Fast and Precise Retrieval of Forward and Back Porting Information for Linux Device Drivers

Julia Lawall, Derek Palinski, Lukas Gnirke, Gilles Muller
(Inria/LIP6)

October 24, 2017
The porting problem
The porting problem
The porting problem
The porting problem

forward ported driver
The porting problem

4.6

forward ported driver

Upstreaming can help
The porting problem

forward ported driver
The porting problem

back ported driver
The porting problem

Upstreaming is no use

back ported driver
Porting methodology

1. Compile the driver with the target kernel.
Porting methodology

1. Compile the driver with the target kernel.
   - See what breaks.
Porting methodology

1. Compile the driver with the target kernel.
   - See what breaks.
   - What breaks is in the interface with the kernel.
Porting methodology

1. Compile the driver with the target kernel.
   - See what breaks.
   - What breaks is in the interface with the kernel.

2. Study commits in the code history that fix each breakage.
Porting methodology

1. Compile the driver with the target kernel.
   - See what breaks.
   - What breaks is in the interface with the kernel.

2. Study commits in the code history that fix each breakage.
   - Other drivers have faced the same issues.
Porting methodology

1. Compile the driver with the target kernel.
   - See what breaks.
   - What breaks is in the interface with the kernel.

2. Study commits in the code history that fix each breakage.
   - Other drivers have faced the same issues.

3. Apply the observed changes to port the driver code.
Porting methodology

1. Compile the driver with the target kernel.
   - See what breaks.
   - What breaks is in the interface with the kernel.

2. Study commits in the code history that fix each breakage.
   - Other drivers have faced the same issues.

3. Apply the observed changes to port the driver code.

   If it compiles, it works?!
Porting example

- lms501kf03 TFT LCD panel driver
- Introduced in Linux 3.9 (commit 1be9ca2, 2013)
- Want to port it to Linux 4.6 (2016)
Step 1: Compilation result (gcc)

lms501kf03.c:433:2: error: unknown field ‘suspend’ specified in initializer
   .suspend = lms501kf03_suspend,
   ^

lms501kf03.c:433:2: warning: missing braces around initializer
lms501kf03.c:433:2: warning: (near initialization for ‘lms501kf03_driver.driver’)
lms501kf03.c:433:2: warning: initialization from incompatible pointer type
lms501kf03.c:433:2: warning: (near initialization for ‘lms501kf03_driver.driver.name’)
lms501kf03.c:434:2: error: unknown field ‘resume’ specified in initializer
   .resume = lms501kf03_resume,
   ^

lms501kf03.c:434:2: warning: excess elements in struct initializer
lms501kf03.c:434:2: warning: (near initialization for ‘lms501kf03_driver’)

Many messages are triggered by earlier failures and may be inconsistent.
Step 1: Compilation result (gcc)

```c
lms501kf03.c:433:2: error: unknown field ‘suspend’ specified in initializer
   .suspend = lms501kf03_suspend,
   ^
lms501kf03.c:433:2: warning: missing braces around initializer
lms501kf03.c:433:2: warning: (near initialization for ‘lms501kf03_driver.driver’)  
lms501kf03.c:433:2: warning: initialization from incompatible pointer type 
```

```c
lms501kf03.c:433:2: warning: (near initialization for ‘lms501kf03_driver.driver.name’) 
lms501kf03.c:434:2: error: unknown field ‘resume’ specified in initializer
   .resume = lms501kf03_resume,
   ^
lms501kf03.c:434:2: warning: excess elements in struct initializer
lms501kf03.c:434:2: warning: (near initialization for ‘lms501kf03_driver’) 
```
Step 1: Compilation result (gcc)

lms501kf03.c:433:2: error: unknown field ‘suspend’ specified in initializer
   .suspend = lms501kf03_suspend,
   ^
lms501kf03.c:433:2: warning: missing braces around initializer
lms501kf03.c:433:2: warning: (near initialization for ‘lms501kf03_driver.driver’) 
lms501kf03.c:433:2: warning: initialization from incompatible pointer type
lms501kf03.c:433:2: warning: (near initialization for ‘lms501kf03_driver.driver.name’) 
lms501kf03.c:434:2: error: unknown field ‘resume’ specified in initializer
   .resume = lms501kf03_resume,
   ^
lms501kf03.c:434:2: warning: excess elements in struct initializer
lms501kf03.c:434:2: warning: (near initialization for ‘lms501kf03_driver’) 

