

Introduction to Linux Kernel Programming

Andy Spencer

Jan. 4, 2020
San Fernando Valley Linux Users Group

The Typo Fix

```
commit 9eac0ae1683575375de8c63166b3596b11d3b56a
Author: Krzysztof Kozlowski <krzk@kernel.org>
Date: 2019-01-08 13:11:40 +0100
```

```
dt-bindings: soc: qcom: Fix trivial language typos
```

```
Fix few trivial language typos in bindings.
```

```
Signed-off-by: Krzysztof Kozlowski <krzk@kernel.org>
Signed-off-by: Rob Herring <robh@kernel.org>
```

```
diff —git a/Documentation/devicetree/bindings/soc/qcom/qcom,glink.txt b/Docume...
index 0b8cc533ca83..cf759e5f9b10 100644
— a/Documentation/devicetree/bindings/soc/qcom/qcom,glink.txt
+++ b/Documentation/devicetree/bindings/soc/qcom/qcom,glink.txt
@@ -55,7 +55,7 @@ of these nodes are defined by the individual bindings for the...
= EXAMPLE
```

```
The following example represents the GLINK RPM node on a MSM8996 device, with
the function for the "rpm_request" channel defined, which is used for
-regualtors and root clocks.
+regulators and root clocks.
```

```
apcs_glb: mailbox@9820000 {
    compatible = "qcom,msm8996-apcs-hmss-global";
```

The Hardware ID

commit 7ab21a8692094872298df172f54d55cba72fd308
Author: Andy Spencer <spenceal@rose-hulman.edu>
Date: 2009-01-02 16:19:13 +0000

i8k: Enable i8k on Dell Precision Systems

Patch to enable i8k on Dell Precisions.

Signed-off-by: Andy Spencer <spenceal@rose-hulman.edu>

Signed-off-by: Alan Cox <alan@redhat.com>

Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>

```
diff --git a/drivers/char/i8k.c b/drivers/char/i8k.c
```

```
index b60d425ce8d1..099fc89a5bdd 100644
```

```
--- a/drivers/char/i8k.c
```

```
+++ b/drivers/char/i8k.c
```

```
@@ -485,6 +485,13 @@ static struct dmi_system_id __initdata i8k_dmi_table[] = {  
    DMI_MATCH(DMI_PRODUCT_NAME, "MP061"),  
    },  
+   {  
+       .ident = "Dell Precision",  
+       .matches = {  
+           DMI_MATCH(DMI_SYS_VENDOR, "Dell Inc."),  
+           DMI_MATCH(DMI_PRODUCT_NAME, "Precision"),  
+       },  
+   },  
+   {}  
};
```

Examples Patches

The Driver Debug

commit ab1796ebdad3017965754d86db64d3cba5c416b9
Author: Andy Spencer <andy753421@ucla.edu>
Date: 2014-05-02 06:48:13 +0000

RTL8192CU: Increase max APFM_ONMAC polling count

With certain hardware combinations the poll interval is exceeded before initialization completes.

Tested on a MacBookPro10,1 using a Sabrent USB-A11N USB adapter.

Signed-off-by: Andy Spencer <andy753421@ucla.edu>
Acked-by: Larry Finger <Larry.Finger@lwfinger.net>
Signed-off-by: John W. Linville <linville@tuxdriver.com>

```
diff --git a/drivers/net/wireless/rtlwifi/rtl8192cu/hw.c b/drivers/net/wireless/rtlw...
index 68b5c7e92cfb..31b79e78c63c 100644
--- a/drivers/net/wireless/rtlwifi/rtl8192cu/hw.c
+++ b/drivers/net/wireless/rtlwifi/rtl8192cu/hw.c
@@ -511,7 +511,7 @@ static int _rtl92cu_init_power_on(struct ieee80211_hw *hw)
     pr_info("MAC auto ON okay!\n");
     break;
 }
-     if (pollingCount++ > 100) {
+     if (pollingCount++ > 1000) {
         RT_TRACE(rtlpriv, COMP_INIT, DBG_EMERG,
             "Failed to polling REG_APS_FSMCO[APFM_ONMAC] done!\n");
         return -ENODEV;
```

The New Protocol

- ▶ New network protocols, filesystems, security modules, etc.
- ▶ Software only improvements to the kernel.
- ▶ No hardware required, can be tested in a VM.
- ▶ Typically done in net/, fs/, security/, etc.

The New Driver

- ▶ Add support for a new hardware device.
- ▶ Cononical example: I2C/SPI thermal sensors. (lm70.c)
- ▶ Find the data sheet
- ▶ Typocally use an existing driver as an example.
- ▶ Requires hardware to test on.
- ▶ Typocally done in drivers/.

The New Board

- ▶ Similar new drivers but focus on boot.
- ▶ Typically for embedded Linux.
- ▶ May require bootloader changes.
- ▶ Typically done in arch/.

Kernel compiling

Could be an entire talk on it's own...

- ▶ `$ make menuconfig`
- ▶ `$ make -j32`
- ▶ `$ make install`
 - ▶ invokes `$HOME/bin/installkernel`
- ▶ `$ make modules_install`

Output files:

- ▶ `vmlinux`: Main build output, ELF file with debug symbols.
- ▶ `bzImage`: Bootable kernel image (wraps `vmlinux`).
- ▶ `ulmage`, etc: Like `bzImage` for other architectures.

Booting the kernel

How to test?

1. Compile, reboot, compile, reboot..
 - ▶ Least efficient, but often needed for debugging.
2. Run the kernel in a VM.
 - ▶ Great for experimenting.
 - ▶ Won't work for driver.
3. Dynamic kernel module loading.
 - ▶ Good for most driver work.
 - ▶ Easy to crash your PC.
4. Boot it on a dedicated machine.
 - ▶ Generally used for embedded Linux.
 - ▶ Prevents crashing your workstation.

Kernel source tree

- ▶ arch/: architecture specific code
- ▶ drivers/: well... drivers
- ▶ fs/: file systems
- ▶ kernel/: core kernel code (process management, etc)
- ▶ lib/: utility code, C library type stuff
- ▶ mm/: memory management
- ▶ include/: shared include files

Sending patches

`https://www.kernel.org/doc/html/latest/process/submitting-patches.html`

- ▶ Linux uses mailing lists patches and code review.
- ▶ Maintainers tend to use pull requests and merges, but general contributors usually do not.
- ▶ Different subsystem (netdev) have slightly different rules.
- ▶ Patchwork is useful for tracking and reviewing patches:

`https://patchwork.kernel.org`

Hello, World

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

static int __init hello_init(void)
{
    pr_info("Hello_SFVLUG.\n");
    return 0;
}

static void __exit hello_exit(void)
{
    pr_info("Goodbye_SFVLUG.\n");
}

module_init(hello_init);
module_exit(hello_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Andy_Spencer_<andy753421@gmail.com>");
MODULE_DESCRIPTION("SFVLUG!");
```