The Progression of Home Console Operating Systems

Gavin Sutherland
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO, United States
gas2bt@umsystem.edu

Jason Ni
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO, United States
jnwkb@umsystem.edu

Hariswar Baburaj
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO, United States
hbkkb@umsystem.edu

Abstract—**Operating systems (OSs) serve the critical role of connecting the hardware and software of a device. As the hardware continues to improve and the needs of the software continue to rise, OSs must improve to meet these needs. The gaming industry serves as an interesting microcosm of this three-way race. By researching the three major gaming companies, Nintendo, Sony, and Microsoft, we can analyze how the progression of hardware and each company's goals impacted the OSs they designed. We found that Sony built its OSs with an emphasis on maximum performance, Microsoft built theirs off of Windows for both compatibility and ease of implementation, and Nintendo built theirs to be as minimal as possible to support cheaper and more accessible hardware.**

Keywords—operating system, memory, performance, gaming

I. Introduction

The video game industry has undergone a tremendous transformation since its early days, shaped by technological innovation and fierce competition among major players. Console gaming enjoyed a surge in popularity during the late 1970s and early 1980s before experiencing a significant downturn with the video game market crash of 1983. This decline was reversed by Nintendo's introduction of the Nintendo Entertainment System (NES) in 1985, which helped reestablish consumer trust and revitalize the industry. In the following years, Sony entered the market with the launch of the original PlayStation in 1994, quickly establishing itself as a powerful competitor. Microsoft followed with the release of the Xbox in 2001, further intensifying the competition. Despite the significant impact that console operating systems have on user experience, system performance, and game development capabilities, there has been limited academic focus on how these systems have evolved across generations and among leading companies. This paper addresses the gap by examining how Sony, Microsoft, and Nintendo have developed and refined their console OSs over time. Understanding the progression of the operating systems of these three big companies helps us understand the history and potentially the future of operating systems.

II. Nintendo

Nintendo is a Japanese video game company. Since entering the industry in the late 1970s, it has become a household name internationally, selling over 5.5 billion games and 800 million hardware units [1]. Since their inception, their focus has been to deliver unique and accessible gameplay experiences. From a technical perspective, this requires the console hardware, and thus the operating systems included, to be as minimal as possible, as it results in a cheaper console with as much room as possible for developers to bring their games to life.

A. Early Nintendo Consoles (1985-2005)

While it is standard practice for modern consoles to come packaged with a fully functioning OS, this was not always the case. Early Nintendo consoles had basic firmware, but the bulk of traditional OS operations were handled by the cartridge/disk. The earliest example is the Nintendo Entertainment System (NES), which completely lacked a dedicated OS [2]. Instead, a game cartridge would connect directly to the CPU. The game would handle all memory allocation itself, while the graphics would be processed by a chip called the Picture Processing Unit (PPU). The cartridge itself could also contain more RAM and ROM as needed for the game. The issue with this system was that the 16-bit address bus on the CPU could only address 64 KB of memory. Developers worked around this by using a chip called a Mapper. A mapper divided the program ROM into groups of addresses called banks. While the CPU always sees the same amount of memory, the program can control the mapper to switch between banks as needed. This allowed developers to bypass this major weakness of the console, especially due to how cheap it was to implement [3].

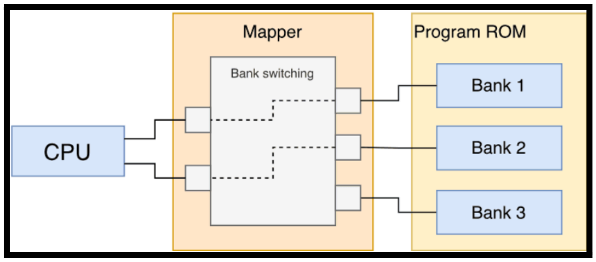

Fig 1: Simple representation of how the mapper expands CPU addressing range.
Source: [3]

Unlike its predecessors, the N64 included a kernel in its design, though it's so minimal that it can hardly be called a full OS. Games are run in Kernel mode and can take advantage of many pre-written operations. The kernel features threading, communication between process threads, interrupts, and events. This allows for more efficient communication between the system and the game itself. The kernel also supports virtual addressing through direct mapping or the Translation Lookaside Buffer (TLB) [4]. While still a long way from a full OS, the N64 represented a major step up in kernel functionality for Nintendo consoles.