Many messages are triggered by earlier failures and may be inconsistent.
Step 2: Study previous commits

- Commit 9d9f780, from January 2015.
- Most recent one treating relevant suspend and resume fields.
Step 2: Study previous commits

@@ -273,9 +273,9 @@
#ifndef CONFIG_PM_SLEEP
static int as3935_suspend(struct spi_device *spi, pm_message_t msg)
+static int as3935_suspend(struct device *dev)
{
- struct iio_dev *indio_dev = spi_get_drvdata(spi);
+ struct iio_dev *indio_dev = dev_get_drvdata(dev);
@@ -293,9 +293,9 @@
static int as3935_resume(struct spi_device *spi)
+static int as3935_resume(struct device *dev)
{
- struct iio_dev *indio_dev = spi_get_drvdata(spi);
+ struct iio_dev *indio_dev = dev_get_drvdata(dev);
@@ -311,9 +311,12 @@
+static SIMPLE_DEV_PM_OPS(as3935_pm_ops, as3935_suspend, as3935_resume);
+#define AS3935_PM_OPS (&as3935_pm_ops)
#else
-#define as3935_suspend NULL
-#define as3935_resume NULL
+#define AS3935_PM_OPS NULL
#endif
@@ -441,12 +444,11 @@
 .driver = {
 ... 
+ .pm = AS3935_PM_OPS,
 },
 ... 
- .suspend = as3935_suspend,
- .resume = as3935_resume,
Studying the commit in more detail

```
@@ -441,12 +444,11 @@
 .driver = {
     .name = "as3935",
     .owner = THIS_MODULE,
+    .pm = AS3935_PM_OPS,
  },
  .probe = as3935_probe,
  .remove = as3935_remove,
  .id_table = as3935_id,
- .suspend = as3935_suspend,
- .resume = as3935_resume,
};
```

- Removes suspend and resume as required.
- The commit does other things, so some compensation seems needed for the removal.
Studying the commit in more detail

@@ -273,9 +273,9 @@
 #ifdef CONFIG_PM_SLEEP
 static int
-as3935_suspend(struct spi_device *spi, pm_message_t msg)
+as3935_suspend(struct device *dev)
 {
- struct iio_dev *indio_dev = spi_get_drvdata(spi);
+ struct iio_dev *indio_dev = dev_get_drvdata(dev);

• as3935_suspend was stored in the removed suspend field.
  – Change parameter list.
  – Change function body to compensate.

• Similar changes for as3935_resume.
Studying the commit in more detail

@@ -311,9 +311,12 @@
+ static SIMPLE_DEV_PM_OPS ( as3935_pm_ops , as3935_suspend ,
+                             as3935_resume );
+ # define AS3935_PM_OPS ( & as3935_pm_ops )
 # else
- # define as3935_suspend NULL
- # define as3935_resume NULL
+ # define AS3935_PM_OPS NULL
 # endif
@@ -441,12 +444,11 @@
 . driver = {
     ...
+     . pm = AS3935_PM_OPS ,
 },
     ...
-    . suspend = as3935_suspend ,
-    . resume = as3935_resume ,

• Constructing replacement information.
Studying the commit in more detail

@@ -311,9 +311,12 @@
+ static SIMPLE_DEV_PM_OPS (as3935_pm_ops , as3935_suspend ,
+                        as3935_resume);
+ #define AS3935_PM_OPS (& as3935_pm_ops)
+ #else
- #define as3935_suspend NULL
- #define as3935_resume NULL
+ #define AS3935_PM_OPS NULL
+ #endif
@@ -441,12 +444,11 @@
  . driver = {
      ...
+      . pm = AS3935_PM_OPS,
  },
  ...
- . suspend = as3935_suspend,
- . resume = as3935_resume,

• Constructing replacement information.

