The BeOS BIBLE

SCOT HACKER

WITH HENRY BORTMAN AND CHRIS HERBORTH

Peachpit Press

The BeOS Bible

by Scot Hacker

with Henry Bortman and Chris Herborth

Peachpit Press
1249 Eighth Street
Berkeley, CA 94710
(800) 283-9444
(510) 524-2178
(510) 524-2221 (fax)

Find us on the World Wide Web at: `www.peachpit.com`

Peachpit Press is a division of Addison Wesley Longman

Notice of liability
The information in this chapter is distributed on an "As is" basis, without warranty. While every precaution has been taken in the preparation of this chapter, neither the author nor Peachpit Press shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this chapter or by the computer software and hardware products described herein.

ISBN: 0-201-35377-6

The Future

In this Chapter

The term “final code” is an oxymoron. No software is ever finished, and there’s always something that can be done better. When you’re talking about the 1.5 million lines of code in BeOS (actually an incredibly small codebase, compared to, say, the 30+ million lines of code in Windows NT), it’s inevitable that there will always be improvements that can be made. Improving BeOS is the reason dozens of Be engineers get up in the morning. It’s all about the future—how to meet it, how to beat it, and how to build the best possible operating system for the hardware that consumers can get their hands on easily. This chapter is all about the future of BeOS: what new capabilities, software, hardware, and OS concepts you can expect to see in the coming years.

Whence from Here?

Time doesn’t stand still, and the world’s technology evolves more quickly than any single company can keep up with (witness Microsoft missing out on the Internet boat for a few years). Making sure that BeOS continues to be the ultimate fine-tuned desktop and workstation operating system is a constant challenge for Be’s engineers. Fortunately, the rapid development cycles possible on BeOS mean that new releases can be offered frequently, accommodating not just new ideas and concepts, but hot new hardware as well. To date, BeOS users have had access to major updates roughly twice a year, with interim releases and bug fixes being made available at sporadic intervals in between. These updates aren’t just a matter of a nip and tuck here or a new I/O card driver there, either. In most cases, they’ve represented major, quantifiable leaps forward in speed, usability, and “feature completeness.”

This book made it out the door just after the endof the R4 beta cycle, and by the time you read this, Release 5 should be just around the corner. We’ll be posting updates to some of the material in this book at **www.beosbible.com,** where we’ll keep you abreast of any significant changes that affect the material you read here. Because Be was completely “nose to the grindstone” working on R4, they didn’t yet have a concrete road map to projected features for releases 6 or 7. However, based on what we know of the system today and early statements Be has made about R5, there are a few areas around which we can speculate and project with various degrees of certainty.

None of the projected hypothetical or real features in this chapter represent a commitment by Be or third parties to ship any particular software. This chapter is speculative in nature.

Release 5 Roadmap

Release 5 of BeOS is scheduled for shipment in early summer of 1999. Table 1 lists the features that were scheduled for inclusion as of November, 1998. Again, nothing in this list represents a commitment to ship specific features.

Table 1

Technology	Notes
NTFS support	Read and write support for Windows NT's NTFS filesystem from within BeOS. All NTFS volumes will appear normally in the Tracker alongside other mounted system volumes.
BFS Windows support	The ability to access BeOS partitions from within Windows 95, 98, and NT.
IEEE-1394 support	IEEE-1394 is the technical name for FireWire, a protocol similar to but competitive with USB. FireWire supports fewer devices on a chain than USB, but with much higher throughputs. As a result, it's suitable as a replacement for SCSI chains and can handle high-bandwidth input/output streams from digital camcorders, satellite dishes, and the like. See the sidebar *USB versus FireWire*.
Digital video, DV codecs, MPEG-2	Real-time encoding and decoding of these codec formats, all centering around the "convergence" industry, bringing television, computing, and digital A/V editing to the home.
Laptops	Support for a broad array of laptops. Most importantly, extensive power management support, which differs from power management at the desktop level and is key to "true" laptop support.
PC Card (PCMCIA) and card bus	Support for peripheral devices such as PC Card modems and hard drives.
OpenGL hardware acceleration	Many modern video cards include acceleration chips specifically designed to handle OpenGL renderings and modeling. Up until R5, these additional chips have been ignored.
Endian-neutral BFS support	The ability to read BFS disks created on PowerPC from the Intel side and vice versa. Note that this effort could be curtailed by potentially changing plans for BeOS/PowerPC in general.
Spatialized audio	Playing Dolby surround sound and Dolby Pro Logic audio formats. According to one engineer, "We might just invent a data type for streaming spatial information that could go parallel to audio information and then have plug-ins."
Multiple monitors	Multiple monitors will either split the available workspaces amongst themselves or each get 32 workspaces to themselves. Giant desktops à la MacOS are also a possibility.
Automated system updates	BeDepot will be revamped completely and will make it possible for OS components to be updated "on the fly." BeDepot will be trolled for out-of-date system components; new drivers, libraries, etc. can be installed transparently in the background. The merging of BeDepot with BeWare will begin prior to R5, with significant enhancements to BeDepot's online commerce functionality later.

Planned BeOS R5 features as of November, 1998. Release 5 should expand significantly on support for laptops, peripheral devices, multimedia formats, and more.