The final console of this era was the GameCube. The functionality was extremely similar to that of the N64. Dolphin OS (the internal name for the firmware. It provided games with an array of low-level system calls and variables, but did very little outside of that [5]. A significant change from the N64 was the memory structure. With the N64, the GPU and CPU would often compete for the limited RAM, leaving the CPU idle up to 50% of the time. They resolved this issue by giving the GPU a dedicated space in memory, which drastically improved the performance of the console [6].

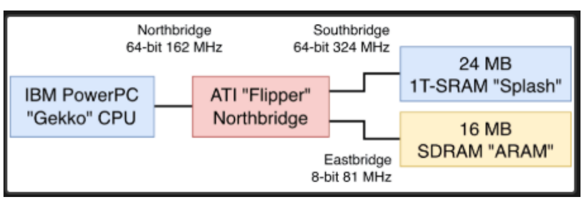

Fig 2: Structure of the GameCube memory
Source: [5]

B. *The Transition of the Wii (2006-2011)*

Released in November 2006, the Nintendo Wii marked a notable shift in firmware implementation. Rather than a kernel that the program interacts with directly, the Wii uses a program called IOS (likely standing for Internal Operating System or Input Output System) to provide a layer of abstraction between the programs and the internal components. It features a microkernel that interacts with the various components through drivers, organizes and schedules processes to be performed on the CPU, organizes file systems, and handles basic security and encryption. The user could interact with the console through a secondary program called Broadway. These two programs communicate with each other through an Inter-Process Communication (IPC) Protocol [7]. It's debatable whether IOS should be considered a true operating system or simply advanced firmware, but the additional layers of abstraction marked a unique shift in the company's implementation of their control systems.

C. *Modern Nintendo Consoles (2012-2024)*

The Wii U was the first Nintendo home console to have a fully functional OS. While it still contains an upgraded version of the Wii's IOS called IOSU, it also runs another operating system called Cafe OS. It contains the kernel, the executable loader, and various system libraries. The kernel handles basic OS tasks such as process isolation, memory management, and interrupt handling [8]. Applications run as user-mode processes within the OS. These applications, as well as others such as an interactive shell and other services generated by the OS, run on top of the kernel. The kernel then uses the same IPC protocol as the Wii to communicate with the IOSU. All of these extra features come at the cost of 1GB of RAM, an issue that was rectified with Nintendo's next console [9]. The inclusion of IOSU also allows the Wii U to be completely backward compatible with the full Wii game library, which was one of the main draws of the console at the time.

The most recent Nintendo home console at the time of writing is the Nintendo Switch. This console is a unique hybrid of a home and handheld console. Rather than adapt the Wii U operating system, the Switch has a custom microkernel called Horizon, which was an upgrade to the 3DS microkernel of the same name. The OS has two main objectives: to be lightweight and secure. The OS is organized as a microservice architecture with each user process being sandboxed for increased security. The system runs full ASLR, a security measure designed to prevent memory exploitation [10]. This does not negatively impact system performance [11]. The memory uses both paging and slab allocation [10], which is optimized to support frequent allocation and deallocation while avoiding fragmentation [12]. Overall, while the OS is not impressive by modern console standards, it manages to accomplish what it needs to do while taking up as little space as possible, perfect for a handheld device.

III. Sony

Sony Interactive Entertainment (SIE), a subsidiary of the Japanese conglomerate Sony, created their very first console, PlayStation, in 1994. SIE has produced a total of 5 generations of consoles, with a total of 9 different consoles [13]. We will be following the progression of the PlayStation Operating System, specifically the PlayStation, PlayStation 2, PlayStation 3, and PlayStation 4. By researching the progression of different generations, we can better understand how SIE advanced its technology and shaped the modern gaming experience over time to suit its consumer needs.

A. Playstation

PlayStation does not have an official Operating System. Instead, it uses PS-X OS, which is a firmware that acts like an operating system. It initializes the system, handles I/O, and manages memory, graphics, sound, and CD-ROMs. It should be noted that PlayStation is one of the first major gaming consoles that uses CD-ROMs instead of cartridges. The PS-X OS is a single-tasking operating system that can only perform one task or game at a time, where all the system resources will be dedicated entirely to it. However, with a bit of tweaking, it is possible to set the system into a multitasking operating system, but its usage is mainly for developers [14].

