8th November 2014

Announce github repo prototype-kernel

Kernel prototype development made easy

For easier and quicker kernel development cycles, I've created a git-repo called "prototype-kernel". To *also* help your productivity I've shared in on github (isn't it great that work at +Red Hat [https://plus.google.com/113311940092135970473] defaults to being Open Source):

- https://github.com/netoptimizer/prototype-kernel [https://github.com/netoptimizer/prototype-kernel]

Getting started developing your first kernel code is really easy: follow this guide [https://github.com/netoptimizer/prototype-kernel/blob/master/getting_started.rst] .

The basic idea is to compile modules outside the kernel tree, but use the kernels kbuild infrastructure [https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/kbuild/makefiles.txt] . Thus, this does require that you have a kernel source tree avail on your system. Most distributions offer a kernel-devel package, that only install what you need for this to work.

On +Fedora Project [https://plus.google.com/112917221531140868607] do: '`yum install kernel-devel`'
On +Debian [https://plus.google.com/111711190057359692089] do: '`aptitude install linux-headers`'

Directory layout:

There are two main directories: "`minimalist`" and "`kernel`".

If you just want to test a small code change go to the "minimalist [https://github.com/netoptimizer/prototype-kernel/tree/master/minimalist] " directory. This contains the minimal `Makefile` needed to compile the kernel modules in this directory. Do you devel cycle of: Modify, make, insmod, rmmod.

The "kernel [https://github.com/netoptimizer/prototype-kernel/tree/master/kernel] " directory is more interesting to me. It maintains the same directory layout as the mainline kernel. The basic idea is that code developed should be in the same locations that you plan to submit upstream. Here you need to edit the files Kbuild [https://github.com/netoptimizer/prototype-kernel/blob/master/kernel/Kbuild] files when adding new modules/code. Notice these Kbuild [https://github.com/netoptimizer/prototype-kernel/blob/master/kernel/lib/Kbuild] files are located in each sub-directory.

Detecting kernel and compiling

In the "kernel" directory you can **simply run: `make`**

The system will detect / assume-to-find the kernel source by looking in `/lib/modules/`uname -r`/build/`. If the kernel you are compiling for are not in this location, then specify it to make on the command line like:

```
$ make kbuilddir=~/git/kernel/net-next/
```

Compiling for a remote host

If you are compiling locally, but want to upload the kernel modules for testing on a remote host, then the make system supports this. Compile and upload modules with a single make command do:

```
$ make push_remote kbuilddir=~/git/kernel/net-next/ HOST=192.168.122.49
```

Localhost development

Development on the same host is sometimes easier. To help this process there is a script [https://github.com/netoptimizer/prototype-kernel/blob/master/kernel/scripts/setup_prototype_devel_env.sh] to help setup symlinks to from the running kernels module directory into you git repo. This means you just need to compile and no need to install the modules again and again (perhaps run 'depmod -a' if needed). Command line for this setup, be in the "kernel" directory and run:

```
$ sudo ./scripts/setup_prototype_devel_env.sh
```

Development cycle made simple

After this setup, standard modprobe and modinfo works. Thus, oneliner work cycle for testing a given kernel module can be as simple as:

```
$ make && (sudo rmmod time_bench_sample; sudo modprobe time_bench_sample) && sudo dmesg -c
```

Happy kernel-hacking :-)

Posted 8th November 2014 by Jesper Dangaard Brouer

Labels: kernel, kernel modules, linux, prototype kernel, prototype-kernel, testing tools, time_bench

2 View comments

2 comments

Add a comment as Stain Rainbow

Top comments

Jesper Dangaard Brouer via Google+ 2 years ago - Shared publicly

Announce github repo prototype-kernel

Kernel prototype development made easy For easier and quicker kernel development cycles, I've created a git-repo called "prototype-kernel". To also help your productivity I've shared in on github (isn't it great that work at +Red Hat defaults to being Open...

+10 1 · Reply

Martin Møller Skarbiniks Pedersen shared this via Google+ 2 years ago - Shared publicly

+1 1 · Reply