USB versus FireWire

While Macintosh users have long enjoyed true plug-and-play functionality for peripheral devices and I/O cards, the same concept as introduced in Microsoft's Windows 95 has often been referred to by critics as "plug and pray" due to its spotty success rate. When it works, it works excellently. When it doesn't, the poor user is thrown back on the arcane method of configuring IRQs and DMAs by hand, manipulating jumpers, rebooting, reinstalling drivers, rebooting, cursing, rebooting, and sometimes giving up. You can't blame Microsoft for trying—the peripheral problem in the PC world is a result of unfortunately chaotic hardware evolution in the x86 space. The real problem lies with the unruly state of cards, buses, specifications, and the way they all talk to the rest of the system, a messy situation that affects all x86-based operating systems equally. BeOS may deal with the problem much more elegantly than does Windows, but the basic problem stemming from a limited array of available IRQs in an environment where jumpered and non-jumpered cards lay side by side still remains.

The industry's attempt to rid both PC owners and OS vendors of these brain-dead bugaboos once and for all is the Universal Serial Bus, or USB, specification. USB addresses all of the most common peripheral hardware problems at once by letting users plug up to 127 devices into the same chain without worrying about details like IRQs and drivers. The need to stuff a computer case full of special cards to handle many new devices is diminished. The tangle of wires behind your computer is reduced to a single chain. The need to have separate power supplies for your Zip drive, scanner, modem, and everything else goes away.

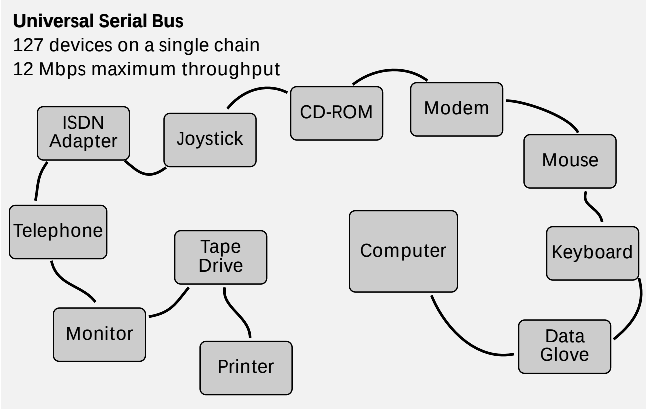

Figure 1

Universal Serial Bus offers a global plug-and-play solution for the connection of any type of device, from mice and keyboards to scanners and monitors, joysticks and data gloves to tape and floppy drives.

Pretty much every PC motherboard shipped after late 1997 includes a USB port, though the list of available USB peripherals is still somewhat short. Ultimately, USB-compatible devices will include monitors, audio devices, telephones, modems, keyboards, mice, CD-ROM drives, joysticks, tape and floppy drives, ISDN terminal adapters, scanners and printers, MPEG-2 video products, data gloves, and digitizers. USB devices can be plugged into the chain and recognized immediately—they're ready to use as soon as they're plugged in, no reboot required.

Sound like a panacea for the industry? Almost, but there a few significant entries missing from the list above: high-speed hard drives, video cameras, VCRs, and televisions. Oops. That's because, despite USB's many advantages, it's limited to a relatively paltry 12 megabits per second throughput. That's plenty for all the low- to medium-throughput devices listed above, but not

USB versus FireWire *(continued)*

enough for high-performance hard drives or serious video throughput. That's where USB's competition steps in.

IEEE-1394, or FireWire, is a specification that is similar to USB but has some significant differences. FireWire is limited to half the number of devices a USB chain supports, but has the advantage of radically higher throughput—a FireWire chain can support up to 400 megabits per second of raw throughput, which is more than enough to accommodate serious disk I/O and digital video needs. Originally developed by Apple but released to the industry as an open specification, FireWire is "isochronous," meaning that it transmits its data without latency—a critical aspect of juggling digital video (and critical to working well with BeOS's many real-time qualities). Combine this kind of throughput with BeOS's native real-time capabilities and you've got a real contender.

FireWire also offers us the opportunity to ditch our persnickety SCSI chains for good. FireWire is faster than SCSI, supports more devices, and does away with SCSI's configuration headaches. Drop a FireWire disk into your chain and it's available to the operating system—end of story.

However, just because the FireWire specification was originally created by Apple doesn't mean it'll only find its way into Macintosh hardware. The standard is backed by companies like Adaptec, Advanced Micro Devices, AT&T, Cirrus Logic, IBM, Lexmark, Microsoft, Molex, National Semiconductor, NCR, Philips, Seagate, Skipstone, and Texas Instruments. Trouble is, USB is on motherboards now, while FireWire is not. Chances are, consumers will get USB support with their machines by default; if they want FireWire's advanced speed they'll have to purchase a separate adapter card until board manufacturers can be convinced to build FireWire into their systems from the start.

Clearly, BeOS will need to support both standards—USB because customers will be buying USB peripherals, and FireWire because it fits so perfectly with the digital-content creation market. Support for both bus types is scheduled for R5, but once again, this is not a promise.

New Architectures

If BeOS is so portable, can we expect to see it running on other types of computers soon? Mostly no … and a little bit of yes.

When Will BeOS Run on Platform X?

A question frequently asked in Be-related newsgroups and mailing lists is whether BeOS will be ported to run on <insert your favorite architecture here>. For example, many people would like to see BeOS running on Alpha-based workstations, Sun SparcStations, or Silicon Graphics O2 graphics machines. The question is a natural one, since BeOS has run on four different computer architectures already (Hobbit, BeBox, PowerMac, and Intel), amply demonstrating how portable the system is.