To boost the performance of the PlayStation, the size of the RAM in PS-X OS is to be kept at a maximum of 64 KB, and it was also designed so that the CPU occupied time is minimized. They also did something very bold, which is never to be seen in today's operating systems, which was to ignore the checks of prohibited items for greater speed [14]. The PlayStation BIOS does routine things such as commands for the CD-ROM drive, filesystem operations, multithreading, and standard C functions [15].

In general, the PlayStation is not designed to be portable, as its design is specifically made for high performance, speed of processing, and disc-based storage. One of the main goals of development was to revolutionize the usage of CD-ROMS in gaming, rather than using cartridges.

B. PlayStation 2

The PlayStation 2's operating system is a compact, modular, and secure system stored in ROM. It provided low-level services to games via system calls and exposed a user interface called OSDSYS. The OS architecture was split between the Emotion Engine (CPU) and I/O Processor, each with its own startup routines and responsibilities. While appearing simple, the OS included forward-thinking features like runtime patching, modular driver loading, and support for secure system updates, making it both flexible and efficient for its time. This is a huge difference between the simple structure of the first PlayStation [16].

Next to the CPU of the console lies 2 blocks of 16 MB of RAM, giving a total of 32 MB of memory for the PS2. The RAM was strategically placed to allow dual-channel architecture, which consists of connecting both RAM modules using two independent 16-bit buses to improve data throughput. The resulting setup provided a theoretical 3.2 GB/sec; thus, memory latency was not a problem for this console. Data transfers are done in batches of 128 bits, but after every 8 batches, the main bus is temporarily unlocked, leaving a small window to perform other DMA transfers in parallel or let the CPU gain access to the main bus [16].

C. PlayStation 3

The PlayStation 3 operating system is a complex, modular system designed to manage both gaming and multimedia features. It evolved from traditional game console firmware into a full-fledged, multi-layered operating system. Unlike older systems (e.g., shell, BIOS), PS3's OS includes boot loaders, kernel, user interfaces, and various runtime modules [17].

It runs on the powerful Cell processor and uses a multi-layered security model with three privilege levels: a Hypervisor (Level 1), Kernel (Level 2), and User Programs (Level 3). The OS includes a chain of encrypted loaders and system files to ensure secure boot and operation. Its graphical interface, the XrossMediaBar (XMB), allows users to navigate games, apps, and media [17].

- Hypervisor: The most privileged layer, developed by Sony. It controls all hardware access, handles exceptions via the MMU, and only responds to Sony-authorized code. It provides low-level system calls and FAT16 support.
- Kernel: The OS kernel abstracts LV1, manages system resources, and provides multi-threading for both PPU and SPU. It also initializes user-level modules.
- User Programs: Where games, the graphical shell (XMB), and all user applications run. These programs cannot directly access hardware or start new processes without kernel approval.

Sony originally supported installing a secondary OS (like Linux) through the OtherOS feature, enabling scientific use of the PS3's powerful hardware. Although later removed due to security concerns, it highlighted the console's potential beyond gaming. The system receives updates via PUP files, but only specific, non-encrypted parts can be changed [17]. Overall, the PS3 OS is a sophisticated example of modern console operating systems, balancing performance, flexibility, and security.

D. PlayStation 4

The PlayStation 4 operating system, Orbis OS, is a modified version of FreeBSD 9.0. FreeBSD is a free version of BSD Unix that is generally fairly compatible with most Linux applications, and to the untrained eye, a BSD-based system looks a lot like Linux [18]. The PlayStation 4's user interface represents a clear evolution from the PlayStation 3's XrossMediaBar (XMB), maintaining its predecessor's speed and performance while introducing greater flexibility. While the new interface initially appears distinct, users familiar with Sony's ecosystem will recognize its lineage as the layout is influenced by Sony's earlier design, especially in how it balances speed with usability. Improvements such as the ability to suspend games mid-session and switch settings without quitting, smoother navigation with minimal lag, and simplified user account management mark significant usability enhancements over the PS3. Features like temporary profile downloads and intuitive app switching further modernize the experience [19].

