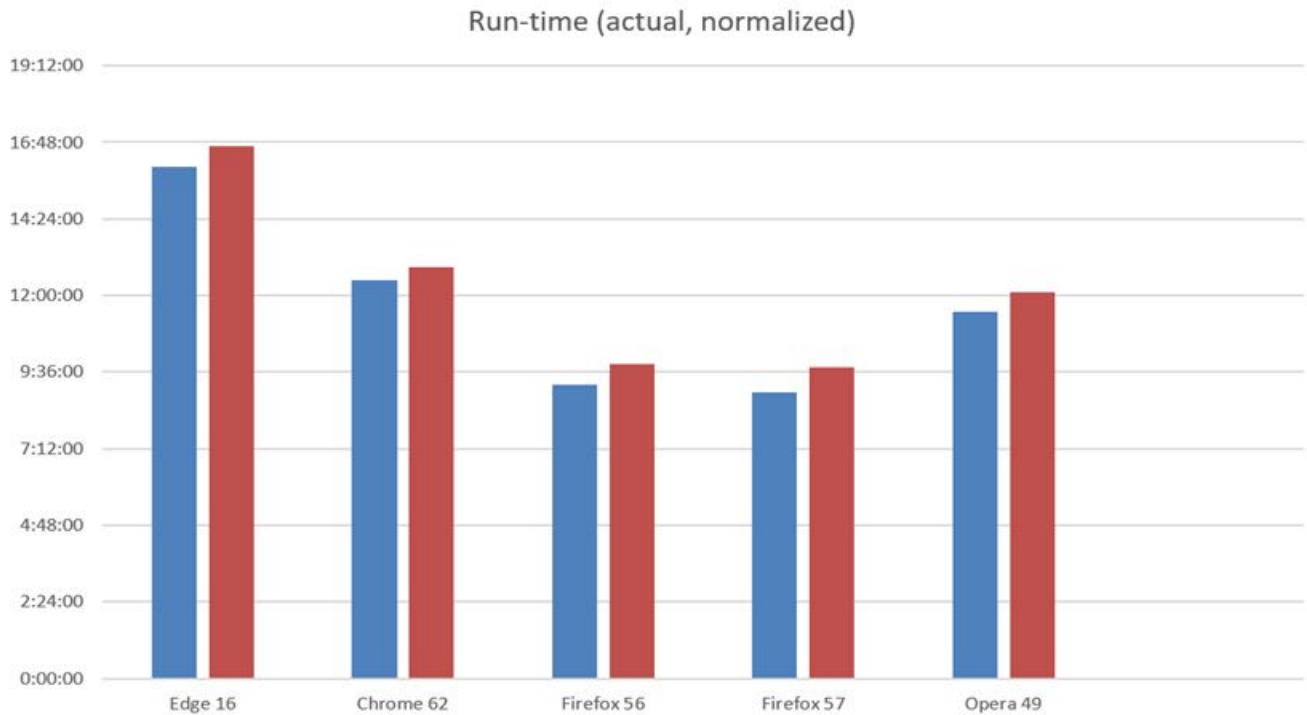
Windows 10 S uses different ways to save the device energy, so it could run the longest time possible. The most common way of wasting the power is when the device is on, but it is not used. There are two techniques to save device's power when the device is not in use. The operating system will detect when the device is not in use then it will cut it over to the sleeping or hibernating mode. Sleeping mode stores the state of the computer in the RAM while hibernation stores the state of the computer in the Disk. The computer would take longer time to wake up if it is in the hibernation mode than if it is in the sleeping mode. To wake up from the hibernation, the device will need about thirty seconds which might be frustrating for some users. However, hibernation does not consume battery

power because it turns off the power use. If the device is left in the sleeping mode and it is unplugged, after a certain time the computer will be put in the hibernation mode.

Windows 10 S provides many settings to reduce energy consumption. For example, by default Windows 10 S devices are using the Wi-Fi while the device is sleeping. However, this setting could be changed, so the Wi-Fi will be turned off while the device is sleeping.

For Windows 10 S devices that are running on a Qualcomm Snapdragon benefits high power performance. A PC running on a Snapdragon processor uses less power than other PCs, so you can go through a typical work or school day without running out of battery and worrying about finding an outlet to plug into. If you want to use the PC for something more fun, you can enjoy up to 20 or more hours of local video playback between battery charges [9].

Windows 10 S benefits from the low-consumption of its default browser Microsoft Edge. Last year, the Microsoft Edge team tested the battery consumption while streaming HTML5 video in full screen using different browsers. Microsoft Edge browser runs for the longest time among other browsers such as Chrome, Firefox, and Opera. The graph below shows the time each browser took until it turns off.



[10] Results of Microsoft Edge team test.

