An operating system for the home can be thought of as a program that acts as an intermediary between a user and devices within the home[1]. Various products have emerged over the past few decades which provide users with access to their home devices, however adopting standards for connecting devices and appliances throughout the home continues to be a challenge. A home operating system that connects to every possible device may be impractical from a system perspective. On the other hand, an overly-general operating system connecting to a handful of devices may not be useful from a user standpoint. This project reviews current standards that exist for connecting home devices.

Variation in living spaces presents another challenge for the design of a home operating system. Homes vary in terms of size, configuration, devices and appliances, and so depending on the home, an operating system may have too many or too few useable features. Although every home is unique, there are several functions that tend to be common across most homes, such as dining, productivity, social life, entertainment, or security. In some cases, a home may encompass all of these functions. In other cases, some homes may revolve around a single function. This project investigates how home operating systems are addressing these functions.

I/O plays an integral role in a home operating system as appliances and devices will connect to the system. This project investigates features such as voice input, and network I/O for a home operating system. Lastly, this project considers the implications of operating system structure, security, and protection.

Abien
Home Operating Systems
CS 575

In the mid-1990's, Bill Gates' The Road Ahead was published, which described his vision for the future based on the assumption that the costs of computing and communication would decrease over time. Around the time that the book was being written, Bill Gates was building a home in which technology could alter each visitor's experience based on their preferences for lighting, entertainment, information, or temperature. In one of the last chapters of his book, he describes how upon entering the home, visitors will take an electronic pin which will allow the house and its services to identify them and their preferences. Lighting would turn on as visitors walked into areas of the house and would dim or turn off as they left. Phone calls or news being delivered to a visitor in one room would seamlessly be delivered to another room as the visitor moved. It would also be possible to ask the house to play certain songs or films. Subsequent visits to his home would result in the home's technologies and other home features adapting to user preference and location within the house. Some of these features, such as on-demand movies and music, are now available to many. Others, such as home lighting and temperature solutions are starting to become more widely available. Although Gates' descriptions mention only services and familiar hardware [2], there would undoubtedly be a computer application that would act as an intermediary between visitors and the home's appliances and services, in other words an operating system.

Many companies have created home operating systems after The Road Ahead was published, however few have been accepted as a standard. The goals of home operating systems are often to provide a means to access and control devices. In the abstract view of a computer system's components provided in Operating System Concepts, the operating system sits between the hardware and the applications/users, as shown in figure 1. Similarly, A home operating system exists to connect the hardware, such as lighting, thermostats, or doorlocks, to applications that multiple users, or home visitors have access to. One of the biggest challenges for creating a home operating system is that devices that could potentially be connected often use different communication protocols which could include DLNA, ZWave, ZigBee, X10, or WiFi. Alternatively, monolithic systems might be used that connect multiple devices very well, but don't offer flexibility in adding new devices.

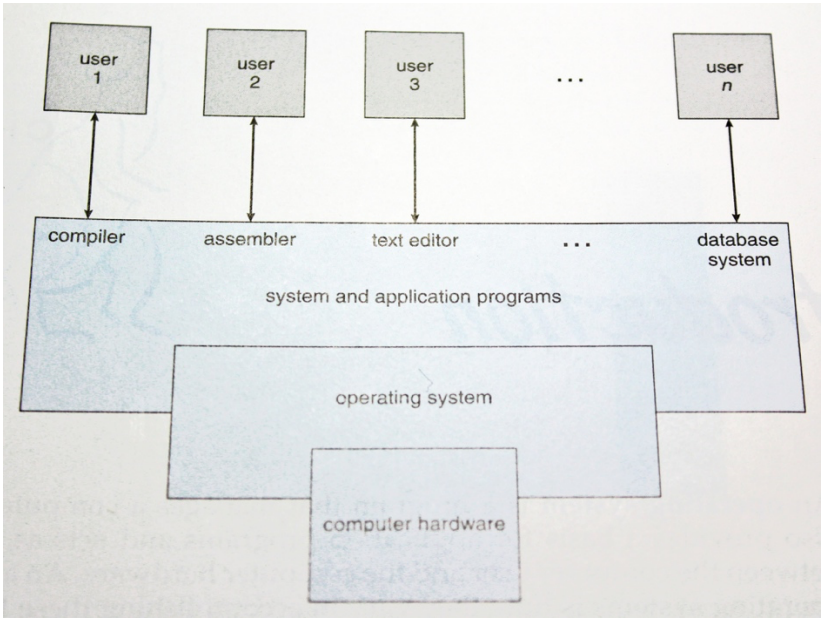

Figure 1

An operating System kernel typically uses device driver modules to manage the various I/O devices that may need to connect to it. The functions of a home operating system often means that devices cannot be connected by means of ports and buses, so network I/O can instead be used to send packets to sockets. I/O Hardware are usually transmission devices for network connections or human interface devices for audio input or output. In their 2010 research paper entitled “Embedded Server and Client System for Home Appliances on Real-Time Operating System”, the authors essentially describe a home operating system consisting of a web-server for home appliances which would have a real-time operating system installed. This web server would allow the device to be internet-connected. Home devices such as lighting fixtures, microwaves, and refrigerators could be accessed from a personal computing device. To achieve this, an open source, real time operating system development environment named T-Engine was used. A system named T-kernel is used as the operating system because of the minimal resources required. A section of software named the T-monitor acts similarly to a system BIOS, ensuring initialization and hardware input and output [3].