Despite these upgrades, the OS retains some of the PS3's structural complexity. Menus and submenus can be confusing, with overlapping categories and options scattered across different sections. For example, friend requests and notifications appear in multiple places, creating redundancy and potential confusion. The system also lacks tools for organizing digital content—installed games are displayed in a long, horizontal list with no way to sort or group them. Additionally, features such as a proper rest mode and high-quality video capture were missing at launch, limiting the OS's functionality early on. Overall, the PS4's operating system is a step forward in usability but still carries some of the organizational issues of its predecessor [19].

FreeBSD is licensed in such a way that Sony can use as much or as little of the code as it likes, all without needing to share the code [13]. Thus, any actual information about Orbis OS is difficult to find. Especially with the lawsuit in the past against people who tried to dig into Sony's console OS [20].

E. Summary

The evolution of the PlayStation operating system from PS1 to PS4 shows a clear progression from simplicity to complexity and versatility. The PS1 used PS-X OS, a minimal, single-tasking firmware focused on speed and basic system functions like I/O and memory management. With the PS2, Sony introduced a modular ROM-based OS with system calls and better hardware coordination, enabling more efficient data handling and dual-channel RAM architecture. The PS3 marked a major shift to a full multi-layered operating system, featuring a secure hypervisor-kernel-user structure and the XrossMediaBar (XMB) interface, supporting both gaming and multimedia. Finally, the PS4 adopted Orbis OS, based on FreeBSD 9.0, offering a modernized interface with multitasking, suspend/resume features, and improved performance, though some UI and content organization issues persisted.

IV. Microsoft

Next, we will take a look at all the Xbox generations' operating systems and see the progression they made each time a new gen has been released. But before that, let's dive into the background of Microsoft and Xbox. In the late 1990s and early 2000s, consoles such as the PlayStation and N64 propelled gaming from a niche industry to a pillar of modern entertainment. That's where Microsoft saw an opportunity to develop Xbox, inspired by the PlayStation's success and the upcoming release of the PlayStation 2. So, they released their first ever console called Xbox in 2001. The growth of each new console was rapidly expanding, but soon had its competition rise with the release of a new generation of PlayStation from Sony Interactive Entertainment, which also had new features and immense growth in the industry. The recent Xbox Series X has quality graphics and performance at the cost of 500$. The goal of Microsoft was to enlarge the cloud gaming market and add cross-platform support for users, which creates a community of Xbox experience with pc or PlayStation.

A. Xbox

The original Xbox was released on November 15, 2001, by Microsoft. The operating system was based on the version of Windows 2000. It is combined by the kernel and the userland application. Since the technologies were super new during those early stages, they didn't have any multitasking, and the original Xbox was executed by running only one request at a time. The OS designed by Microsoft is designed to be user-friendly and provide a smooth console experience for the users. In terms of External Interfaces, the Xbox architecture includes 4 USB with 1.1 ports and 100 BASE-TX Ethernet ports, which are primarily used for web-based applications/platforms [21]. In terms of Internal Interfaces, Microsoft uses IDE Controller and SMBus to create a high-quality experience for users and a high-level performance interface [22]. Due to a lack of technology development in the 2000s, all the services, like the kernel and more, had to be compiled into one executable to enhance memory performance. The first gen of Xbox had several new features and advancements to compete with the PlayStation 2.

B. Xbox 360

The Xbox 360 was released on November 22, 2005, 4 years after the original Xbox. The Xbox 360's operating system is based on an improved version of Windows [23]. It was mainly manufactured to implement new features, such as adding multiplayer to make a better experience for its users. Some key features and improvements compared to the previous console in terms of architecture were that the Xbox 360 now supports numerous kinds of media, and the kernel supports many different applications for each one. And with this gen, the console can execute several games compared to the previous console, a vast variety in terms of games that can be run. Xbox 360 also has a modifiable OS, which means that the software developed by Microsoft can download packages in the console to provide a better gaming experience for the user base.