Do these changes apply to our driver?
Step 3: Updating the lms501kf03 TFT LCD panel driver

static struct spi_driver lms501kf03_driver = {
    .driver = {
        .name   = "lms501kf03",
        .owner  = THIS_MODULE,
    },
    .probe   = lms501kf03_probe,
    .remove  = lms501kf03_remove,
    .shutdown= lms501kf03_shutdown,
    .suspend = lms501kf03_suspend,
    .resume  = lms501kf03_resume,
};
Step 3: Updating the lms501kf03 TFT LCD panel driver

static struct spi_driver lms501kf03_driver = {
    .driver = {
        .name = "lms501kf03",
        .owner = THIS_MODULE,
    },
    .probe = lms501kf03_probe,
    .remove = lms501kf03_remove,
    .shutdown = lms501kf03_shutdown,
    -.suspend = lms501kf03_suspend,
    -.resume = lms501kf03_resume,
};
Step 3: Updating the lms501kf03 TFT LCD panel driver

```c
static struct spi_driver lms501kf03_driver = {
    .driver = {
        .name = "lms501kf03",
        .owner = THIS_MODULE,
        + .pm = ...,
    },
    .probe = lms501kf03_probe,
    .remove = lms501kf03_remove,
    .shutdown = lms501kf03_shutdown,
-   .suspend = lms501kf03_suspend,
-   .resume = lms501kf03_resume,
};
```
Step 3: Updating the lms501kf03 TFT LCD panel driver

```c
static int lms501kf03_suspend(struct spi_device *spi, pm_message_t mesg)
{
    struct lms501kf03 *lcd = dev_get_drvdata(&spi->dev);
    dev_dbg(&spi->dev, "lcd -> power = %d\n", lcd->power);
    return lms501kf03_power(lcd, FB_BLANK_POWERDOWN);
}
```

Change parameter list.

- Likewise for resume function.
Step 3: Updating the lms501kf03 TFT LCD panel driver

```c
static int
-lms501kf03_suspend(struct spi_device *spi, pm_message_t mesg)
+lms501kf03_suspend(struct device *dev)
{
    struct lms501kf03 *lcd = dev_get_drvdata(&spi->dev);
    dev_dbg(&spi->dev, "lcd -> power = %d\n", lcd->power);
    return lms501kf03_power(lcd, FB_BLANK_POWERDOWN);
}
```

Change parameter list.

- Likewise for resume function.
Step 3: Updating the lms501kf03 TFT LCD panel driver

```c
static int
-lms501kf03_suspend(struct spi_device *spi, pm_message_t mesg)
+lms501kf03_suspend(struct device *dev)
{
    struct lms501kf03 *lcd = dev_get_drvdata(&spi->dev);
    dev_dbg(&spi->dev, "lcd->power = %d\n", lcd->power);
    return lms501kf03_power(lcd, FB_BLANK_POWERDOWN);
}
```

Change function body.
Step 3: Updating the lms501kf03 TFT LCD panel driver

```c
static int
-lms501kf03_suspend(struct spi_device *spi, pm_message_t mesg)
+lms501kf03_suspend(struct device *dev)
{
    struct lms501kf03 *lcd = dev_get_drvdata(&spi->dev);
    dev_dbg(&spi->dev, "lcd->power = %d\n", lcd->power);
    return lms501kf03_power(lcd, FB_BLANK_POWERDOWN);
}
```

Change function body.

- Problem:
  - The example commit calls `spi_get_drvdata` on `spi`.
  - This driver calls `dev_get_drvdata` and `dev_dbg`. 
Returning to Step 2

Commit 9d9f780 was the most recent related to this issue.

