Evolution of the Linux Kernel Variability Model

SPLC 2010

Rafael Lotufo, Steven She, Krzysztof Czarnecki @ University of Waterloo
Thorsten Berger @ University of Leipzig
Andrzej Wasowski @ IT University of Copenhagen
Contents

- Motivation
- Why Linux?
- Model’s growth
- Characterization of individual edits
- Implications for tool support
- Conclusion
Motivation

- Variability models are an essential part of software product line development

Thousands of features + Thousands of constraints → Maintenance challenge
Motivation

- Existing work on VM maintenance
  - Reasoning about feature model edits [Thum09, Janota08]
  - Feature model refactoring [Alves06]
  - Synchronizing artifacts in product lines [Kastner08, Janota08]

- Not motivated by problems faced in real projects
We investigate

- The evolution of the Linux kernel variability model from 2.6.12 to 2.6.32
- How does it grow?
- What changes?
- How?
- Reasons for edits?
- Real issues found in industry
Why Linux kernel?
Linux kernel is an SPL

- Compiles for over 20 different architectures
- Over 10000 configuration options
- Granularity
  - Subsystem (networking, cryptography)
  - Driver (wireless driver, USB)
  - Feature (suspend to RAM)
  - Feature configuration (enable freezer for suspend to RAM)
  - Small tweaks: logging, debugging, hacks
Linux kernel has an FM

- There is a variability model that can be interpreted as a feature model.
- This relationship has been studied in several papers
  - Sincero 2008, She 2010, Berger 2010
- The model is specified by the Kconfig language
Linux kernel has a configurator

- Three configuration tools
  - make config
  - make menuconfig
  - make xconfig
Linux kernel is active

- New release every 3 months
- 10000 patches per release
Linux kernel’s FM is big,
- Over 10000 features

complex,
- Most features have over 60 transitive implications

has grown
- Has doubled in size over the last 5 years
Linux is a successful multi-platform project

- Millions of users
  - Servers
  - Mobile phones
  - Desktop
  - Other devices

- Thousands of developers
  - Over 5000 developers
  - Over 500 companies
Kconfig language

- Declarative language for configuration options and their constraints

```c
config PM_SLEEP
    bool "Power Management Sleep"
    depends on SUSPEND || HIBERNATION || XEN_SAVE_RESTORE
    default y
```

- Allows for XOR groups, mandatory features, defaults
- Hierarchy is inferred from order, nesting, and dependencies
- For more info, see She 2010, Berger 2010
The model’s growth
Snapshot of release 2.6.28 for X86 architecture

- > 5400 features
- > 9000 constraints
Size

![Graph showing Size vs Revision]

- **Total** vs **Revision**

  - **X86 features**
  - **All features**

---

GSD Lab @ University of Waterloo

SPLC 2010
Direction of growth

- Growth in breadth, not depth

Cryptographic API
- Serpent cipher algorithm
+ Twofish cipher algorithms (i586)

Device Drivers
- Graphics support
  ... Sound card support
- Advanced Linux Sound Architecture
  - MIPS sound devices
  - USB sound devices
    - Tascam US-122L USB driver
+ USB Audio/MIDI driver
Constraints

![Graph showing the increase in number of constraints over revisions.](image)
Characterizing individual edits
Reasons for edits

- Sampled 200 patches to identify relevant classes
- Classified another set of 200 patches
- Look at commit logs and diffs in Git repository
- Patches are self-contained and complete
- Motivation for patches is clearly stated in commit logs
New functionality

- Edit C code, Makefile, add configuration option
- 87% of feature additions as leaves
- Depth does not increase
[SCSI] bnx2i: Add bnx2i iSCSI driver.

New iSCSI driver for Broadcom BNX2 devices. The driver interfaces with the CNIC driver to access the hardware.

Signed-off-by: Anil Veerabhadrappa <anilgv@broadcom.com>
Signed-off-by: Michael Chan <mchan@broadcom.com>
Signed-off-by: Mike Christie <michaelc@cs.wisc.edu>
Signed-off-by: James Bottomley <James.Bottomley@HansenPartnership.com>

drivers/scsi/Kconfig
drivers/scsi/Makefile
drivers/scsi/bnx2i/57xx_iscsi_constants.h [new file with mode 0644]
drivers/scsi/bnx2i/57xx_iscsi_hsi.h [new file with mode 0644]
drivers/scsi/bnx2i/Kconfig [new file with mode 0644]
drivers/scsi/bnx2i/Makefile [new file with mode 0644]
drivers/scsi/bnx2i/bnx2i.h [new file with mode 0644]
drivers/scsi/bnx2i/bnx2i_hwi.c [new file with mode 0644]
drivers/scsi/bnx2i/bnx2i_init.c [new file with mode 0644]
drivers/scsi/bnx2i/bnx2i_iscsi.c [new file with mode 0644]
drivers/scsi/bnx2i/bnx2i_sysfs.c [new file with mode 0644]
Build fix

- **Cause:** dependency not in sync with code
- **Commit logs:**
  - “as far as I can tell…”
  - “after carefully examining the code…”
  - “it’s a nightmare working out why…”
- Indicates lack of support for reasoning and synchronizing dependencies with code
[ARM] pxa: corgi backlight driver should not select ssp drivers

Resolves build errors with eseries and magician defconfigs (which make use of the corgi backlight driver.)