2.6. Security

Microsoft says that Windows 10 S is “Streamlined for security and superior performance” [16]. There are not any significant new security features put into place to stop attackers, but instead they limited the attack surface of the device. This means there are not any SMB, RPC, NetBios, or other ports open like many other Windows operating system versions have open. The Wannacry ransomware, which infected computers around the Internet recently, is an example of how this OS is more secure than Windows 10. A Windows 10 S Surface is in its default configuration, is on the same network as the Wannacry ransomware, then it would not become infected. This is because SMB ports are closed. The obvious disadvantage of this reduced attack surface is less functionality[15]. However, most standard users will not notice the difference, as they do not use this functionality. Since there are

fewer programs, tasks, features, and services to take care of, the operating system is more efficient than standard Windows 10, so Microsoft's claims are supported.

Most all other operating systems allow you to write your own programs, install third party applications, and execute system commands through a terminal or command prompt. Windows 10 S does not allow the user to do any of these things, which again reduces the attack surface of the device for attackers[15]. Third-party and free and open source software is often times the easiest things to exploit because they do not invest as much time or money into secure coding practices and penetration testing like Microsoft does with their products. Attackers will run their own programs on operating systems to further elevate their privileges, scan the system for sensitive information, or other post-exploitation activities that leads to harm. Since Windows 10 S does not allow this, it makes it that much harder to attack, but not impossible. Applications from the Window's Store have many exploits that could be used by attackers. For example, searching for Microsoft Office exploits on exploit-db.com yields over 60 results. Adobe, which is also available in the Windows Store, has had more than 350 exploits created for it. The Windows Store contains many applications that have many vulnerabilities because of the nature of current software development practices but also because it is extremely easy to publish a certified application in the store. Essentially, the application just can't contain malware has to pass several technical compliance tests [18]. Just because it comes from the Windows Store, does not mean attackers can not easily exploit it.

Windows 10 S comes with most all security features included in Windows 10 like automatic updates, memory protection, secure password storage, and built-in anti-virus. Data Execution Prevention (DEP), Address Space Layout Randomization (ASLR), and Structured Exception Handling Overwrite Protection (SEHOP) are all great built in memory protection features that protect the system from attackers attempting to use memory manipulation techniques to control the computer's flow of

execution[15]. DEP makes the OS mark the pages in memory that it does not need to execute for the program to run. Attackers will often attempt to write their own code in the heap, stack, or memory pools and then execute it later. ASLR makes the OS load DLLs into random memory addresses. This makes it difficult and sometimes nearly impossible for attackers to execute the code they previously placed on the system because the associated program should always be using different memory addresses. There are ways to get around this though with some advanced exploit development techniques. Some exploits take advantage of overwriting the exception handler which is what SEHOP attempts to prevent. SEHOP prevents overwrites of the exception handler at runtime. Combining all of these features makes Windows exploit development difficult. If the device is compromised by an attacker, passwords are NTLM hashes which makes password cracking very difficult. On top of this, Windows 10 S comes with a default antivirus solution called Windows Defender. This anti-virus does not just look at executable signatures, but also protects against system files and settings from being altered from untrusted processes. Windows Defender also analyzes the properties of the content it is about to execute. If it comes from an untrusted place, was stored in an odd location, etc it will be closely examined by Windows Defender.

Windows 10 S also comes with several hardware security features that you can configure yourself with the correct hardware and firmware. Since July 28, 2016 all device models must implement a trusted platform module (TPM) which gives the device a variety of security functions[12]. Since all devices have TPM 2.0 enabled by default, users can easily configure Secure Boot and BitLocker. Secure boot ensures that the operating system only loads trusted applications during the device's boot process[11]. This means that the pre-boot environment has a database of all applications it believes are trustworthy and their corresponding public certificates. When a component is being

loaded the name and signature is matched against what is in the database. This is important because attackers could potentially install rootkits on the system that could maliciously alter the boot process.

Secure boot will stop most rootkits. Attackers may also attach other malicious devices to the host device during the boot process to inject malicious applications or change information but Secure Boot also stops this. BitLocker is the other great tool that protects user's files confidentiality and integrity. BitLocker uses the device's TPM during boot and shutdown to decrypt and encrypt the user's files while offline[13]. This security prevents attackers from removing the hard drive to perform malicious analysis on the device's contents that could reveal: personal information, passwords, and Windows license information. Attackers must decrypt the drive before they could do any of those things.