Other possibilities:

- Commit 01f9326 from March 2013 contains:
  
  - struct snd_card *card = dev_get_drvdata(&spi->dev);
  + struct snd_card *card = dev_get_drvdata(dev);

- Commit eba3bfb from April 2013 contains:
  
  - dev_dbg(&spi->dev, "lcd->power = %d\n", lcd->power);
  + dev_dbg(dev, "lcd->power = %d\n", lcd->power);
Returning to Step 2

Commit 9d9f780 was the most recent related to this issue.

Other possibilities:

- Commit 01f9326 from March 2013 contains:
  
  - struct snd_card *card = dev_get_drvdata(&spi->dev);
  + struct snd_card *card = dev_get_drvdata(dev);

- Commit eba3bfb from April 2013 contains:
  
  - dev_dbg(&spi->dev, "lcd->power = %d\n", lcd->power);
  + dev_dbg(dev, "lcd->power = %d\n", lcd->power);

These examples plus commit 9d9f780 give enough information to port the driver.
Challenges

- Redundant and inconsistent compiler messages.
- A few compiler errors can require many changes.
- Multiple example commits may be needed.
  - Essential to choose the right ones:
    - Over 80 Linux kernel structure types have suspend and resume fields.
    - For our driver, need commits affecting spi driver structures.
- How to find these examples?
Challenges

- Redundant and inconsistent compiler messages.
- A few compiler errors can require many changes.
Challenges

- Redundant and inconsistent compiler messages.
- A few compiler errors can require many changes.
- Multiple example commits may be needed.

Essential to choose the right ones:
- Over 80 Linux kernel structure types have `suspend` and `resume` fields.
- For our driver, need commits affecting `spi` driver structures.

How to find these examples?
Challenges

- Redundant and inconsistent compiler messages.
- A few compiler errors can require many changes.
- Multiple example commits may be needed.
- Essential to choose the right ones:
Challenges

• Redundant and inconsistent compiler messages.

• A few compiler errors can require many changes.

• Multiple example commits may be needed.

• Essential to choose the right ones:
  – Over 80 Linux kernel structure types have suspend and resume fields.
  – For our driver, need commits affecting spi_driver structures.
Challenges

• Redundant and inconsistent compiler messages.

• A few compiler errors can require many changes.

• Multiple example commits may be needed.

• Essential to choose the right ones:
  – Over 80 Linux kernel structure types have suspend and resume fields.
  – For our driver, need commits affecting spi_driver structures.

• How to find these examples?
Proposed solution

gcc-reduce:

- Eliminate redundant gcc error messages.
- Collect complementary information from the source code.
- Focus on information needed to find porting examples.

Prequel:

- Find commits that provide example solutions to porting issues.

Future work: Generalize examples to porting rules.
Prequel

Pattern-matching based commit search

- Builds on previous work on Coccinelle.
- Notation familiar to Linux developers.

**Example:** Find commits that, at least, remove a suspend field

```c
identifier i;
expression e;
```

```c
struct spi_driver i = {
    .suspend = e,
};
```
**Results**

- c14dd26: 10% (514/762)...
- 9d9f780: 6% (189/762)...
- 3aff5b1: 6% (486/762)...
- 91debb0: 6% (498/762)...
- 01f9326: 5% (485/762)...
- c47ddc2: 5% (501/762)...
- d50d7ac: 5% (471/762)...
- ba3601a: 5% (474/762)...
- eba3bfb: 5% (473/762)...
- 26cd2a4: 5% (472/762)...

...
How is a patch query applied?

70K Linux kernel commits per year

• Impractical to extract, parse, and match all of them.

Indexing and abstraction:

• Indices:
  – words on changed lines.
  – words within 3 lines of changed lines.

• Abstract patch query to words that must be present on or near changed lines.
  – Process only the relevant commits.
  – Some risk of false negatives

• For more distant words, find relevant files in a reference version, and only take commits affecting those files.
Ease of use

Typically, no need to write patch queries for porting:

Reduced compiler error messages of few types:

- Undefined variable/function/field/type.
- Wrong number of arguments to function or macro.
- Incompatible types in assignment/initialization/argument.
- **No syntax errors:** the driver compiled with its original kernel.

