

Operating System Classification Around Data

Terry Yin

October 8, 2014

Abstract

Based on the different roles a computer system plays around the data, it will require different features from the operating system underneath. This paper provides three roles that the operating system might play, and the specific features might be needed from each role. One OS doesn't necessarily only play one role. Some OS might have the feature to support two or even all the three roles. These three roles around data are Data Renderer, Data Transporter and Data Processor.

1 Introduction

All the computer systems are used to do something with data. There are several different roles a computer system could play around the data, like some are used to render and collect data, some are used to transport data and some are used to compute or store the data. We can generalize some common requirements for the operating system from different roles. So I would roughly categorize the operating systems into **Data Renderers**, **Data Transporters** and **Data Processors**. One system might have more than one role.

2 The Current Classifications

My classification about the operating system around data is futurespective. So I have some assumptions about the future computer system, say, in the coming decade.

One assumption is that the processor speed and memory capacity will have less limitation by the size of the machine than today. One of the classic OS classifications is the **embedded system** (Brookshear 2011, p.114). Today it's getting harder and harder to distinguish an embedded OS from a non-embedded OS. I have a Raspberry Pi mini computer which is the size of, well, a slice of a 6 inch pie, but runs the desktop version Ubuntu, Edwards (2013). I used it to watch movies in the hotel rooms when I'm travelling. I certainly won't call it embedded system. There will still be OS that is closer to the hardware or has limited resource than the others. But as they are getting more and more insignificant, I won't use embedded as one of the main OS classification.

Another assumption is that the current user habits are subject to change. People are used to the way of working with their OS. Some users even developed the habit of installing third party anti-virus software on their OS. Software companies also helped their users to develop the habit of purchasing applications from online "App Stores". I would assume all these habits are subject to change. It won't happen overnight. But as long as the new technology serves the need better, I think the new user habit will be forced. Like the emerging **WebOS**, I think people will get used to run most of their applications in a browser without caring about the upgrade of the applications by themselves, and people will get used to put their data on the cloud.

Desktop System is another traditional OS category, which is actually a mix of several user needs. The main job of a desktop system is to interact with end user and render data. But sometimes people also expect to have more computing power with their desktop system so that they can do things like graphical design, run lightweight database, run peer-to-peer sharing application and test their development work. I would use my new classification to make it clearer.

Another traditional OS classification is **Real-time System** (Brookshear 2011, p.115).

$$RealTime \neq RealFast$$

The need for real-time OS is not because we need high performance system, McKenney (2008). They are often used in the situation that the system have little knowledge about the content of the data it's dealing with, and their job is to collect data and/or transfer the data. Thus, in my new classification, they are data transporters.

Distributed Operating System is another often used OS classification. I have seen at least two different scenarios for the application of the distributed systems. One is in the telecom equipments, where the distributed OS is mainly used to facilitate the network resources to provide transportation capacity and reliability. Another scenario is in the cloud computing, where the distributed OS is used to virtualized computing resource and scale up the computing capacity. So I would put the two different scenarios under Data Transporters and Data Processors respectively.

3 Operating System for Roles Around data



Data Renderer



Data Transporter



Data Processor

	Renderer	Transporter	Processor
Windows 8	good		ok
OS X	excellent		ok
Ubuntu Desktop	ok		ok
Ubuntu Server			great
Android	good		
VxWorks		good	good
Real-Time Linux		great	

3.1 Data Renderers

Some systems are designed to represent data to the user and/or collect data from the user. The desktop OSs Microsoft Windows, Apple OS X, the Linux implementation Ubuntu fall in this category. The mobile OS like iOS and Android also fall in this category.

These operating systems need to optimize for better end user experience, like having better battery endurance, better graphics and sound, and quick response to user requests. Usually, they need to support a

large variation of media hardware like displays, printers, input devices and cameras.



United States v. Microsoft Corp.

United States v. Microsoft Corporation is a United States law case, which was initiated in 1998 and settled in 2001, [Geier \(2001\)](#). Microsoft was accused of becoming a monopoly, and the main reason is to bundle its Internet Explorer, a web browser, with its operating system, the Microsoft Windows. “Microsoft’s Windows operating system is the sun in the solar system of the information economy,” as mentioned by [Geier \(2001\)](#). And when all the Windows user get an IE browser by default, it was considered not fair to the other browser makers. Microsoft was also questioned over whether it manipulated the Windows API to favor IE over other third party web browsers.



Chrome OS

Google announced the Chrome OS for the first time in 2009. Chrome OS is an open source operating system designed by Google based on the Linux Kernel, [Adee \(2010\)](#). Chrome OS is often called a “web OS”, because it’s main functionality is just a browser. The Chrome OS depends heavily on the cloud computing and Web applications. With the Web application provided by Google and other vendors, you can pretty much do anything the Microsoft Office application can do.