2.7. Application Programming Interface (API)

The operating system comes with a great application programming interface (API) to develop and run applications on all Windows platforms. This framework is called Universal Windows Platform, it was designed so that users could easily create Windows 10 applications and allow Xbox One users to use them as well and vice versa. In order to make this possible, Windows includes the runtime libraries in every platform that are required in Universal Windows Platform programs. Application development is done by setting up Microsoft Visual Studio, a development environment, and downloading the Universal Windows application add-on. Visual Studio offers a complete development environment that allows you to use several Microsoft libraries and runtimes to rapidly make an efficient and secure application. UWP runtime libraries also can automatically provide services to applications such as: memory management, type safety, exception handling, garbage collection, and thread management. When applications are compiled optimized machine code is created for the application, and Microsoft is continually optimizing their compilers for their code. .NET Core Framework can also be used when developing UWP applications. This framework contains libraries

for other modern applications and cloud computing tasks. .NET Core is also contains all of the previous .NET APIs in the Windows Store app development repository[14]. This allows programmers to utilize the same libraries and functions they are used to.

3. Unique Characteristics

There are many characteristics Windows 10 S has which may not found in other operating systems. The only apps could be used in Windows 10 S are those are downloaded from the Windows store. However, users could upgrade Windows 10 S to Windows 10 Pro which will allow them to use desktop apps. In Windows 10 S the default browser could not be changed. Microsoft Edge is the default browser and you are not allowed to set other browser as a default browser. Moreover, the default search engine in Edge browser is Bing and the user is not allowed to change that. Third party application cannot be used on the platform. Windows 10 S blocks all scripts and applications not from the Windows store. This is a severe disadvantage for gamers and developers. The creation of Windows 10 s, a OS that can seamlessly switch between desktop and mobile OS. It is extremely unique as before Windows, Canonical, developer of Ubuntu, tried to implement an OS that would easily switch between desktop and mobile OS, but they were unsuccessful. Microsoft has developed a robust API that guarantees cross platform capabilities [3], something even the iOS cannot say for sure. Another unique characteristics is the UWP implement on Windows 10 devices, Now, developer only needs to focus on one platform and the API guarantees that it will run on all platforms. This provides the developer more options of freedom and creativity and spend less time debugging issue due to cross-platform incapacibilities.

4. Opinions on Usage

4.1. Simon Owens

The Windows 10 S operating system (OS) is well designed and will be able to accomplish everything most users want. In my opinion, this is one of the best operating system's Microsoft has delivered because it removes tons of the unnecessary functionality in Windows 10 that most users won't ever use like: the ability to run scripts and non-Windows Windows store applications. However, for more advanced users or gamers, this OS will severely limit their capabilities. I would use this operating system on a tablet, what it was designed for, but I could not accomplish any school or work because I often use 3rd party applications and have to write my own applications and scripts. For a laptop, I prefer Ubuntu over Windows 10 s because it is lightweight, efficient, and extremely customizable design. For a desktop or server, I would prefer standard Windows because it is the most supported and easily configurable. Everyone's use case is different, so they must decide what they need and want before using piece of hardware and OS.

4.2. Kunal Mukherjee

Windows 10 S is one of the most versatile well thought product design. My opinion will be based on two different aspects: developer and user. As a developer, I will be very content with Windows 10 as now I can write applications which are backward compatible with Windows 10 as well as the latest UWP applications. The backward compatibility was possible with the implementation of the standby list. Also, I would prefer to use this OS because of UWP platform itself. The fact that I can write an application respective to what OS I want to write on and not the type of device that is running the OS, makes my life as a developer way easier. I do not have to spend time and energy as to debug issues that were not present in the native platform that I developed for but now it is present as the

platform has changed. The programmable API is also well documented and well developed. That means it opens plethora of opportunities for me as a developer to develop application in a very customizable manner. The only downfall, I can comprehend is in terms of connection functionality as many ports are closed off or removed altogether due to the improvement security vulnerabilities. The price of this trade off maybe that the developer would not be able to utilize the full potential of the API, as some of them would not work as the connection port that they require does not exist.

As an user, I would prefer to buy hardware that run Windows 10, as I will have a wide range of cross platform compatibility usage. For example, an app that I like running on my surface pro, will be exactly the same as running the app in my personal desktop or laptop. I would not need to learn any new usage patterns to able to use the app in the new platform.

4.3. Abdulaziz AlShabibi

I would not use Windows 10 S neither as a user nor programmer because of the software restriction it has. Microsoft is trying to enforce users to use Windows store applications. There are limited apps on the Windows store. I consider Windows 10 S both a desktop and mobile operating system at the same time. As a mobile operating system, there are better alternatives such as iOS and Android, both iOS and android are mobile operating systems that run on the tablet or phone. Apple store on iOS and Google Play on Android offer much more applications than Windows store does. As a desktop operating system, if I wanted to use Windows operating system, I would prefer to use Windows 10 rather than using windows 10 s. Windows 10 S might be designed for simple users who want to use simple kind of tasks like streaming, emailing, and browsing. Windows 10 S might be safer since windows store applications are verified by Microsoft. However, not being able to use non-windows store apps is very restricting to make me not wanting to use Windows 10 S.