However, there are a couple of important things to keep in mind here. For one, just because BeOS is easy to port doesn't mean that such a project would be a piece of cake. Yes, the initial port from the BeBox to the PowerMac practically occurred over the course of a weekend, but that was just the initial port—making everything work *properly* was another matter altogether. Then there was the small matter of critical differences between various Macintosh motherboards. While the CPUs remained pretty much the same from model to model within a given class of machine, custom chips on the motherboard were different for almost every single model, which meant that Be had to do a ton of work to get BeOS working on as many PowerMacs as they could.

Of course, the Intel port presented similar challenges. While the x86 boot sequence is publicly documented and pretty much identical from one machine to the next, Be had to deal instead with the huge array of video cards, CD-ROM drives, sound cards, network cards, etcetera. To undertake similar challenges on yet another platform would be suicide—Be isn't Microsoft or Apple, and doesn't have the kind of resources it takes to dedicate a team to facing these or similar hardware hurdles on other platforms.

Secondly, BeOS was a lot younger and simpler when those early ports were taking shape. BeOS may contain only a small fraction of the system code found in MacOS or Windows, but if you compared the codebase in DR7 or DR9 to the codebase in R5, you'd find that the system has grown immensely. Not that the Be engineers have waned in their commitment to keeping the system light on its feet, but in order to deliver the features that users demand, the system's complexity must grow almost by definition. You want to boot from SCSI drives on the Intel side? You have to add code to the kernel. You want task-switching with hotkeys? Add a few more lines to the application server. You want spatialized audio? Grow the Media Kit. Finding a balance between the contradictory demands of staying clean and simple on one hand and offering everything the other systems do (and then some) on the other is a continual battle. Sure enough, the codebase grows in size and complexity as time passes. It's unavoidable, and makes the prospect of a port all the more daunting with every passing release.

Third, managing the codebase for multiple platforms is a major operation, requiring sophisticated tracking systems and painstaking documentation of every little change to ensure that the platform remains identical regardless of the machine you're running it on. As anyone watching Be throughout 1998 could tell you, just supporting PowerPC and Intel at the same time was a herculean task for Be; adding a third or fourth platform to the source tree would likely break the camel's back. While it's true that the codebase stays the same between various platforms and that only the boot code, kernel, and drivers change significantly from one platform to the next, the reality of the situation is that maintaining a cross-platform source tree is a somewhat messy job.

But the fourth—and most important—reason why you're not likely to see BeOS running on other architectures in the foreseeable future is one of simple economics, and involves the requirements of running a sane business. You don't need a scientific calculator to figure out what percentage of potential users out there are running Alpha-based machines compared to the percentage of users who are running stock Intel or x86 clone hardware. Granted, the proportion of unique platforms in Be's target high-end graphics and multimedia markets is higher than that in the office suite computing space, but it's still not nearly high enough to make a sound business case for another port. And in the high-end CAD, 3D modeling, animation, and design fields, Windows NT has encroached strongly onto territory once owned by the likes of Sun and Silicon Graphics, so there are a lot more stock Intel machines in Hollywood and Silicon Alley graphics studios than there once were. These machines are perfect candidates for BeOS partitions. It simply doesn't make sense for Be to move toward obscure architectures when the world's most popular architecture is doing such a great job of coming to Be.

Bastard Child All of that said, the *possibility* for a port to another hardware architecture is certainly there. It's easy to imagine someone like SGI hypothetically wanting Be to bring the OS to their hardware and Be hypothetically responding, "You give us a check and we'll give you the source code—you port it." Unfortunately, a scenario like that would mean a splintering of the sacred cross-platform solidarity of BeOS code—a bastard child, if you will. On the other hand, it would also mean you'd be seeing BeOS-generated special effects in more summer blockbusters, and that *can't* be a bad thing.

Evolutionary Designs I think it's safe to say that Be will endeavor to keep up with the latest and greatest chip designs and motherboard revisions appearing in the consumer market. Intel's 1,000 MHz "Merced" chip promises a brand-new branch in the x86 family tree. You've probably heard for years that the CISC architecture is "on its last legs," "gasping for breath," "can't possibly go any faster." Well, they've been saying that for five years, and x86 chips are ten times faster than they were then. Nevertheless, there are a number of fundamental problems with the basic CISC architecture that *do* limit its growth potential. Without going into a chip design discussion, suffice it to say that Merced throws most of those problems away and gears up for the truly high-bandwidth future—just as BeOS has done. It's a marriage made in heaven.

CISC stands for Complex Instruction Set Computer, and is more or less a philosophy of CPU design used in x86-compatible processors. It stands in contrast to RISC, or Reduced Instruction Set Computer, which is the design philosophy behind PowerPC processors. In reality, both chips borrow design aspects from one another.

Like most new high-end chip releases, Merced CPUs will appear first in the server market (probably around summer 2000), then trickle slowly but surely down into consumer machines when the price comes down to the mortal realm. Be hasn't made any announcements about Merced chips (how could they? No one but Intel engineers has played with them), but I'll wager that Be will be enthusiastically Merced-compatible.

Meanwhile, transistors keep getting faster and more efficient. IBM recently announced plans to build chips wired with copper, rather than the traditional aluminum. Copper is a better conductor than aluminum and can be molded into thinner wires, which in turn lets chip manufacturers pack more transistors onto a single chip. Soon after IBM's announcement, several other chip vendors—including Intel—announced plans to release similar chips.

As clock cycles continue to rise, current memory and bus speeds will become major bottlenecks, so we can expect fundamental rethinking of those architectures at the same time. Peripheral connection standards like FireWire will go a long way toward keeping your external devices and storage media humming along in step, and BFS has already shown itself to be excellent at milking as much performance out of hard drives as possible.

