

GSoC Interview: Johannes Wischert.

Contributed by DaaT
Sunday, 10 May 2009

Haiku Gazette, through Humdinger, interviewed GSoC student Johannes Wischert and provided ICO with an english translation. Here's the full piece. Enjoy:

Johannes Wischert is one of six Haiku students at Google's Summer of Code 2009. He'll be working on a port to the ARM architecture. His blog has all the details to the goals of his project. Besides, we learn that he's 25 years old, studies computer science and has been programming since he was a kid. As part of a series of GSoC interviews that are being published by various Haiku news sites, Johannes was so kind to answer a few of our questions.

[You can find German original of this interview at the Haiku Gazette.]

HG - Congratulations, you made it to GSoC! Damn, now you have to continue working hard all through summer, right? :)

Johannes Wischert - I didn't really expect to be chosen, let alone that my application got such a positive reaction. ;) I'm well aware that it'll be much work. Luckily, I have enough time to get it done.

HG - Do you already have all you need to start your project?

Johannes Wischert - On the hardware side I'll be focusing on the Beagleboard, that was generously donated to me by Haiku Inc. Softwarewise Haiku offers all I need to port to the ARM/Beagleboard without too big problems. Everything else needed is time, which I've got thanks to Google's grant...

HG - You've already got some experience porting the nano kernel SmartOS from MSP430 to SuperH. What do you expect from the Haiku port in comparison?

Johannes Wischert - SmartOS being "just" a nano kernel, the amount of code was of course much smaller and easier to handle than Haiku. On the other hand, many things there were written very platform specific for performance reasons and porting to other systems was never a focus.

To Haiku's advantage, much is already cleanly separated into platform specific/unspecific due to the porting to m68k and other processors. That helps me a lot, of course. Then again, Haiku is much more complex than SmartOS, which doesn't use an MMU, for example.

I'll start by porting the kernel and the essential hardware drivers, i.e. SD-card and serial console. If I still have some time left at the end of GSoC, I'll also code the framebuffer and a driver for the USB controller. If I can't fit it into GSoC, I'll just do it afterwards. ;)

HG - How do you go about porting an OS to another platform?

Johannes Wischert - At the moment I work under Linux where I've built a haiku-arm cross-compiler and run the code inside a slightly modified version of qemu. I'll soon post about the specifics on my blog at Haiku-os.org. With qemu I start U-Boot and from there I can load files from the emulated SD-card. This should work similar with the real Beagleboard. The output comes through the serial port and it's also possible to connect qemu with gdb to observe Haiku while running. This may also work with the real hardware via the JTAG interface, but I didn't have a look at that yet. ;)

First, I have to build a minimal kernel that only contains the most essential methods to which all others are added piece by piece. Currently, I have just assembled this stubbed kernel without functionality from parts of the m68k port and will now fill it slowly with working code. After that I can begin to test everything in qemu. For that I still have to build the U-Boot loader, however. This shouldn't be too hard, though, thanks to the U-Boot API. After the kernel itself boots up I'll continue with the necessary driver support.

HG - Provided you achieve all the goals for your GSoC, how much work is still needed to have Haiku run identical on ARM as on x86?

Johannes Wischert - Most of the software should be easily portable (SSE, for example, is mostly used in "external" software like ffmpeg and there are already ports for that). At least on the Beagleboard, endianness isn't a problem as ARM is running little endian there. I'm not sure about the situation with other board, ARM processors are available as little and big endian...

HG - So... would an end user be happy with an ARM netbook, say by the end of next year?

Johannes Wischert - The main things missing for a "real" Haiku is the graphics driver, which should be quite easily ported from Linux (though without 3D, as that's still closed source as far as I know) and the driver for the USB controller, which also shouldn't take to long...

Netbooks probably need some more special hardware drivers. According to the manufacturer, the Touch Book is supposed to be closely designed after the Beagleboard and should therefore be relatively easy to support. Depending how much time my studies leave me, Haiku on an ARM netbook may even be possible by the end of this year. But don't nail me down to this. ;)

I will definitely get an ARM netbook, sooner or later.

HG - Do you think some components, like Tracker or the MediaKit, may be too resource hungry to run on ARM?

Johannes Wischert - I don't think Tracker or MediaKit are too much for an ARM processor. It's running with 600mhz on the Beagleboard and Linux is able to playback fullscreen videos on it without problems. Admittedly, they do use the built-in DSP, but I expect Haiku to do that too, sooner or later.

Writing hardware drivers is a big part toward a complete port of Haiku. Here, the Beagleboard has the advantage that most of the peripheral devices are connected via USB. You only need a driver for the controller to use all Haiku supported hardware.

HG - How did you come up with an ARM port as GSoC project?

Johannes Wischert - The idea of an ARM port came up when I joined the #haiku IRC channel while trying out Haiku once more. Just for fun I suggested to port Haiku to MIPS (which actually would also be an interesting project, with the chinese MIPS Clone "Longsoon" and all the related devices coming out. But that's for another day...). We slowly came to the conclusion that porting to ARM makes more sense. Urias McCullough offered to donate a Beagleboard to anyone porting Haiku to it...

That's when I looked into the matter to evaluate the complexity of this task and started playing around to get the necessary toolchain working...

One day I saw on the Haiku-os.org website that Haiku is again part of Google's Summer of Code. Since I planned to apply for that anyway, I applied with the ARM project.

I've known Haiku for a while now, ever since I've been shopping around for alternative operating systems (back then it was still called OpenBeOS...). I also knew BeOS, which I encountered on a Chip Magazine CD. I played around with it a bit, but I got bored after a while because I didn't have an internet connection at home back then.

HG - Every applicant had to pick a ticket from the bugtracker and solve it. To do that, you had to set up a working developing environment and become a bit better acquainted with the Haiku code base. How did that go for you?

Johannes Wischert - I have set up the developing environment a couple of times before, so this wasn't difficult for me. I remember although, the first time I tried that, I had to look around for the right guides for some time. Those are compiled into one updated document at the moment, as far as I know. That's a general problem of the Haiku website: it contains lots of good information, but it's hard to find...

Apart from that, everything worked out fine. ;)

HG - What about Haiku as a mentor organization. Is there anything that could make Haiku even more attractive for future GSoCs?

Johannes Wischert - I had a look at a few other GSoC mentors, but since I was pretty much sold on the ARM port anyway, I only contacted the Haiku project. Everything went very well! My questions were quickly answered and once I've sent off my application, I immediately got constructive feedback.

HG - Is there anything you'd like to tell potential future GSoC students?

Johannes Wischert - Don't be intimidated, the others aren't that much better anyway. ;) Even if there are only 1,000 slots altogether, it's not that improbable to get in.

Thanks for that nice interview, Johannes. Have a successful and fun Summer of Code!

We at ICO would also like to thank Johannes and Humdinger, both for the interview and the translation :)