Home operating systems such as Crestron and Savant connect to devices using a variety of connection protocols ranging from WiFi, Radio Frequency, Zigbee, or Bluetooth. Devices are often combined into a mesh network, where all nodes are connected to one another. WiFi is a robust communication protocol used for connecting to devices however its power consumption tends to be higher than most other options. Zenzys introduced Zwave, which uses low radio waves to connect appliances over a mesh network. The Amazon Echo, Nest, and Apple Homekit are compatible with Insteon, introduced by Smartlabs, which uses radio frequency or power lines to connect devices. Zibgee is a standard that allows for the creation of personal area networks. The Zigbee alliance publishes profiles, allowing vendors to create connected products. Bluetooth is a standard that has limited range and devices, but its power consumption is very low. These standards are compared in figure 2.

	WiFi	Bluetooth Low Energy	Thread	ZigBee	Z-Wave
Network Type	Hub-and-Spoke	Point-to-Point	Mesh	Mesh	Mesh
Power	High	Low	Very Low	Very Low	Low
Data Rate	11 to 300 Mbit/s	1 Mbit/s	~90 kbit/s (app layer)	~90 kbit/s (app layer)	40 kbit/s
Range	10 to 100 meters	2 to 50 meters	Up to 300 meters	10 to 300 meters	30 meters
Security	Med	Low	Med (+)	Low	Med

Figure 2

Researchers and manufacturers have taken many approaches to developing a home operating system. One approach is that of a centralized home operating system with only specific devices connecting to it. Manufacturers such as Crestron, and Savant have created operating systems that exist on a computer and connect to devices that are proprietary or

manufactured by hardware partners. Crestron uses a Windows embedded based operating system. Savant uses an operating system built on the Mac OSX system.

Another approach to operating systems is that of manufacturers producing devices compatible with home operating systems, and various systems will exist to that can connect and control these devices. This is the approach taken by Amazon Echo, Apple HomeKit, and Google Home [4].

These varied approaches to home operating systems have resulted in designs that use a hybrid approach, includes many of the features of layered approach described in Operating System Structures. The hardware layer is the lowest layer, and the user interface exists at the top layer, as shown in Figure 3.

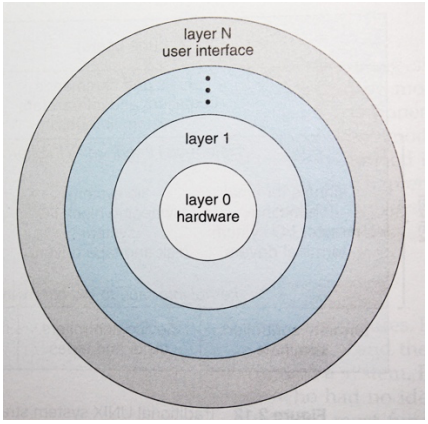

Figure 3

In 2006, researchers at National Taiwan University designed an Operating System for home controlling and monitoring. In addition to the creation of an Operating System, the authors also built a medium access protocol which facilitates the connection of devices to the operating system. This hardware is described as consisting of coordinators and devices. The coordinator sends and receives data from devices via infrared or power lines. The designed network could have up to 256 coordinators, each supporting up to 15 devices [5].

The architecture of the operating system consists of two layers - the Port layer and the Service Layer, as shown in figure 4.. The port layer provides interfaces between the Operating system and the hardware. Meanwhile, the Service layer is where applications are implemented. The Service layer can have up to 15 processes whose priorities are maintained by an array,

priorities, and a scheduler. Eight binary semaphores and message queues are used for synchronization. Additionally, the unified Medium access control protocol allows devices to send messages to one another via the coordinator.

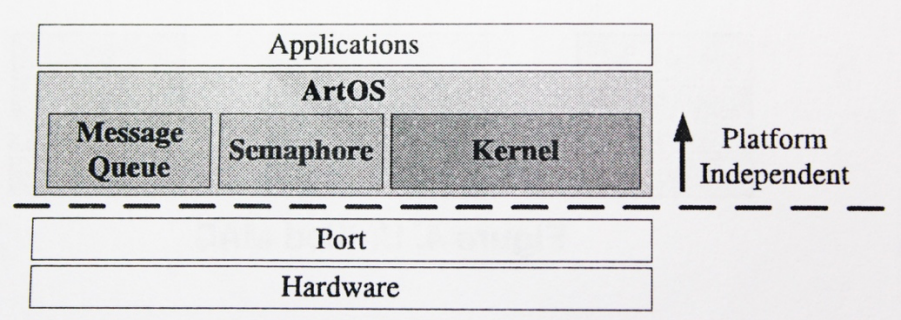

Figure 4

In 2017, Researchers from Wayne State University published a paper introducing ideas about an operating system named $EdgeOS_H$ that connects devices within the home using Edge Computing. It also acknowledged that such an Operating system has several requirements. It would require Internet of Things devices to be available – that is, internet connected devices that contain low-cost sensors and chips. It might also require reasonable adoption to be feasible. Edge Computing would ensure that most of the data processing occurs close to the devices, while reducing network load, decreasing latency, and increasing security.