These two events in 10 years show the progress of people’s cognition of the operating system for the user end computer. Microsoft was accused of bundling OS with Internet browser, and now Internet browser becomes the core of the emerging OS.

Some people believe that Javascript is the new assembly language for the Web [Hanselman & Meijer \(2011\)](#). And the browser is the core feature of a client-side Web OS. We may see that in the future most user devices including laptop, netbook, handsets, wearables-devices, all run on OS that natively support HTML rendering and run JavaScript at the lower OS level.

3.2 Data Transporters

Some systems are designed to move data around, instead of processing them or rendering them. These include the mobile radio network, the switches, the media transmitters, and the home-based WiFi APs. These OSs are traditionally known as **real-time** or **embedded**, or sometimes, **distributed**. Examples are Wind River VxWorks and the Real-Time Linux.

These systems usually have limited knowledge with the data they are transferring. Their jobs are to best utilize the network resource, minimize the delay and maximize the throughput. So traditionally, they are “**real-time**” system. But as the capacity of the processor and other hardware keep increasing, my very personal opinion is the “real-time” factor in these OSs are becoming less and less significant. This is because the system performance will benefit from a simplified design when removing the hard requirements like a “real-time” framework from the design, [McKenney \(2008\)](#).

The type of the data being transported sometimes matters, e.g., streamed video, web content, chats. But, on the other hand, nowadays, TCP/IP networks have replaced most of the other networks. For example, the progress of 2G GSM, 3G WCDMA and 4G LTE in mobile network is the progress towards an all-IP network, [David et al. \(2009\)](#). So, it seems having a sound and extensive TCP/IP stack is a basic requirement for the OS in this classification.

I’m a bit hesitate about where to put the emerging **Internet of Things** system. Because a basic common feature of this kind of system is machine-to-machine communications, [Wikipedia \(2014\)](#). I would like to put it also under the Data Transporters.

3.3 Data Processors

Finally, someone need to consolidate, process, store and dispatch the data. These systems usually have strong computing power and sometimes high speed inter-connections. They tend to manage more computing or storage resources rather than other rich media resources. Examples are Microsoft Windows Server and Linux Server. The system that process data sometimes works in a distributed way to balance the load or increase the capacity (Brookshear 2011, p.113).

Today, people still quite often expect their desktop/laptop computers to also have strong computing power, like running complicated games, processing documentations or build software locally. So desktop operating systems like Microsoft Windows and Apple OS X also provide the ability or data processing.

Unlike the use of **Distributed System** in Data Transporters, the Data Processors often use a different set of algorithms on the distributed systems, like Map-Reduce, big data and cloud computing algorithms.

Another notable feature that distinguish the Data Processors from other operating system roles is that Data Processor is more sensitive with security, because the data processor is where the data reveal its meaning and get aggregated. It's the target for attackers to stealth information without permission. So usually data processors would have stronger authentication system and come with extra protections.

4 Conclusion

In terms of the computer system dealing with data, probably we need specialists rather than generalists. Some operating systems are designed to be specialized in certain roles around the data and some in other roles. Hopefully, these classifications will provide one more dimension to consider when you need to decide which operating to choose to solve your problem.

References

- Adee, S. (2010), 'Chrome the conqueror', *Spectrum, IEEE* **47**(1), 34–39.
- Brookshear, J. G. (2011), *Computer science: an overview*, Paul Muljadi.
- David, M.-S., Jose F, M., Jorge, C.-P., Daniel, C., Salvador, G., Narcís, C. et al. (2009), 'On the way towards fourth-generation mobile: 3gpp lte and lte-advanced', *EURASIP Journal on Wireless Communications and Networking* **2009**.
- Edwards, C. (2013), 'Not-so-humble raspberry pi gets big ideas', *Engineering & Technology* **8**(3), 30–33.
- Geier, M. (2001), 'United states v. microsoft corp.', *Berk. Tech. LJ* **16**, 297.
- Hanselman, S. & Meijer, E. (2011), 'Javascript is assembly language for the web: Semantic markup is dead! clean vs. machine-coded html'.
- McKenney, P. E. (2008), 'real time'vs.'real fast': How to choose?, in 'Ottawa Linux Symposium (July 2008), pp. v2', pp. 57–65.
- Wikipedia (2014), 'Internet of things — wikipedia, the free encyclopedia'. [Online; accessed 8-October-2014].
URL: http://en.wikipedia.org/w/index.php?title=Internet_of_Things&oldid=628660278