5. Conclusion

Windows 10 S is an extremely consumer friendly as well as service oriented operating system and the use of UWP has made this OS one of the most powerful OS out there. Windows 10 S users have the possibility of purchasing apps that work on all the devices running windows 10, along with constant updates of improvement from Microsoft, ultimately reaches the goal of creating a platform that is customer friendly. But, unfortunately due to certain minute but important differences between the mobile os and the native Windows 10, certain apps could not be accessed in certain platform, mostly because the developer chose so. Windows 10 S has also tried to give certain privileges to the user by the addition of Powershell and the inclusion of Bash. The power to Windows S could be seen by the fact, to transform of a Windows 10 S to a much more traditional view of Windows 10, one just has to plug in an external monitor. UWP has made creating cross platform apps for the Windows 10 very easy as well as making the developer experience more smooth and error free app creation. Overall, Windows 10 S makes the user and developer experience much more streamlines, make this platform attractive to both.

Appendix A: Sources and References

1. Eric S. Raymond. "daemon". The Jargon File. Retrieved 2008-10-22.
2. Whitney, Tyler. "What's a Universal Windows Platform (UWP) App? - UWP App Developer." UWP App Developer | Microsoft Docs, 27 Mar. 2018,
<<https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide>>
3. Krzyzanowski, Paul. "Process Scheduling." Process Scheduling, 18 Feb. 2015,
<<https://www.cs.rutgers.edu/~pxk/416/notes/07-scheduling.html>>
4. Khanse, Anand. "Processor Scheduling in Windows 10/8/7." The Windows Club, 28 Feb. 2016,
<www.thewindowsclub.com/processor-scheduling-in-windows-7-8>
5. Microsoft. "About Processes and Threads." About Processes and Threads (Windows), 2015,
<[https://msdn.microsoft.com/en-us/library/windows/desktop/ms681917\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms681917(v=vs.85).aspx)>
6. Denning, Peter j. "The Working Set Model for Program ." Massachusetts Institute of Technology, Cambridge, 1968, <http://denninginstitute.com/pjd/PUBS/WSModel_1968.pdf>
7. Windows. "Working Set."
[https://msdn.microsoft.com/en-us/library/windows/desktop/cc441804\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/cc441804(v=vs.85).aspx), 2016,
<[https://msdn.microsoft.com/en-us/library/windows/desktop/cc441804\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/cc441804(v=vs.85).aspx)>
8. Anderson, Kareem. "Microsoft Educates Insiders on How Windows 10 Handles Memory." On MSFT, 20 Aug. 2015,
<www.onmsft.com/news/microsoft-educates-insiders-windows-10-handles-memory>.
9. Microsoft Windows 10 S on a Qualcomm Snapdragon processor,
<https://support.microsoft.com/en-us/help/4020089/windows-10-s-faq>
10. Browser efficiency comparison - Fullscreen Video Microsoft Edge team, December 2017,
<https://microsoftedge.github.io/videotest/2017-12/>

11. DawnWood. "Secure Boot." *Microsoft Docs*,
docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot.
12. DawnWood. "Trusted Platform Module (TPM) 2.0." *Trusted Platform Module (TPM) 2.0 | Microsoft Docs*, docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-tpm.
13. DawnWood. "BitLocker Drive Encryption in Windows 10 for OEMs." *Microsoft Docs*,
docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-bitlocker.
14. TylerMSFT. "What's a Universal Windows Platform (UWP) App? - UWP App Developer." *UWP App Developer | Microsoft Docs*,
docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide.
15. Justinha. "Mitigate Threats by Using Windows 10 Security Features (Windows 10)." (*Windows 10*)
| *Microsoft Docs*,
docs.microsoft.com/en-us/windows/security/threat-protection/overview-of-threat-mitigations-in-windows-10.
16. Rashid, Fahmida Y. "Respect: Windows 10 Security Impresses Hackers." *InfoWorld*, InfoWorld,
15 Aug. 2016,
www.infoworld.com/article/3107132/security/respect-windows-10-security-impresses-hackers.html
17. "What Is NTFS (NT File System; Sometimes New Technology File System)? - Definition from WhatIs.com." *SearchWindowsServer*, searchwindowsserver.techtarget.com/definition/NTFS.
18. JnHs. "The App Certification Process - UWP App Developer." *UWP App Developer | Microsoft Docs*, docs.microsoft.com/en-us/windows/uwp/publish/the-app-certification-process.
19. Understanding the Windows I/O System, Microsoft, Mark E. Russinovich, Alex Ionescu, and David A. Solomon
<<https://www.microsoftpressstore.com/articles/article.aspx?p=2201309>>