Note 28_12_22

(12:55)
Today, I'm going to start working on writing my Rust Mouse driver on the VirtualBox instance. By the end of today, I'd like a very very basic skeleton to be set up.

(13:58)
For some reason, the rustc compiler can't be found.... I have no clue how this happened but it seems I can't have a day of development work where there is some kind of roadblock...

It's really frustrating because rustc is clearly installed its just that the kernel build can't find it.

Currently rebuilding the kernel and we'll see if this fixes it.

(15:00)
I've decided to look for Rust driver repos on GitHub in order to test them to check whether I can actually compile a Rust driver on this VM instance.

https://github.com/alexandruradovici/rust_linux_drivers

(15:05)
Fairly certain that this VM simply can't build any sort of Rust driver. Trying to build with Cargo doesn't work at all and likely isn't intended to work. Trying to build as if it were a traditional kernel module (with Make and `obj-m`) does not work either and returns errors.

(16:07)
After taking a break and trying the repo driver again, I'm now seeing Rust errors.

(16:15)
I can confirm that Rust itself works so it seems I have some kind of Rust kernel build issue.... and once again the rustc compiler has somehow disappeared from existence.

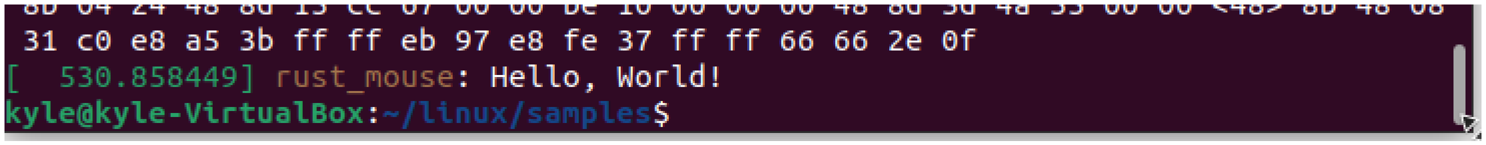
Rebooting and trying to build that repo driver once again provides weird results. I keep running make and it writes some kind of output but doesn't produce anything.

I'm unable to get any sort of result from the Virtual Machine. I find this very strange considering 2 months ago I was at least able to produce a .ko file (even if insmod would reject it).

(16:49)
On the QEMU virtualbox which I set up earlier in the project, I managed to build the Char driver in Rust by adding it to the samples list (Kconfig and Makefile). I'm hoping I can pull off the same thing in the VirtualBox instance and essentially force my driver to be built.

This will then mean for testing, I may need to rebuild the entire kernel or at most the modules though (if this works) I think it is my best option at the moment. Trying to build Rust modules in the same way as C drivers doesn't seem to work. At least, I wasn't able to get it properly working again. It still bothers me though that earlier in the project, I was able to get things building this way.

(17:02)
After enabling the samples and trying to rebuild, I ran into the same errors which were experienced earlier. After comparing a Rust sample that sucessfully compiled, I found that several issues with my code that are likely to be the cause of the previous errors. Fixing these issues has now allowed my kernel to build and I now expect the module to be able to run.

(17:09)


```
 31 c0 e8 a5 3b ff ff eb 97 e8 fe 37 ff ff 66 66 2e 0f
[  530.858449] rust_mouse: Hello, World!
kyle@kyle-VirtualBox:~/linux/samples$
```

After successful compilation and execution of the mouse driver (which at the moment, simply prints hello world). I decided to try the fixed code with a makefile and can confirm the build system has no problems. I should now be able to build my driver with no issues.

(17:18)
The driver is able to independently build with zero issues. I can now start providing the skeleton for my mouse driver.

I think before I do that it might be a good idea to figure out what I can do with Rust and what I'll need to do in unsafe C. The current Rust code available in the Kernel is limited and I previously established that the USB library is not yet available.

I'd imagine that a lot of the data structures and so on can pretty much all be kept in Rust so it might be that most of the functionality is programmed in unsafe C. I'll also need to look into using C within Rust as it has just ocurred to me that I have never worked with this feature before.

Will be good to also write up an explanation of how my driver works etc.

(18:22)
I worked from a very short tutorial and created a very basic project that links to C from Rust
I'm going to continue later by watching this talk which discusses embedding Rust in C/C++.
(19:57)
Working on RustConf 2018 Talk notes.

(20:57)
I finished my notes a short while ago and will be ending the day here.