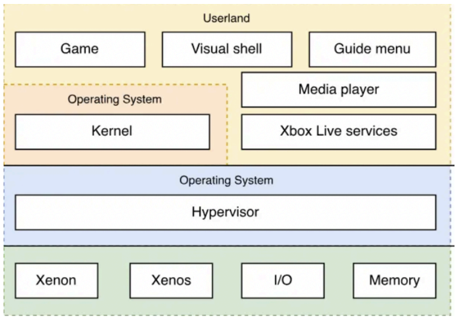

Fig 3. Architecture of the Xbox 360
Source: [23]

C. Xbox One

Xbox One was released on November 22, 2013. Microsoft took a different approach to the operating system compared to previous consoles. The operating system used for Xbox One is the Host OS, which is used to run the Virtual machine and provide assistance for hardware materials. It also helped with providing visualization for the Xbox One. Even though there were few upgrades in terms of OS compared to previous consoles, Microsoft still created the Xbox One OS based on Windows 8. But in 2015, they made a move by switching the OS based on Windows 8 to Windows 10. In terms of gaming visualization, it was better and more structured, even though it was based on Windows 8 LNM. The storage capacity did increase and had an ability of 8 to 10 GB [24]. The console was also designed to include cloud integration and provide better gaming performance and with smoother upgrades and installation.

D. Xbox Series X/S

Microsoft's most recent console, the Xbox Series X, was released on November 10, 2023. The operating system that was developed for Xbox Series X is called Xbox System software, which is primarily based on Windows 10 and 11, which means that Microsoft is using the Windows kernel and the DirectX [25]. DirectX is primarily used in the Xbox to allow all the Windows implementations to work with the graphics of the game and the quality of what the Xbox produces. In terms of comparing it with past gen consoles, this is Microsoft's fastest and most powerful console they have developed so far. Microsoft also included an expansion of memory usage that can be installed on the console. Some additions that were done by microsoft is that they designed the Xbox series X to allow this console to play all different games that were played from previous Xbox gen which creates a new liking for players to test out games that was developed in early stages and how the graphics turned out to be in early generations. Xbox Series X is fully optimized for cross-play, which allows for more flexibility for users to play with people who have different gaming systems, unlike previous Xbox consoles. With the performance of Xbox Series X, Microsoft has worked on providing the best for Xbox customers. The graphics have turned from 8-bit to 16-bit compared to last gen; it developed from 2D to 3D and SD to HD, also moving to 4k as well [25]. According to Jason Ronald, Director of Product Management for Xbox Series X, "The Xbox Series X will deliver a massive increase in GPU performance and continue to redefine and advance the state of the art in graphics with new capabilities such as hardware-accelerated ray tracing" [25]. This shows the enhanced technologies and better architecture implementation that have been created by Microsoft in the Xbox Series X/S.

V. Conclusion

Each of the major gaming companies uses the same operating system technology of their time to achieve very different ends. Sony's focus on delivering the most powerful console of any generation leads to operating systems that optimize the system's resources as much as possible. Microsoft modifies its Windows Operating systems to work on the consoles rather than building them from the ground up, allowing them to focus on developing advanced features on top of their solid code base. Nintendo's focus on accessibility leads them to create OSs that perform their function with as little overhead as possible, keeping their hardware costs down and their consoles cheaper. Despite some missteps, all three companies' OSs accomplish their goals expertly. While future research may be difficult as domestic copyright laws tighten, we're looking forward to seeing how the next generation of operating systems progresses.