More than Eight is Enough The original logic of the BeBox design—that it makes more sense to have a bunch of cheaper processors working in concert than it does to put all your clock cycles in one expensive basket—still makes a lot of sense. Do the math: At this writing, you can get 200 MHz CPUs for around $100 each. Four of those running in a system that knows how to distribute the workload efficiently would give you practically 800 MHz of desktop power for $400, when you can't buy an 800 MHz processor today at any price. However, the vast majority of operating systems installed on the world's desktops don't know how to communicate very well with multiple processors.

Windows 98 and MacOS 8.5, in fact, completely ignore extra processors (though applications can be specifically coded to take advantage of them). That pretty much leaves Windows NT and Linux to carry the multiproc market, and both of them suffer from serious impediments to efficient symmetric multiprocessing (SMP): NT wastes too many clock cycles in multiprocessor configurations, and Linux's SMP implementation is rather coarse (not to mention the fact that most Linux software isn't as finely multithreaded as similar BeOS software).

Of course, you know that BeOS handles SMP elegantly and automatically. When and if BeOS becomes popular enough to drive the multiprocessor market by itself, we'll all cheer. Until then, it's up to Windows NT 5, MacOS X, and future versions of Linux to improve their SMP behavior. Of course they ultimately will, because one processor per person isn't enough. And when they get really good at it, the multiproc market will explode and we'll all have to install seatbelts in our computers.

Using two or four processors is one thing, but the question is, when will consumer machines begin to ship with eight or more CPUs? As you recall from Chapter 1, BeOS can theoretically work its magic on any number of processors, but an upper limit of eight is in place for now. That limit can be removed by Be simply by uncommenting a few lines of system code, but motherboard and bus designs will have to take another form (the overhead required to juggle instructions between that many processors becomes hideously complex). In any case, BeOS will likely be more ready for that day than any other operating system in the mainstream market.

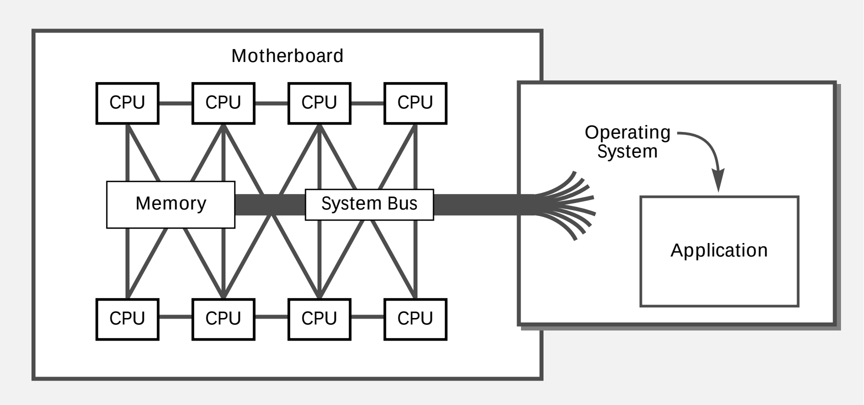

Figure 2

While BeOS is theoretically capable of handling any number of processors simultaneously, current motherboard and bus architecture limitations make the task of dealing with more than eight processors enormously complex, as each CPU needs to stay in touch with what the other CPUs are doing—a problem of logic and design inherent in the hardware world, not in BeOS itself.

Despite the state of the consumer market, there are military and government computers out there with *thousands* of CPUs running in parallel. These machines run on custom-designed motherboards with custom-designed operating systems and, naturally, cost millions of dollars. Whether and when massively parallel computing will filter down to the consumer market is anyone's guess, although some groups are running interesting experiments with "clustering"—roping hundreds of machines together in parallel to build a sort of pseudo-supercomputer. While few of us have hundreds of computers sitting around in our garages and attics to dedicate to cluster computing, the possibilities are fascinating. See Chapter 10, *System Tools and Utilities*, to find out how you can participate in distributed computing projects on BeOS.

While *Wired* magazine speculates about the future of chemical-based computers and lightning-fast RAM made of blue-green algae, you might want to chase more humble pursuits, like stuffing pennies into a piggy bank for your first quad-Merced machine.

The Future of BeOS on the PowerPC

Since the introduction of BeOS for Intel Architecture, speculation in the BeOS community has run high regarding Be's plans for continued support of BeOS on the PowerPC side. Initially, however, this speculation was based solely on numbers—everyone knows how much more potential revenue can be garnered from the Intel market, not to mention the additional management work required to maintain the codebase for multiple versions of the system simultaneously.

It wasn't until months later, when Apple declined to release technical specifications for the motherboards shipping with their new G3 processors, that this kind of speculation started to gain serious traction. The equation, it appeared, was quite simple: If Apple owns the entire PowerPC market and also refuses to work with OS vendors such as Be to add value to Apple hardware, all roads are blocked; there's simply nowhere for BeOS on PowerPC to turn.

Intel had come to Be, sending in a team of engineers to assist in the port of BeOS to their hardware, while Apple was curling up into a solipsistic little ball and refusing to dance. Many members of the Be community were up in arms about this turn of events. After all, they argued, PPC is where BeOS got its start in the public sphere (the original Hobbit processor design was never made public). The original BeBox had been PowerPC-based. Be had previously put all of their eggs in the PowerPC basket, and were now turning their backs on this market—the same market many associated with the creativity crowd (Be's target). In addition, all of BeOS's early adopters were about to be left high and dry.

The only thing keeping BeOS running on PowerPC Architecture at all, it seemed, was a promise made by CEO Jean-Louis Gassee after the death of the BeBox: He had publicly declared that the BeBox would be supported by Be, Inc. for a period of three years. As long as the BeBox was supported, PowerMacs would be supported along with it.