Signed-off-by: Russell King rmk+kernel@arm.linux.org.uk

diff --git a/drivers/video/backlight/Kconfig b/drivers/video/backlight/Kconfig

config BACKLIGHT_CORGI
         tristate "Generic (aka Sharp Corgi) Backlight Driver (DEPRECATED)"
         depends on BACKLIGHT_CLASS_DEVICE
-         select CORGI_SSP_DEPRECATED
         default n
Clean-up/maintainability

- Help text, comments, feature rename,
- Constraint refactoring/simplification
- Hierarchy refactoring
  - Mainly by removing parent features
  - Features move in groups

![Graph showing the number of patches for different reasons]

- New functionality
- Build fix
- Clean-up/maintainability
- Adherence to changes in C code
- Change variability
- Retiring obsolete feature

Number of patches
x86: simpler SYSVIPC_COMPAT definition

X86_64 part is entirely redundant.

Signed-off-by: Alexey Dobriyan <adobriyan@gmail.com>
Signed-off-by: Ingo Molnar <mingo@elte.hu>

diff --git a/arch/x86/Kconfig b/arch/x86/Kconfig

    config SYSVIPC_COMPAT
       def_bool y
-       depends on X86_64 && COMPAT && SYSVIPC
+       depends on COMPAT && SYSVIPC

    endmenu
Synchronize Dependencies

- Changes to constraints in code and in feature model
- Typically motivated by code refactoring, bug fixes
ALSA: sound/core/pcm_timer.c: use lib/gcd.c

Make sound/core/pcm_timer.c use lib/gcd.c

Signed-off-by: Florian Fainelli <florian@openwrt.org>
Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
Signed-off-by: Takashi Iwai <tiwai@suse.de>

diff --git a/sound/core/pcm_timer.c b/sound/core/pcm_timer.c
   #include <linux/time.h>
+  #include <linux/gcd.h>
   #include <sound/core.h>
   #include <sound/pcm.h>
   #include <sound/timer.h>

diff --git a/sound/core/Kconfig b/sound/core/Kconfig
config SND_PCM
    tristate
      select SND_TIMER
+     select GCD
Change variability

- Add/remove configurations from feature model
  - Functional code already exists
  - Add/remove/change configuration options and dependencies
  - Complex: changes in 200 constraints in 44 files
[PATCH] BLOCK: Make it possible to disable the block layer
[try #6]

Make it possible to disable the block layer. Not all embedded devices require it, some can make do with just JFFS2, NFS, ramfs, etc - none of which require the block layer to be present.

```
config MTD_BLOCK
    tristate "Caching block device access to MTD devices"
    - depends on MTD
    + depends on MTD && BLOCK

config MTD_BLOCK_RO
    tristate "Readonly block device access to MTD devices"
    - depends on MTD_BLOCK!=y && MTD
    + depends on MTD_BLOCK!=y && MTD && BLOCK
```
Feature retirement

- Formal schedule of code and feature retirement
- Reasons for retirement
  - No maintainers
  - Obsolete code
[WATCHDOG] the scheduled removal of the i8xx_tco watchdog driver

This patch contains the scheduled removal of the i8xx_tco watchdog driver.

Signed-off-by: Adrian Bunk <bunk@stusta.de>
Signed-off-by: Wim Van Sebroeck <wim@iguana.be>

diff --git a/drivers/char/watchdog/Kconfig b/drivers/char/watchdog/Kconfig

- config I8XX_TCO
-  tristate "Intel i8xx TCO Timer/Watchdog"
-  depends on WATCHDOG && (X86 || IA64) && PCI
-  default n
Implications for tool support

- Edits to hierarchy
  - Move groups
  - Feature removal with minimal impact
- Synchronize dependencies with external model
- Simplifying constraints
- Edits to constraints in batch
Conclusion

- Feature models are a feasible abstraction for large, complex, mature software systems

- Tool support is needed to aid maintenance and evolution of variability models
  - Hierarchy edits
  - Feature removal
  - Batch edits
  - Synchronizing dependencies
  - Simplifying constraints
Thank you!
Questions?