

The Value Line

What's an operating system worth? At various moments in history, companies have paid millions for OSes, and more for hardware to go with it. Today, as commodity PC hardware drives down the cost of workstations and servers, proprietary OSes and applications match and outstrip the cost of the systems they run on. And then there's Linux.

But what is Linux worth? The answer depends on whom and how you ask the question. On one hand, actual development costs for Linux were surely enormous – using formula's like Boehm' Constructive Cost Model (CoCoMo), building software of Linux size and complexity from scratch would cost at least US\$500 million, with the whole GNU/Linux OS running to billions of dollars.

On the other hand, acquiring the Linux kernel and much of the OS costs practically nothing – you can download it from kernel.org and myriad other locations. Or you can purchase a standard distribution from Red Hat, SuSE, Miracle, Red Flag or others distro suppliers. And they charge you an amount concomitant with their particular added value. That value comes in the form of packaging, integration, testing, quality assurance, and tools for installation, configuration, system management, development, and security.

"Added Value" is a marketing term, bandied about with little regard for its actual meaning. "Value-Added" begs the question, "added to what?" In this discussion, we respond "Added to Linux, of course", but what is Linux, anyway?

The kernel? The kernel plus libraries like glibc? Tools and utilities? There is no right answer for Linux, except that value lies above the Value Line; everything below the line is a commodity.

In the proprietary world, a commodity is defined as "someone else's software." With Linux and Open Source, the situation is not that different.

Divergence or forking below the value line wrecks havoc for both technical and commercial ends

Where the value line is drawn is usually determined by your place in the ecosystem. For chip and systems vendors, all of Linux, from kernel to application stack, is a commodity that helps fill sockets and move boxes. For ISVs (Independent Software Vendors) and for many open source project participants, Linux is a commodity platform on which they build and run their value-added applications. For distribution suppliers, the commodity is that shared technology base of Linux and dozens of other Open Source projects, with the Value Line drawn right beneath their brands.

This demarcation was the basis for efforts like United Linux, whose members sought to pool efforts and reduce costs (below the Value Line) by ceding responsibility to the consortium to supply a common platform and agreeing to add value only on top. With the release of LSB version 2.0, there is now some new consensus about what Linux should look like, at least (I would posit) below the Value Line.

How distros treat the Value Line supersedes individual business models, with implications for the entire Linux ecosystem. Enterprise adoption of Linux, while proceeding apace in the data centre, must still bridge a range of gaps before it achieves true mainstream status.

Looming large are challenges faced by commercial software vendors in supporting Linux as a platform. I'll not argue here about whether Linux should be accompanied by a 100 per cent open stack – the fact is that open equivalents do not exist for all elements of enterprise work loads and work flows.

When ISVs support Windows, Solaris, or other legacy OSes, they see reasonably well-defined APIs and functionality, and fairly well-constrained versions and variants for each. So, ISVs usually support no more than two versions/revisions of each OS. With Linux, by contrast, ISVs may need to target multiple Linux distributions. With the speed at which revisions occur, ISVs may also need to support multiple versions of each. One company I know is maintaining support for nine Linux variants and versions (vs. two for their

larger proprietary OS-based business).

"Value added" becomes tenuous when, regardless of real value, Linux-based platforms set themselves apart as a means to lock users and customers into a particular platform. I myself have been burned by distribution "value add". I used to work for a "lesser" distribution vendor, specialized in embedded deployment. In our niche, we had enough clout to gather a partner program with over 100 ISVs with our platform as "the standard". After several years of operation, we began playing with the "big boys" - enterprise ISVs, mostly database vendors. Mutual customers needed to know that critical applications and stackware ran on our Carrier Grade Linux just like it did on enterprise distributions – which (bless Linux) they basically did. That is, until the enterprise platform of choice began "adding value" by changing directory paths, and using newer (and incompatible) compiler and library versions. Our enterprise software partners came back to us with a simple message – we had to follow the enterprise distributions in their practices and conventions, no matter how non-standard or inappropriate for our business, because they were right and we were small.

Diversity engendered by Open Source creates choice and holds down costs of software acquisition. Divergence or forking below the value line wrecks havoc for both technical and commercial ends. If you can add value above the Value Line, what is your value-add, anyway?



Bill Weinberg