Intro to Unix

OVERVIEW

With this hands-on seminar, I want to give techies who are familiar with Windows a guided start to dipping their toes into Unix. During our time together, I plan to glance at popular Windows concepts and tools and then compare them to their equivalents under Unix. Where possible, we'll look at both command-line and GUI versions of a tool.

I hope that when you leave this seminar, you will have acquired a taste of what I mean when I say "It's all the same", meaning that most operating systems are the same once you scratch the surface. With a brain wired to rapidly classify what we see as "us" or "them", "save" or "dangerous", humans like to make a fuss about differences … but when we calm down that primal part of our brain and take a more reasoned look … I claim that the differences melt away and the similarities dominate.

Caveat: I'm not an expert at Unix. In particular, I know next to nothing about Unix GUIs. I am, however, skilled at saying "I don't know" … so please ask questions, and let's explore together. Feel free to ignore what I'm saying and poke around on your own – if you find something interesting, I encourage you to bring the rest of the class' attention to it.

If you have questions, please ask them. I don't have a particular agenda or curriculum to complete – we'll finish when time is up – in the meantime, I would prefer to focus on what interests **you**, rather than on the arbitrary list of topics which I have compiled.

PRINCIPLES

These principles help me to understand the larger world into which Unix fits.

- Life is pain. Anyone who tells you otherwise is selling something.
- With authority comes responsibility.
- Age brings both maturity and baggage[1].
- The world is a big place.
- All choices carry pros and cons.

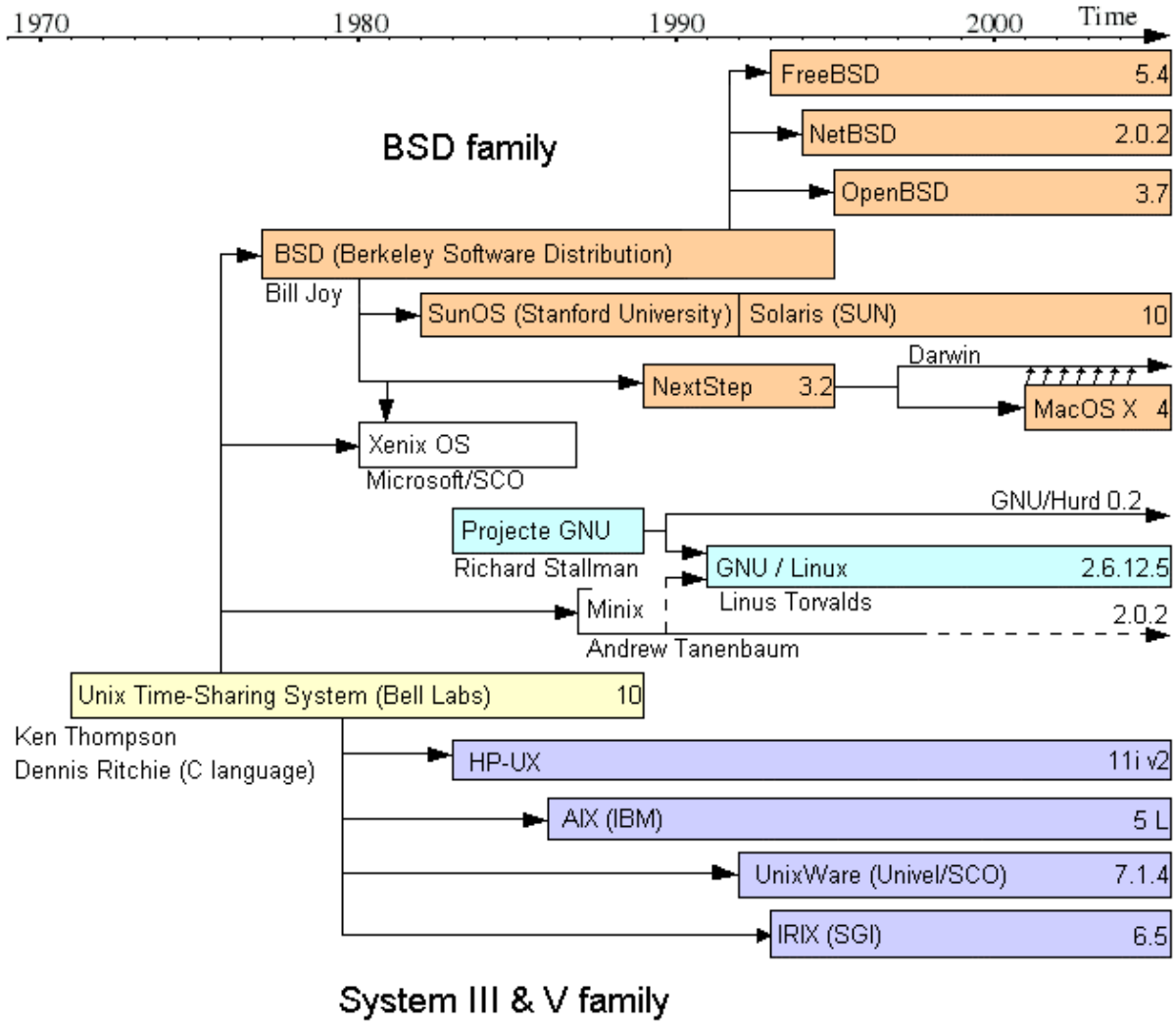
[1] "Maturity" defined here as self-transcendence, the ability to look past oneself, the ability to look at the larger picture. "Baggage" defined here as the scars and accompanying memories of pain from which we shrink when making future choices.

