Subject: Help with port to ARM architecture
Posted by Stephen on Fri, 16 Nov 2007 04:37:28 GMT
View Forum Message <> Reply to Message

This is my first post - so, let me thank the developers for providing such an interesting and capable open-source product! The wiki and documentation in general have also been very useful in getting me up and running.

Now I would like to ask for some help; if this is not the right place, please let me know where I can redirect this.

I am porting OpenVZ to the Arm architecture as part of an academic project. I have a working initial port, running on a Qemu emulated Versatile board, on a linux host. I am able to create, start and stop VPS, remotely connect to them from outside of the emulator, as well as enter them with vzctl. Within a VPS, ps -A seems to show only init and those pids associated with that VPS (though I have not tested this extensively).

I am aware of this thread on porting:
http://forum.openvz.org/index.php?t=msg&goto=3379
But I have zero kernel programming experience, except that which I have gained from getting the OpenVZ patches to compile and run for Arm, and not much system programming experience, so the instructions there are not enough for me to be sure I am doing the right things.

So far all I have done are the changes necessary to enable compilation and running of OpenVZ and vzctl, vzquota etc. What can I do to expose the bugs that very likely remain, so I can fix them? I have gdb available, I am looking for test cases, or ways to force things to break.

I realise that this is the sort of thing that practised OpenVZ
developers would be able to solve much faster than me, and that it may be annoying to ask for help for something that is probably quite simple. Please let me know if I am asking for too much, or would be taking up too much of your time.

In case it is relevant, I am currently working with somewhat old versions of most things, partly due to the environment of the intended target: kernel 2.6.18, OpenVZ 028stab039, vzctl-3.0.16, vzquota-3.0.9. Notably, the target uses uclibc. I am currently working with 0.9.28.