

Migrating from Linux to NetBSD

A Guide for OEMs



Many embedded OEMs, faced with harassment from alleged Linux copyright-holders and enforcement actions from free software partisans, are seeking alternatives to Linux for their next generation products. This guide addresses certain issues that may arise in migrating from Linux to NetBSD, one of the more prominent open source alternatives to Linux.

As explained in the white paper *The GPL in Embedded Systems*, NetBSD is free from Linux's GPL license, which often requires the disclosure of sensitive code as part of its requirements, and which has led to much of the recent litigation. As explained in the white paper *Linux or BSD: Which OS is Better for Embedded Systems*, the difference in licensing is one of the two primary reasons identified by OEMs for choosing NetBSD over Linux; platform support is the other. Readers interested in the GPL, in portability/platform support issues, or in other factors that differentiate Linux from NetBSD are referred to those documents. Here, we will address frequently asked questions by OEMs wishing to adopt NetBSD after previously using Linux.

Q: I would like to use NetBSD, but I am concerned about application support. I've heard that NetBSD can run in "Linux Emulation" mode, but is there a loss of performance running in that way? How efficiently do Linux applications run on NetBSD?

A: There is no difference in execution time between running Linux applications on NetBSD as compared with running natively on Linux. Depending on the application, some users may experience slightly longer startup time. However, once the application is running, there is no difference.

Linux applications simply use a different system call vector when the application is executed - a note in the ELF binary denotes that Linux system call numbers should be used. For every system call, if arguments to the system call need to be modified, this is done, and the same fixups happen on the way back out of the kernel. There is a minor amount of work to be done for some system calls, therefore, but not noticeable in practice.

In the past, because of the way that certain operations were carried out in the different kernels, NetBSD would actually be more efficient in some applications than native Linux. However, any performance gain is likely to be negligible.

Wasabi's own engineers have used the Linux toolchain, in NetBSD Linux emulation mode, to produce Linux binaries, which were themselves run using the Linux emulation on NetBSD. This was for a production web server, which used a third party Linux library to do a UK postcode/zipcode to address mapping. No problems were encountered.

Q: I've heard that NetBSD has its own C-library, and does not use GLIBC. But it appears from the NetBSD operating system documentation that the Linux emulation requires the installation of Linux libraries such as the GLIBC. How have Wasabi's customers dealt this issue, particularly in terms of licensing?

A: NetBSD has its own C library. It does not use GLIBC. Support for GLIBC and other software needed for Linux emulation is usually done by using the NetBSD packages collection, known as pkgsrc.

GLIBC is distributed subject to the LGPL – not the GPL. Thus, distributing GLIBC binaries requires distributing the source for GLIBC, but does not require distributing the source for any userland applications or for the kernel itself. The LGPL was developed precisely this purpose. The usual practice is to distribute the source for GLIBC along with the libraries. Wasabi has not had an instance where this posed any problem for one of its customers.

Q: What portions of the Wasabi NetBSD version fall under the GPL and/or LGPL?

A: No part of the kernel is under the GPL. This is the most important element for OEMs, who routinely modify the kernel or port it to new hardware architectures. Under the BSD license and Wasabi Certified NetBSD's licensing terms, those changes may be kept proprietary.

Some userland programs are distributed under the GPL. These are the GNU toolchain (gcc, gdb, assembler, linker, binutils, etc.) and a few other GNU utilities: awk, bc, cvs, diff, grep, groff, gzip, rcs, uucp. Any userland tools that are covered by the GPL (e.g., GCC or glibc) can be distributed separately or provided through other means.

Note that NetBSD 2.0 will ship with a version of nawk not subject to the GPL. NetBSD 2.0 will also ship with a BSD-licensed gzip, written by Wasabi engineer Matt Green.

Q: What are the terms for releasing source code of Wasabi Certified NetBSD? Suppose an OEM wishes to give source code to downstream integrators who would then make certain modifications for incorporation into our chips. Does Wasabi's Certified NetBSD license allow modifications to the kernel? Can OEMs then give the source code to their customers for possible further modification? Does the general BSD license speak to this issue?

A: Wasabi does issue source code licenses to OEMs in this position. Modifying the kernel is allowed, as is limited distribution of source to third parties.

The BSD license is deliberately silent regarding how developers can restrict code created on the basis of BSD-licensed software. Theoretically, it would be possible to simply take the NetBSD kernel, package it under some commercial name, and create a new license that would restrict its distribution. This was the intention of the license: unlike the GPL, developers are free to do with the code whatever they wish, including restricting its downstream distribution.

In the case of Wasabi Certified NetBSD, of course, the original BSD code has been extensively modified with proprietary extensions, optimizations for particular embedded uses, and benchmarked performance enhancements for embedded and networked applications.

Q: Are there any restrictions imposed by the BSD license? How do we know that we won't have the same problems we now have with the GPL?

A: As of its most recent revision, the BSD license contains exactly two restrictions on redistribution: the redistribution must include the text of the BSD license itself (including copyright information and disclaimers) and any accompanying documentation must do the same. Those are the only conditions placed upon redistribution. As long as they are met, distributors may use the code for any purpose, release the code to the public, restrict downstream distribution, or keep the code entirely secret.

The BSD license was created by University of California, Berkeley, legal professors with three intentions: allowing code to be used in any way developers wished, preserving credit for the original writers of the code, and eliminating threats of liability. Unlike the GPL, the BSD license has been tested in court, where it prevented AT&T from alleging ownership of BSD code – much the same as SCO is alleging ownership of Linux code today. When the AT&T suit was settled, in 1994, all allegedly controversial lines of code were removed from the BSD distributions, including NetBSD. Thus, NetBSD and its license have already been tested; they are litigation-proof in a way that Linux is not.

Jay Michaelson
Ian Taylor

(c) 2004 Wasabi Systems Inc. All rights reserved. This paper may not be sold or distributed without the permission of Wasabi Systems Inc. (www.wasabisystems.com). Citations and quotations from this document must include the copyright notice.