References

[1] Nintendo, "Nintendo History," *Nintendo of Europe AG*, 2016. https://www.nintendo.com/en-gb/Hardware/Nintendo-History/Nintendo-History-625945.html
[2] B. Cockfield, "The NES Gets Its Own OS," *Hackaday*, Sep. 29, 2022. https://hackaday.com/2022/09/28/the-nes-gets-its-own-os/ (accessed May 01, 2025).
[3] R. Copetti, "NES Architecture | A Practical Analysis," *The Copetti Site*, Jan. 25, 2019. https://www.copetti.org/writings/consoles/nes/
[4] "N64 Programming Manual," *Ultra64.ca*, 2025. https://ultra64.ca/files/documentation/online-manuals/man/pro-man/pro06/index6.1.html (accessed May 01, 2025).
[5] R. Copetti, "GameCube Architecture | A Practical Analysis," *The Copetti Site*, Nov. 19, 2019. https://www.copetti.org/writings/consoles/gamecube/ (accessed May 01, 2025).
[6] "Nintendo GameCube Programming 101," *Gdcvault.com*, 2025. https://www.gdcvault.com/play/1022547/Nintendo-GameCube-Programming (accessed May 04, 2025).
[7] R. Copetti, "Wii Architecture | A Practical Analysis," *The Copetti Site*, Jan. 05, 2020. https://www.copetti.org/writings/consoles/wii/ (accessed May 01, 2025).
[8] "Cafe OS - WiiUBrew," *wiiubrew.org*. https://wiiubrew.org/wiki/Cafe_OS
[9] R. Copetti, "Wii U Architecture | A Practical Analysis," *The Copetti Site*, Oct. 26, 2022. https://www.copetti.org/writings/consoles/wiiu/
[10] Plutoo, Derrek, and Naehrwert, "Console Security - Switch," *Media.ccc.de*, Dec. 28, 2017. https://media.ccc.de/v/34c3-8941-console_security_-_switch (accessed Apr. 29, 2025).
[11] J. Detter and R. Mutschlechner, "Performance and Entropy of Various ASLR Implementations," 2015. Accessed: Apr. 29, 2025. [Online]. Available: https://pages.cs.wisc.edu/~riccardo/736finalpaper.pdf
[12] "Allocating Kernel Memory (Buddy System and Slab System)," *GeeksforGeeks*, Mar. 08, 2018. https://www.geeksforgeeks.org/operating-system-allocating-kernel-memory-buddy-system-slab-system/ (accessed Apr. 29, 2025).
[13] "Evolution of PlayStation: The Beginning," *PlayStation*, 2024. https://www.playstation.com/en-us/playstation-history/1994-ps-one/

[14] “PlayStation Operating System,” Aug. 1998. Available: https://psx.arthus.net/sdk/Psy-Q/DOCS/Devrefs/os.pdf
[15] R. Copetti, “PlayStation Architecture | A Practical Analysis,” *The Copetti Site*, Aug. 08, 2019. https://www.copetti.org/writings/consoles/playstation/
[16] R. Copetti, “PlayStation 2 Architecture | A Practical Analysis,” *The Copetti Site*, Apr. 08, 2020. https://www.copetti.org/writings/consoles/playstation-2/
[17] R. Copetti, “PlayStation 3 Architecture | A Practical Analysis,” *The Copetti Site*, Oct. 20, 2021. https://www.copetti.org/writings/consoles/playstation-3/
[18] S. Anthony, “PS4 Runs Orbis OS, a Modified Version of FreeBSD that’s Similar to Linux - ExtremeTech,” *www.extremetech.com*, Jun. 24, 2013. https://www.extremetech.com/gaming/159476-ps4-runs-orbis-os-a-modified-version-of-freebsd-thats-similar-to-linux
[19] “Polygon’s PlayStation 4 Review,” *Polygon*, 2025. https://www.polygon.com/a/ps4-review/online
[20] D. Kravets, “Sony Settles PlayStation Hacking Lawsuit,” *Wired*, Apr. 11, 2011. https://www.wired.com/2011/04/sony-settles-ps3-lawsuit/
[21] R. Copetti, “Xbox Architecture | A Practical Analysis,” *The Copetti Site*, Jun. 26, 2020. https://www.copetti.org/writings/consoles/xbox/
[22] “Xbox OS [Original/360/XBone],” *BetaArchive*, Sep. 30, 2018. https://www.betaarchive.com/forum/viewtopic.php?t=38940&sid=8bb15de0213ac2d5644d55067ca42ad0
[23] R. Copetti, “Xbox 360 Architecture | A Practical Analysis,” *The Copetti Site*, Jun. 08, 2022. https://www.copetti.org/writings/consoles/xbox-360/
[24] “Xbox Operating System - Xbox One Research Wiki,” *Github.io*, 2016. https://xboxoneresearch.github.io/wiki/operating-system/xbox-operating-system/
[25] W. Tuttle, “Xbox Series X: A Closer Look at the Technology Powering the Next Generation,” *Xbox Wire*, Mar. 16, 2020. https://news.xbox.com/en-us/2020/03/16/xbox-series-x-tech/