The Future of BeOS on the PowerPC (continued)

To complicate matters further, astute users noted that the Linux community had already succeeded in making their operating system run on G3-based PowerMacs by reverse-engineering the new motherboards. If Linux could run on G3s, why not BeOS? The answer given by other users was that Linux users are hackers. With no corporate representation, you can get away with reverse-engineering specs, but a company cannot base a business on the hacker's ethic. Be, curiously enough, was silent on the matter, saying nothing in favor of nor against these speculations.

While all of this was going down, an almost predictable phenomenon appeared on the horizon: Significant BeOS developers were choosing not to compile PPC versions of their applications. The first company to go public with this information was RoDesign, who make the popular BackRow and RoColor applications for Web developers (see page 732 in Chapter 13, *Graphics Applications*). While disappointing, none of this came as a total shock to anyone. Given that Be had sold more copies of BeOS for Intel in its first week of release than it ever had sold copies for PowerPC, the numbers had told the story long before these announcements were made.

On the other hand, more interesting news began to arise from the Apple camp. After years of steady decline and virtual acknowledgement by all but the staunchest PPC supporters that their platform was indeed doomed for good, Steve Jobs' closed-door policies were beginning to have an apparently positive effect on the Macintosh purse. Profits were up, the iMac appeared, and the usual gloom and doom over Apple's situation did yet another about-face in the press.

Be remained (and as of this writing, remains) diplomatically silent on the whole issue. Never once have they publicly stated that days are numbered for BeOS on PPC. Perhaps that will change by the time you read this, but for now, BeOS is still officially a cross-platform operating system. Personally, I support and encourage the notion of an operating system that doesn't care what hardware it's running on—biological diversity is good for all ecosystems—but I also fully understand that Be has a business to run.

Back to Earth on PIOS One

If prospects seem glum for BeOS on PowerPC, there's another architecture waiting in the wings that may just see the light of day by the time you read this. Since the death of the BeBox in April, 1997, the Be community has buzzed over what many have hoped could become its replacement: a PowerPC-based machine called the PIOS One from German manufacturer PIOS (**`http://www.pios.de`**). This unique company staunchly refused to roll over and play dead when all hopes for the long-awaited PowerPC Platform (or PPCP) spec (formerly known as CHRP, which in turn was formerly known as PReP) were dashed on the rocks by the companies behind it (the Apple/IBM/Motorola alliance, or AIM). CHRP had been intended to merge the technical superiority of the RISC CPU with the operating-system agnosticism and easy upgradeability the Intel community enjoys. Later, Apple came back to the table with an amended spec including some Mac-specific technology such as ADB, and called the new spec PPCP. PPCP machines would have supported MacOS while simultaneously loosening Apple's stranglehold on the RISC market with a nonproprietary design capable of multibooting MacOS, BeOS, Linux, and potentially other systems, without the boot ROM headaches facing alternative operating systems under Apple's then-current hardware design.

PIOS's project manager Dave Haynie had this to say about this period in CHRP's history:

"Everyone had considered MacOS vital to the launch of CHRP. We didn't. The original PIOS One was launched as something of a next-generation BeBox. I came to it with ten years of experience designing the high-end Amiga systems at Commodore, and I studied what Joe Palmer did on the BeBox, which oddly enough was very much the direction I was trying to move Commodore to, before they bought the farm. In early '97, I met with Apple, and they convinced me to support CHRP, the MacOS, motherhood, and apple pie. If this were a classic tragedy, that would have been the mistake that has the whole castle dead by the end of Act III. I did the second redesign last fall and winter, based on where we had to point things (multiplatform, even) with the loss of MacOS as a possible market. [PIOS] got a better system out of the deal, but we've since been plagued with little things like 'how do we pay for all of this with the MacClone business going away Real Soon Now.'

"CHRP was clearly too good to succeed. A market 'standard' is supposed to be someone's proprietary design, usually a crappy one, never intended for the greater life it enjoys as an industry standard. I won't claim CHRP was perfect, but it's way beyond the PowerMac and PCs in basic architecture. Which means very little the first time around, but over time, it has a great impact on what designers like me can put under the hood without the kind of kludges you find in the PC industry. The real promise of CHRP was that it made a better, much more flexible kind of personal computer, naturally centered around PowerPC, a better, more flexible kind of CPU. We're still going at it, jaws locked on the thing like some kind of pit bull terrier."

Like the BeBox and the BeOS itself, the PIOS One takes the best from the PPC and Intel worlds—its CPUs are RISC, but its I/O cards (sound, video, networking) all come from the Intel space. Uniquely, though, these machines can *also* utilize a few hardware standards from the Mac world, such as ADB mice and keyboards and LocalTalk networking. Users get more choices, more competitive prices, more expandability, and more performance. But best of all, the PIOS One is designed to rock as a BeOS workstation.

If you've ever heard an ex- (or current) Amiga-head ranting about how great that platform was in its time, you'll appreciate the fact that PIOS is run almost exclusively by former Amiga movers and shakers. At one point in time, some people called the BeBox "the next Amiga." Now they give the PIOS One the same accolades. The icing on the cake? This machine—if it ever sees the light of day—should only cost around a grand.