The $EdgeOS_H$ has a communication layer, a Data Management Layer, Self-Management Layer, and the Programming Interface layer. The communication layer would connect to devices using a variety of protocols such as Wi-Fi, Bluetooth, ZigBee, or a cellular network. The Data Management layer brings all data into one database. The Self-Management layer is intended to register, maintain, and replace devices. Lastly, the Programming Interface Layer allows for the creation of software for users. A communications adapter would enable sending and receiving of data to or from devices through drivers. An event Hub would capture system events and communicate them either to the communication layer or up to the Programming Interface or Service levels. These hybrid-layered approaches to Operating System structure would ultimately allow for efficient construction and debugging. [6]

Manufacturers such as Crestron constantly monitor for security vulnerabilities. Crestron's support website periodically release firmware to address operating system

vulnerabilities such as OS Command Injections, improper access control, and improper elevation of privileges in Crestron terminal protocols. Crestron also allows for the creation of password protected accounts and groups to protect hardware and software objects. If this password protection were abstracted as an access matrix, the users would be the domains and connected devices would be the objects.

Manufacturers of voice activated assistants ensure security and privacy by not allowing third party applications on the device, data encryption, and providing users with security options. A report published by Symantec describes some of the security and protection challenges of voice assistants. The voice assistants are often linked to an owner's online shopping accounts but has not always had the protection and authorization capabilities of the centralized operating systems, so orders can be made by visitors, children, or even audio from a television. The paper describes a situation where a fast food commercial asked a device to explain one of its products, prompting any device that was near a television to reply with a Wikipedia description. In another situation, a character from a popular television shown asks the Google, Apple, and Amazon devices repeat expletive phrases. The devices on the show and in viewers' homes responded. Although these devices are always listening for their activation keywords, security is ensured as the only recorded information stored is that which is recorded after the keyword. To ensure communication security, recordings are encrypted then transmitted to the servers for processing requests, searches, or purchases [7].

The various types of home operating systems available today each have their merits. The centralized systems are always on. The Ad hoc systems offer ease of connectivity, and essentially transform the controlling mobile device into home's operating system. Although the voice assistants behave more like input devices to web-connected systems, the ad-hoc voice assistants offer similar connectivity with a voice interface. The centralized systems such as Crestron and Savant are similar to traditional computer operating systems as they have a dedicated central processing unit and are capable of having user profiles. The drawbacks of the centralized system are that they are often monolithic systems that have a limited number of devices that can be connected to them. The Ad hoc systems on the other hand, allow for more

connections, however they often do not offer cohesive and standardized access as each manufacturer and device may be using a different communication protocol or user interface. Neither approach seems to be the accepted standard, however the number of homes using the ad-hoc approach is growing.

Most home operating systems are designed to control lighting, temperature, security, and entertainment. Given the various other functions of homes, and their multitude of configurations, homes may ultimately benefit from having a network of operating systems, with coordinators being responsible for major functions, such as dining, productivity, social life, entertainment, or security. Devices specific to those functions could connect to the coordinator devices. Such a configuration might offer a modular alternative to the centralized or ad-hoc operating systems.

References

[1] Silberschatz, Abraham, Peter Baer Galvin, and Greg Gagne. *Operating Systems Concepts.* John Wiley & Sons, 2013.

[2] William H. Gates III, Nathan Myhrvold, Peter Rinearson. *The Road Ahead.* Viking Press, 1995.

[3] Takako Nonaka et al., "Embedded Server and Client System for Home Appliances on Real-Time Operating System", Ryukoku University, 2010.

[4]Predicting Our Future, "Episode 14: Your Home's Operating System & The Artificial Intelligence that will power it", January 24th, 2018, podcast, www.predictingourfuture.com/14-your-homes-Operating-System-the-Artificial-Intelligence-that-will-power-it/, accessed October 2018

[5] Chuan-Yue Yang et al., "Another Real Time Operating System and Unified MAC Protocol for Home Controlling and Monitoring", Department of Computer Science and Information Engineering, Graduate Institute of Networking and Multimedia, National Taiwan University 2006.

[6] Jie Cao et al., "EdgeOS$_{H:}$: A Home Operating System for Internet of Everything", Department of Computer Science, Wayne State University, 2017.

[7] Wue, Candid. ,"A guide to the security of voice-activated smart speakers", Symantec, 2017

Nadkarni, Prakash M., Lucila Ohno-Machado, and Wendy W. Chapman. "Natural Language Processing: an introduction." Yale University School of Medicine, University of California, San Diego School of Medicine, 2011.

Weingarten, Florian, Marco Blumendorf, and Sahin Albayrak. "Towards Multimodal Interaction in Smart Home Environments: The Home Operating System" Technical University of Berlin, 2010.