HISTORY

Ken Thompson and Denis Ritchie wrote the first version of Unix in 1969 on a PDP-7. Unix spent its first decade as a research OS[2] inside Bell Labs. In 1978, it started to appear in the university environment (the free versions of Unix from Berkeley) as *BSD 2.x*. In 1982, AT&T released the first commercial version of Unix as *System III*.

Historically, Berkeley and AT&T defined the two main flavors of Unix. Berkeley Software Distribution (BSD) from the University of California at Berkeley was one flavor; System V from AT&T was the other.

In the early 90s, Linus Torvalds instantiated a third Unix flavor, popularly called *Linux*. Torvalds owns the 'Linux' trademark, while The Open Group owns the 'Unix' trademark.



[2] Meaning, an operating system on which people who experiment with how to design operating systems could try out their choices.

This diagram makes the Unix family tree look small. It isn't. There are zillions of flavors of Unix, occupying various niches, from personal printers to telephone switches. Eric Levenez documents a few of them at http://www.levenez.com/unix/.

GENERALIZATIONS

Operating systems may look different on the outside – and we humans may exchange harsh words with each other over those differences – but inside, they are the same: they talk to disk, they juggle system tasks and user tasks, they accept input from a keyboard and return output on a monitor. As we walk through this session, we'll even see that many of the nitty-gritty steps involved in managing them are the same, though the precise names of the tools involved may vary.

- In Unix, the command-line came first; GUIs came later and are still developing. In Windows, the GUI came first; the command-line tools are still developing.

- The behavior of a Unix box is controlled by a horde of text configuration files, concentrated under the /etc directory. Behind the scenes, Unix GUIs used for configuration are merely reading and writing the appropriate text file. The behavior of a Windows box is controlled by a horde of 'keys' found within a binary file called the 'Registry' and edited using a dedicated GUI application: 'regedit'.

- Unix boxes tend to contain many different programs with overlapping functionality reflecting the baggage accumulated from decades of growth in an open enivironment. Different flavors of Unix will contain slightly different versions of even simple binaries like "ls", and multiple GUIs compete with one another for end-user affection. Windows boxes tend to host far fewer programs, and a particularly function is performed by fewer binaries … sometimes exactly one. While there exist multiple GUIs for Windows, most everyone uses the one which Microsoft includes on the CD.

- Unix utilities lean toward being small, single-function programs which can be chained together in complex ways; Windows utilities tend to be larger, multi-function applications which operate independently of one another.

- The Unix file system is case-sensitive, meaning that programs processing commands and passwords and filenames pay attention to case. The Windows file system is case-aware, but not case-sensitive. For example, if your current directory contains a file "memo.doc" and you save a new file as "Memo.doc", under Unix you will have two different files, while under Windows "Memo.doc" will replace "memo.doc".

- Unix has little in the way of naming conventions to distinguish between one type of file and another; use the 'file' command to guess the nature of a file. The Windows environment exhibits discipline around adding extensions to document names to indicate the contents of the file (e.g. memo.doc, notepad.exe, and notes.txt).

CULTURAL CONSIDERATIONS

- Unix was born as a research operating system[3] and grew up in the university environment, both cultures value transparency, flexibility, customizability, interoperability. If, at times, Unix feels like a kit car, that's because it is – its designers wanted to expose how it worked and make it easy for others to add or modify components. Windows was born into an environment driven by profit: the officers of Microsoft are legally obligated to make money for their stockholders. If, at times, Windows feels like a black box, realize that if Microsoft were to expose the details of what they are doing to their customers … they would also be exposing those details to their competitors, behavior which could land them into trouble with their stockholders.

- The Unix environment tends to be labeled with human names. For example, the documentation for a particular feature has the author's name on it, and the names of the people who wrote the code are widely publicized: the inventors of the operating system itself, the people who made significant changes to the kernel or who wrote new kernels, the people who added each new chunk of functionality, the creators of client/server features like file-sharing (NFS), name-to-address translation (DNS), and mail exchange (Sendmail). For the most part, these inventors are alive and well and continuing to add features to Unix. Windows, of course, has names behind it, too – it was written by people! However, I know of no way to discover those names.

COMPARISONS

Windows	Linux
Control Panels	YAST, ps
C:, D:, E:	file systems mounted under /
dir, cd, del, type	ls, cd, rm, cat (less)
edit	pico
Task Manager	Gnome System Monitor
Explorer/IExplorer	File system browsers
Safe Mode	Run levels
Registry (startup)	text configuration files (/etc/init.d scripts)
find	grep
Help	man
?	su/sudo

[3] An operating system employed to model how operating systems work and to experiment with alternate designs.