Unfortunately, after a year of construction and community buzz, PIOS One machines haven't yet shipped to the public. As the PIOS One was finally approaching readiness, PPC support in BeOS seemed to have question marks circling its head. In addition, the unusual hardware design of the PIOS One will require some close QA work with Be to ensure compatibility, primarily in the boot code arena. PIOS expects to be shipping BeOS-ready machines by the time you read this, but the company has faced an uphill battle gaining traction in the quickly changing PowerPC market. As you'll discover in the sidebar *Interview with Dave Haynie*, PIOS's brief history has been tumultuous, though the company has held tenaciously to its vision. In an ideal world, a radically successful PIOS could be the factor that ensures continued support for PowerPC chips under BeOS. Only time and consumer spending patterns will tell.

Interview with Dave Haynie

Hardware Project Manager of PIOS and one of the "godfathers" of the Amiga architecture.

Interview by Scot Hacker

With Be completely removed from the hardware scene and Apple having completely disconnected the once-thriving PowerPC clone market, the business argument for continued support of PowerPC chips under BeOS is threatened. Furthermore, the BeBox represented something cool and seemingly unreplaceable: The combination of superior RISC CPU technology with cheap, standard, and often superior I/O cards from the x86 world. PIOS represents a sort of "last hope" for a continuation of many of the ideas that drove the original BeBox. I recently discussed the origins of and prospects for PIOS with Dave Haynie, hardware project manager of PIOS and one of the "godfathers" of the Amiga architecture.

SH: *How did PIOS come about? What were some of the motivations behind forming the company? Are you "picking up where the Amiga left off"?*

DH: That's how we got started. Set the wayback machine for the fall of '95. I had participated in a few potential "save the Amiga" projects, lost much of my savings account in the process, and was wholly disillusioned with the prospects of anything happening in that market. In November, I heard noises in the crowd the day Be announced their Web site to the world, and went to check it out. A few hours later, I had signed up as a developer, hoping to be one of the "lucky few" with a project intriguing enough to get a BeBox. This happened (I was planning some digital audio workstation software—kind of on hold for the last few years), and so I was a Be developer.

Dave Haynie (continued)

A few weeks later, I get a call from a Mr. Stefan Domeyer at Amiga Technologies, the new Amiga arm of ESCOM GmbH. He would like to fly me and my good friend Andy Finkel to Germany to talk about The Future of the Amiga. In Germany, I learn they're planning to do it right. They want to hire Andy's 20–30 person software team and build a new PowerPC-based Amiga system, targeting the $500 price point that most Amigas sold around. Back in the States, we each write up formal proposals, and before I know it, I'm the head of hardware development for Amiga Technologies. In my copious spare time (I'm still working for Scala, Inc. at the time), I fly to Austin and Phoenix to meet with Motorola, I fly to Germany again, etc.

Life isn't perfect. The PowerAmiga architecture I describe (look at today's MedixGX from Cyrix, put in PowerPC and Rambus, and you'll have a first-order approximation of what I was trying to do for AT) is too aggressive: Motorola doesn't even have a Rambus license; IBM does, but neither has PPC cores flexible enough to be reworked in short order (the goal was to finish in a year). But it was still possible to make low-cost systems using PPC602s, PPC603s, or some of the embedded chips, coupled with a "media processor" Motorola had in super-secret development with a partner in California, VM Labs, some kind of startup.

I went to the '96 CeBIT show and gave a talk to selected developers on the Power Amiga architecture. While the $500 machine was not CHRP, we had plans from the get-go to build a new AmigaOS based on an OS-level HAL (see Windows NT for an example), so this could run not only on our $500 machine, but on any possible PowerPC machine one would care to support, including PReP, CHRP, and even old Amigas with PowerPC cards in them. This was all well received, but by the end of March, ESCOM wasn't paying their bills.

In April, it became clear there were big problems with ESCOM, mainly from private mail from Stefan to me. He was looking for money to keep the project going. Later in April, Amiga Technologies dumped most of its technical staff. In May, I met with Mr. Domeyer, Andy, and another partner from Germany, Geerd Ebeling, and we formed PIOS. With the total collapse of ESCOM (which had been the second largest PC vendor in Germany), the Amiga technologies were being kicked loose, and the front-runner to buy them was apparently VIScorp. We met with VIScorp to discuss the idea of an Open Amiga platform.

VIScorp, it seems, was a Chicago-based company, with some ex-Amiga/C [Commodore?] personnel, working on a kind of WebTV-style browser box, which would extract dedicated or Internet content embedded in analog TV signals. They used the AmigaOS to do this, and did not care about desktop systems. We formed an agreement with their technical people to develop a sort of co-op, where various pieces of the OS could be developed by interested parties, under the direction of a small intercompany council. So we could work on desktops, they could work on set-tops, others on maybe handhelds or whatever interested them.

A month later, VIScorp, who had still not actually purchased the Amiga assets, was talking about making desktop Amiga systems. It got weirder and weirder, as they threatened legal action against anyone working on Amiga or Amiga clone systems. Ultimately, they went away, and the Amiga technology was once again homeless. By late summer, we had had our fill of this nonsense. Mr. Domeyer was out at the MacWorld Expo—PIOS was reselling Mac clones in the German market as a way to establish a presence and get into the PowerPC market at the

Dave Haynie *(continued)*

time. He met with the Be, Inc. folks, not knowing I was already into BeOS myself. He called me, asking about BeOS, I immediately gave it the green light, and that become the new target. BeOS wasn't the AmigaOS in the sense that it would be good on a $500 machine, but it was also much more likely to enable the markets I was familiar with from the Amiga high end. That high end had been good for 50K–100K machines/year during most of Commodore's life; not great for Commodore, but plenty large enough for our purposes. That fall, I started working on the design.…

SH: *What have been some of the biggest hurdles PIOS has had to overcome?*

DH: I think, in order, it would be: money, Apple, and human resources.

