

# Futurespective Operating System Classification

Terry Yin

October 5, 2014

## Abstract

Basically, choosing which operating system depends on doing what kind of task.

Future spective.

Assumptions:

- Processor capacity and memory size is not limited by the size of the machine.
- The price is not a main issue as well.
- Patent is not an issue.
- The current user habit is subject to change.
- Network transformation is still slow than message bus in a computer, and the backbone of a distributed system that sharing a same rack.

Embedded is not a category. Real time is not a category.

For this Assignment, you will research the landscape of OSs (including distributed OSs) and analyse criteria with which to classify them.

To prepare for this Assignment

- Review your weekly Learning Resources with a focus on operating systems.
- Research the features of different operating systems.
- Reflect on the criteria that is of importance to you when you select an operating system.
- Identify at least three distinct criteria for classifying an operating system.

To complete this Assignment:

- Submit a paper in which you address the following:
- Summarise your three distinct criteria for classifying operating systems.
- Explain your motivation for suggesting each of the criteria, and how they are different from standard OS categorisations.
- Create an OS catalogue:
  - List the distinct criteria you identified.
  - Describe at least two examples of existing operating systems for each criterion.

Fully state and justify any choices, assumptions, or claims that you make using the suggested resources for this week and/or your own research. Cite and reference at least 3 sources (including diagrams) that support your arguments.

The OS Joke.

# 1 Roles Around data

## 1.1 Data Renderers/Receiver

Some systems are design 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. Examples are Wind River VxWorks and the Real-Time Linux.



### 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 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 part 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 Kernal, [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 shows the progress of people’s recongnition of the operating system for a user end computer. Microsoft was accused for 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 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.

## 1.2 Data Transferers

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 trditional known as **real-time** or **embedded**, or sometimes, **distributed**.

These systems usually have very little knowledge about the data they are transfering, their jobs are to best utilize the network resource and minimize the delay. So trditionally, 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 “real-time” from the design.

The type of the data will distinguish the requirement for the OS it’s running on, 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 a full IP network. So, it seems having a sound and extensive TCP/IP stack is a basic requirement for the OS in the area.

### 1.3 Data Processors

Finally, some one need to consolidate, process store and redispach the data. Examples are Windows Server and Linux Server.

## 2 Examples

	Renderer	Transferer	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	
CentOS		ok	ok

## 3 License

1. Who
2. How
3. Where

## References

Adee, S. (2010), ‘Chrome the conqueror’, *Spectrum, IEEE* **47**(1), 34–39.

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’.