Honey, I shrunk the Linux system!

Jerry Epplin (June 27, 2001)

The onward march of embedded Linux

If you've followed the progress of Linux in embedded systems you've seen it evolve in stages. Originally, it was strictly a desktop operating system, with critics laughing at suggestions that it might be useful in embedded systems. Gradually, it progressed to where those same critics acknowledged its usefulness for *certain* restricted high-end embedded applications, but not for *real* embedded systems.

By now, we see Linux used in a growing variety of devices such as set-top boxes and PDAs. What's more, a recent readership survey from *Embedded Systems Programming* magazine reports that Linux has jumped to the number two spot in terms of consideration for future embedded system designs.

Even skeptics must now acknowledge that Linux won't disappear from the embedded market anytime soon!

One aspect of this progression is the appearance of a large number of single-board computers (SBCs) catering to developers wishing to use Linux inside embedded systems and smart devices of all shapes and sizes.

Linux cubed

Intrinsyc's CerfCube is one of the more interesting examples of these products. The Linux CerfCube comes complete with a Linux kernel and Apache web server, and sells for $379 at quantity one.

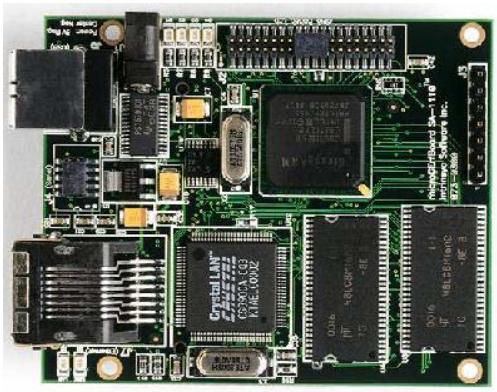
The CerfCube is basically a miniature Linux "box" consisting of a StrongARM-based SBC (called a CerfBoard) enclosed in a cast aluminum cube, three inches on a side. Do the math: with a CerfCube, you can fit a complete (networked) Linux system in just 0.015 cubic foot!

As shown in the photo below, the CerfBoard SBC forms the bottom face of the cube, leaving the cube mostly hollow. (Despite the plentiful extra space inside the cube, the unit is powered by a rather large external wall-mount power supply.)

One good reason for leaving the CerfBoard exposed at the bottom of the cube is to allow access to its CompactFlash slot, which can be outfitted with a variety of expansion functions. Because of the exposure of its internal electronics, the CerfCube comes with an antistatic bag and a wrist strap for handling.

Honey, I shrunk the SBC

The tiny (2.2 x 2.4 in.) CerfBoard has a 192 MHz StrongARM 1110 with 32 MB of RAM and 16 MB of nonvolatile flash memory. Like much larger SBCs, the CerfBoard is packed with I/O -- a 10 Mb Ethernet controller, a CompactFlash (type II) interface, three serial ports, 16 general-purpose I/O lines, and a USB port.



Like many (but not all) developer-oriented SBCs, the CerfBoard is also excruciatingly well documented, with all of its hardware described in a design document, and good instructions on modifying the system software.

Developing with the CerfCube

It's important to realize that the CerfCube is not meant to be an end product, as shipped. Rather, it's intended to serve as a conveient (and cute) platform to make it easy for original equipment manufacturers (OEMs) to evaluate Intrinsyc's CerfBoard SBC, develop products based on it.

An initial look at the CerfBoard's development model shows it to have a simple and straightforward design, providing an easy-to-use environment while still providing access to the lower-level details.

The CerfCube system configuration model at first seems chaotic, but one quickly becomes acclimated to the environment. Setting up the host for cross-development is an annoying process -- it consists of manually untarring nine files and running some scripts. This installs the source code for the target system, the StrongARM cross-compiler and binutils (for an x86 cross-development host), and various build scripts and tools.

The installation process deposits its files throughout the development host, and there is no simple way to uninstall them. The installation process could be improved by providing RPMs or .deb files; or better, just make the installation one big tar file that installs all its files in a user-selected location. Intrinsyc reports that they plan to provide uninstallation scripts with the next release of the toolchain.

Making mod's

Intrinsyc's documentation for modifying the system software is clear and well laid out, and they provide helpful scripts to simplify the process. In general, you simply modify the default sources, run a script to rebuild the image, and download to the CerfCube using TFPT or BOOTP.

You need a serial connection to the CerfCube's monitor program to initiate image transfers, and an Ethernet connection to transfer the data. This can be done through a hub, or by using the Ethernet crossover cable which Intrinsyc helpfully provides, which lets you make a direct connection between your desktop computer and the CerfCube.

Modifying the kernel is a matter of making the changes in the source directory, running a script to build a new zImage file, and downloading the image to the CerfCube using either TFTP or BOOTP. This process is well documented, and works well. The kernel provided with the CerfCube is 2.4.0, with Russell King's ARM patches and Nicolas Pitre's StrongARM patches applied.

Software organization and packaging system

The system has two memory-based filesystems. *rootdisk* is uncompressed from flash into RAM and mounted as /. Packages can be added to this filesystem fairly easily; again, Intrinsyc has provided helpful scripts to accomplish this, and the process is well documented. *usrdisk* is a flash-based filesystem that is mounted read-only directly as /usr. Modifying this filesystem is a simple matter of remounting the filesystem read-write, making the changes, and remounting read-only.

Intrinsyc uses a simple but functional packaging system in some ways similar to that used by Midori Linux. The unmodified source tarfile is provided for each package, along with a patch file with CerfCube-specific modifications, and a script to build the whole thing. These three files are placed in a package-specific directory. Thus, for example, the joe-2.8.color.5.dir directory contains the unmodified joe-2.8.color.5.tar.gz, a patch file called diff-joe-2.8.color.5.dir, and a script called 'build'. Building and installing new or modified packages is a matter of running some scripts using a well-documented sequence.

Conclusions

Intrinsyc has put together a very nice package with their Linux distribution for the CerfCube. With the exception of the chaotic and currently hard-to-reverse installation process, the package is well designed, with helpful scripts and good documentation allowing you to focus on your application rather than the environment. The software is also structured in a simple and intuitive way, so you can easily figure out how to accomplish anything that is not covered by the generally complete documentation.

Linux, with its extensive and mature networking capabilities, is an ideal OS for specialized network appliances. Expect the CerfBoard SBC -- with its tiny size, built-in NIC, flexible expansion, and good support for Linux -- to be a success in

this market as well as in more traditional SBC markets such as industrial control applications where Ethernet connectivity is required.

Author's bio: Jerry Epplin has written embedded software for the past fifteen years, primarily for medical devices. He can be reached at jerry@linuxdevices.com

---

Talk back! Do you have comments or questions on this article? talkback here