I started working on the PIOS One in my garage, with about 1/3 of the development costs out of my own pocket. I had a very basic "midnight engineer"-type in-the-PC combination logic analyzer and slow-scan oscilloscope. I was shut down most of the summer of '97 due to our lack of higher-level equipment. While the computer revolution was founded in the garage, there's a big difference between working on an 8-bit computer at 1 MHz and a 64-bit (external) computer at 66+ MHz.

Until last fall it was only me, but we managed to hire Thomas Rudloff, formerly of Phase V, who's working in Germany on additional PowerPC things, including our four-processor board and CPU modules for some of the PowerMac systems.

SH: *Was the PIOS One well under way before Be announced termination of the BeBox?*

DH: Yes. In fact, during our trip to California in 1/97, for the BeDevCon and the Apple talks, we also met with Be at their home office. Joe Palmer showed me the next-generation BeBox, using, oddly enough, a CPU-slot architecture not unlike what I had for the PIOS One. Great minds and all that. Unfortunately, it was equally clear that they had decided to stop hardware production, and that it had been a decision made quite recently. This wasn't announced at the BeDevCon, and I had to be quiet while they spoke of continued life in the BeBox.

My take on the whole cancellation was, as you can view most things in the PPC market, as a reaction to Apple. Now, Be's structure was clearly pro-software, as it had to be. You basically had Joe on the hardware side, everyone else on the software side (they may have gotten someone else for hardware before the end, I know they were looking). The discussions about the end of the BeBox ranged from "too costly to make" to "not our focus," but I suspect it was the whole notion of noncompetition. Apple has always had the problem of being in competition with its hardware vendors. Commodore had that too; even if we didn't have Amiga clones, the add-in pieces were always a bit risky for developers, because Commodore or Apple could come out with the same thing any time, on a much larger scale. If Be was in direct competition with Power or Motorola or any of the other supposed big licensees, it would be a problem.

Another footnote: At this time, PIOS was the BeBox reseller in Germany, and while it wasn't a monstrous business, we were selling them faster than Be could deliver. We tried to negotiate for the rights to make the BeBox and all the leftover pieces at Be, but they apparently wanted too much money for this.

SH: *Performance-wise, how does the PIOS machine compare with similarly clocked Macs and Intel boxes?*

DH: We're very similar to the Mac G3 machines, depending on processor. This isn't accidental; we're using the same MPC106 system controller chip, from Motorola, that Apple uses. This chip determines the kind of bus speeds and memory speeds and types the system can support. We expect to move to 100 MHz buses, at least by next year on single processor machines. There's something of an issue with today's PPC versus the Intel chips for SMP. Intel has these horrible SEC modules—the last thing in the world you want in your most critical path is a connector. But they use a special kind of transceiver logic, GTL+, to allow two SEC modules to run at 100 MHz on the same bus, if you're careful. The BGA packages for PPC are nearly ideal (you can fit four PPCs in the space of one SEC module), but they're using a more conventional CMOS/TTL-type bus. So while it may be possible to get two PPC750s going at 100 MHz, you're not going to see four-processor systems with that bus speed. Intel has a new spin of their Gunning drivers, AGTL+, for the "Slot 2" architecture, which is what, in theory, allows four such modules on the same bus at the same time.

If Motorola were to really push the architecture, I would favor something like the Alpha 21264 bus, rather than heading in Intel's direction. This is a 64-bit point-to-point bus (there's only two devices ever on the bus), which can run at 350 MHz today, a good 3.5 times faster than any Intel or PowerPC bus. SMP systems are managed with a system chip using an internal cross-point switch to main memory, rather than the traditional shared, arbitrated bus of Intel and Motorola today.

The cool thing about the PIOS One is that, the way I did the CPU cards, if they did support this, or even the new "Max" bus (the first "Max" processors will run in old PPC-bus mode, but also offer a more advanced bus mode, yet to be supported by a Motorola system chip), I can offer an upgrade CPU module for the PIOS One that takes full advantage of this new bus. You don't get this in any other machine currently announced. For that matter, we could deliver a Pentium or Alpha module, if that seemed like a reasonable direction at some point. The CPU module is being presented as a free and open spec for anyone else to use, too. It's based mainly on PCI technology; it's not all that hard for anyone to duplicate anyway, but with 10+ years of experience doing CPU slots (since the Amiga 2000, in which some people today stick PowerPC boards, 11 years after the 2000 first shipped), I might have an idea or two others would miss.

Epilogue In August of 1998, PIOS apparently came under the umbrella of another German company and/or product, called MetaBox (**`www.metabox.de`**). However, MetaBox's pages are not offered in both German and English, as were the pages on PIOS's original site. If you'd like to try and determine the status of the PIOS or MetaBox projects, you might want to try running the MetaBox pages through AltaVista's BabelFish translator (**`babelfish.altavista.com`**). At this writing, the future of the PIOS line appeared fuzzy, and inquiring e-mail was not returned.

Converging on the Media

Much has been made over the past few years of the confluence of our most common media delivery mechanisms. Every year, another study indicates that people are spending more time in front of computer screens and less time in front of televisions. One may make the argument that this is a good thing for our culture, as television viewing is generally a more passive activity than Web surfing or game playing. Interestingly, though, it's slowly becoming more difficult to tell television and the Web apart. The Web struggles to deliver an advertising model as successful as television's, TV adopts Web-like interfaces in sports coverage, cable channels specialize in niche topics as obscure as some Web sites, TVs acquire increasingly sophisticated onscreen menus that resemble point-and-click Web pages, flat-screen TVs borrow technology from laptop displays, digital broadcasting becomes mandatory in some areas … the list goes on.