Patch query templates mostly sufficient.
Evaluation: dataset

Drivers and driver files that raise porting issues (targeting Linux 4.6):

<table>
<thead>
<tr>
<th></th>
<th>forward port drivers</th>
<th>forward port driver files</th>
<th>back port drivers</th>
<th>back port driver files</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>108</td>
<td>135</td>
<td>130</td>
<td>149</td>
</tr>
<tr>
<td>2015</td>
<td>97</td>
<td>119</td>
<td>114</td>
<td>125</td>
</tr>
</tbody>
</table>

Main contributors:

<table>
<thead>
<tr>
<th></th>
<th>2013</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fwd</td>
<td>back</td>
</tr>
<tr>
<td>clk</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>iio</td>
<td>22</td>
<td>32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2013</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fwd</td>
<td>back</td>
</tr>
<tr>
<td>net</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>rtc</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>
Evaluation: performance

Prequel execution time per 2013 forward port issue:

Issue reduction achieved by gcc-reduce (all ports):
Evaluation: porting experiments

Data set:

- 13 forward ports from original 2013 version to Linux 4.6 (2016).
- 10 back ports from Linux 4.6 to original 2013 version.
- 10 forward ports from original 2015 version to Linux 4.6 (2016).
- 107 issues reported by gcc-reduce.
Evaluation: porting experiments

Data set:

• 13 forward ports from original 2013 version to Linux 4.6 (2016).
• 10 back ports from Linux 4.6 to original 2013 version.
• 10 forward ports from original 2015 version to Linux 4.6 (2016).
• 107 issues reported by gcc-reduce.

Results:

• Success for 80 issues.
  – Prequel-provided examples enabled replicating the original modifications.
• For 86% of these 80 issues, the first commit returned by Prequel gave the needed information.
Limitations

- Our approach assumes each issue instance addressed in a single commit.
  - Affects 5 issues in our experiments.
  - For example, field $x$ may be renamed as $x_{\text{new}}$ temporarily, while the old value is deprecated.
  - Prequel finds $x \rightarrow x_{\text{new}}$, but not $x_{\text{new}} \rightarrow x$.
  - Iteration a possible solution.

- Gcc-reduce may misidentify a compiler error root cause.
  - Affects 6 issues in our experiments.

- Backport patch searches may find bug fixes or conversion from driver-specific code to generic code.
  - Hard to identify the right transformation in the examples.
Limitations

- Our approach assumes each issue instance addressed in a single commit.
  - Affects 5 issues in our experiments.
  - For example, field x may be renamed as x_new temporarily, while the old value is deprecated.
  - Prequel finds $x \rightarrow x_{\text{new}}$, but not $x_{\text{new}} \rightarrow x$.
  - Iteration a possible solution.

- Gcc-reduce may misidentify a compiler error root cause.
  - Affects 6 issues in our experiments.
Limitations

- Our approach assumes each issue instance addressed in a single commit.
  - Affects 5 issues in our experiments.
  - For example, field $x$ may be renamed as $x_{\text{new}}$ temporarily, while the old value is deprecated.
  - Prequel finds $x \rightarrow x_{\text{new}}$, but not $x_{\text{new}} \rightarrow x$.
  - Iteration a possible solution.

- Gcc-reduce may misidentify a compiler error root cause.
  - Affects 6 issues in our experiments.

- Backport patch searches may find bug fixes or conversion from driver-specific code to generic code.
  - Hard to identify the right transformation in the examples.
Conclusion

• Finding example commits to guide device driver porting
  – Gcc-reduce for minimizing error messages.
  – Pattern language for searching commit histories.
  – Templates connecting error messages to patch queries.
  – 33 driver back and forward porting experiments.

• Future work:
  – Improving commit ranking algorithm.
  – Inferring changes from identified commits.

• http://prequel-pql.gforge.inria.fr/