According to Sony, it won't be long before televisions begin to break their display and computation functions into separate boxes, for independent upgradeability. Set-top boxes such as the WebTV system exemplify this crossover between old media delivery systems and new. It won't be long before televisions come preenabled with voice-recognition technology, displays in billions of colors, surround-sound stereo and Dolby noise reduction, and immersive, three-dimensional virtual reality. All of this causes pundits to ask questions like "Are computers becoming more like TVs, or vice versa?"

The answer, of course, is yes. Both are true, and one can envision a point in the future when the distinction becomes negligible. Currently, most computer users lack a connection to the outside world capable of delivering a 3GB movie in 90 minutes, and even if they had that kind of throughput, most of the hardware/OS combinations out there would not be capable of presenting "download on demand" copies of *The Terminator* full-screen at 30 fps anyway. At the same time, non-Internet-enabled televisions don't offer the scope of choice users find online, and don't let users toggle over to work on a spreadsheet during the commercials .

The first decade of the next millennium will likely bring high-bandwidth Internet to millions of homes, and faster processors inside high-definition televisions (HDTVs) will enable TVs to perform more computer-like tasks. Nor can we leave music distribution out of the picture. Just as you'll soon be able to log into **blockbuster.com** to download a midnight movie, you'll also be able to log into record labels' Web sites to rent or purchase audio tracks. The custom cassette tape from days of yore will give way to custom CDs burned on the home network server. Tracks will be paid for with virtual credit cards meting out micropayments measured in hundredths of a cent.

One of the first "bridges" between computers, television, and hi-fi systems is the DVD, or Digital Versatile Disc, format. While DVD is getting off to a rocky start with standards wars that eclipse those waged between Betamax and VHS in the '70s, we'll eventually be able to swap DVD disks containing custom combinations of computer data, audio tracks, and video information effortlessly between the TV, stereo, and computer—which may, in turn, become a single device.

Tying it all together will be the home network, which will finally rise above its current "geeks only" status and become the norm in modern households. The living room server will dish up movies, data, and audio to terminals scattered throughout the house, make your house look occupied while you're on vacation, relay images of your kids from the preschool to your work PC, answer your phone, take and deliver voice notes for other family members, auto-download this week's programming guide into your TV's memory, find a cure for gingivitis, and give you a pedicure every Tuesday at 3:00. You get the picture.

Okay, a lot of that sounds like hype, but the writing's on the wall—all of our most important technologies are converging on a point, and all of them come back ultimately to the thing BeOS does best: juggling media. Every one of these technologies feeds perfectly into the realms Be has so carefully targeted. Of course, certain other operating system vendors have their eyes on the convergence prize as well, so it should be an interesting decade. But even if Bill somehow finds his way into your DVD player and set-top box, you'll still get to pick the OS your home network runs on.

Embedded Chips

While we think of an operating system almost exclusively as an environment for the desktop or laptop computer, computing doesn't stop at the desktop. The physical world is getting "smarter," and it seems that the computer industry is hell-bent on installing a little CPU in every greeting card, movie theater seat, “one-time-use” camera, and wristwatch around. Not all of these devices require an operating system (in any traditional sense) to function, but many of them do. How do you think they synchronize the stoplights to make them turn red right before you get to them? A team of evil monkeys programming in some Transportation Department basement? Nope—embedded chips.

Be has never announced plans to work with embedded chip makers to have a variant of BeOS implanted in circuit boards or ASICs (custom chips); that’s not Be’s media target, several companies already excel at this, and Be is nose-to-the-grindstone perfecting the MediaOS for desktop computers. On the other hand, BeOS has something going for it that the other mainstream OSs don’t—it’s incredibly compact and efficient.

Though it’s never been released to the public, Be once successfully squashed the OS down to its bare essentials and fit it onto a single bootable floppy, similar to the famed QNX demo disk (QNX is the maker of a compact operating system intended for customized, vertical applications).

Be is too focused on their stated target right now to compete in the embedded chip market. Then again, play CEO for a moment and think about the sheer size and scope of the following industries. Ask yourself which of these would benefit from a stripped-down version of BeOS running behind the scenes:

photocopiers

personal digital assistants

robots

information kiosks

airline reservation machines

automated toll booths

calculators

fax machines

avionics

factory automation

handheld computers

pay-at-the-pump kiosks

smart phones

game consoles

medical and laboratory tools

network computers

palmtop computers

personal organizers

ATMs

retinal scanners

set-top boxes

cash registers

planetary probes

supermarket UPC readers

Chapter Summary

- Because it's built for the future rather than for the past, there are tremendous possibilities for BeOS as chips and system bus speeds get faster and as "the age of convergence" moves from theory to reality.
- Advanced peripheral connection buses like USB and FireWire will make it even easier to connect external devices, from keyboards to video cameras, to your computer, and BeOS support for these protocols is in the works.
- BeOS will gain desktop and laptop power management support, ultimately making it possible to run the system on laptops alongside other laptop-compatible operating systems.
- Future versions of BeOS will support multiple monitors, spatialized audio processing, NTFS support, and further improvements to the networking infrastructure.
- The functionality of BeDepot will expand, and it will be possible to let your system troll regularly for automated updates to system components.
- It is possible—but quite unlikely in the near term—that BeOS could find its way onto other hardware architectures. The most likely scenario is that Be will continue to follow the current x86-based market as it evolves toward its own next generation of high-bandwidth processors.