Git Bisect Your Way to Fame and Fortune

by Trevor E Cordes
(c) 2015
Presented to Manitoba UNIX User Group
June 9, 2015

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

Presentation Method

- Lots to cover
- Gloss over details / commands
- Download and reference the detailed slides later
- Story
- Template for attacking your own bugs
- Start to finish

OS Upgrades Are Fun

- NOT!
- My MythTV / fileserver box
- From Fedora 19 to 21
- All seemed well

2 Days Later...

- MythTV records TV shows off cable: free Tivo
- But better
- Why is MythTV recording the wrong channel?
- Everything is channel 2... lame
- Formula 1 season is about to start (go McLaren!)
- Downton Abbey season isn’t done yet!
- Gasp!

There Goes WAF

- Just when I get MythTV fully Wife Acceptance Factored...
- Eventually broke out the ol’ VHS tape and used a VCR
- Knew this bug wasn’t getting solved quickly

The Bug Is...

- MythTV uses LIRC infrared tool
- To change channels on the cable box
- Via an IR blaster plugged in via USB
- LIRC blasting seems broken
- Test: irsend SEND_ONCE dct700 info
- timeout

Solving The Bug Part 0

- March 10
- Report bug on RedHat Bugzilla (rhbz)
- https://bugzilla.redhat.com/show_bug.cgi?id=1200353

Debug Mode

- Many programs have a debug/verbose mode
- lircd can be run in debug mode

strace

- Can run strace against any program
- See where it is hanging:
- strace irsend
- strace lircd
- lircd is hanging on write to /dev/lirc0

Cheat With Wrong OS Ver Pkgs

- Installed Fedora 22’s lirc rpm
- Installed Fedora 20’s lirc rpm
- Doesn’t always work: package / library conflicts
- Got lucky
- But bug remains

Try Older Kernel

- Fedora 20 original kernel
- v3.11
- On F21’s userspace
- Ick
- But bug disappears!

Bug Is In Kernel You Git!

- So let’s use git
- git is the source code control system used by Linus Torvalds and the kernel devs

30s Git Primer

- Source code control / versioning system
- Taking the world by storm
- Like: RCS, CVS, subversion, mercurial
- But no central or master server or repo
- Everyone works on their own copy
- Everyone is equal, even Linus
- Stores full files, not diffs
- Easy merges (push, pull) put it all together

Handy Git Cheatsheet

- Update your repo to the very latest version
- git fetch origin
- Reset your repo to start state after tinkering
- git reset --hard origin/master
- Reset to start state after a bisect
- git bisect reset

Git Yourself The Kernel Source

- git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git linux-git
- Shockingly fast...

What Source File?

- lsmod
- Unplug the USB blaster, replug
- What module got loaded?
- mceusb
- find . | grep mceusb
- drivers/media/rc/mceusb.c

Simple Driver Hacking

- Like all programmers...
- Have a bug, add some prints
- printk(KERN_DEBUG "reached point 0\n");
- Output will go to dmesg or /var/log/messages (and friends)

Compile Just One Module

- Super handy!
- Create Makefile in same dir as mceusb.c

```
obj-m += mceusb.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Compile mceusb.c

- make all
- mv /lib/modules/4.0.4-202.fc21.x86_64/kernel/drivers/media/rc/mceusb.ko.xz /lib/modules/4.0.4-202.fc21.x86_64/kernel/drivers/media/rc/mceusb.ko.xz.bak
- cp mceusb.ko /lib/modules/4.0.4-202.fc21.x86_64/kernel/drivers/media/rc/
- rmmod mceusb
- modprobe mceusb
- May have to deal with dependent modules
- No reboot required!

So What Changed?

- git log drivers/media/rc/mceusb.c
- And: git log -p ...
- DEMO
- git identifies every commit via a unique hash
- Often shortened to unique prefix
- 6 to 8 characters

What Now? Suspicious!

- Look for suspicious commits
- English text and C code
- Try to reverse those changes
- Using diff/patch
- Or manually if they’re small
- But...

Time Travel

- Why not just “go back in time” in the git tree
- Incompatible commits
- Changes in parent / support libraries
- Refactoring
- Often very rapidly a dead end

Start Your Emails

- git log showed you committer email addresses
- Get a general idea of who is working on the relevant code:
  - Temporal: around the time where you think the buggy commits occurred
  - Quantity: who is doing the most commits to this code
- I emailed 2 guys and got 1 responding

Start Your Bisect

- No scalpels needed
- What is git bisect?
  - Binary search
  - Just say Good or Bad
- Not great for intermittent, non-deterministic, hard to reproduce bugs
- Luckily my bug is none of those things

Pentium-D For Depressingly Slow

- Kernel bisection can require build of entire kernel
- Dozens of times!
- Each build: 4 hours on my Pentium-D (with SSD!)
- Can we leverage distro (Fedora) prepackaged binaries (RPMs) to reduce time?
- Yes...

Fedora Koji

- Koji = Fedora's automated build system
- Keeps copies of nearly every rpm built
- Amazingly useful resource
- Great web interface
- http://koji.fedoraproject.org/koji/packageinfo?buildStart=0&packageID=8&buildOrder=-completion_time&tagOrder=name&tagStart=0#buildlist
- DEMO

RPMs From Koji Narrow Search

- Downloading prebuilt kernel RPMs from Koji
- Doing my own version of “git bisect”
- Pseudo-binary-searching Koji

works:

3.14.0-1.fc21.i686+PAE

3.16.0-1.fc21.i686+PAE

3.16.3-302.fc21.i686+PAE

broken:

3.17.0-0.rc7.git1.1.fc22.i686+PAE

3.17.0-301.fc21.i686+PAE

3.17.1-302.fc21.i686+PAE

Big Version Problem

- git / kernel use commit hashes (35a19c)
- kernel uses occasional major release numbers to 2 levels of digit groups (3.17)
- Fedora uses complex 3rd and 4th level (3.17.0-301.fc21.i686+PAE)
- No easy way to convert from Fedora version to git hash
- Worse: Fedora-specific patches, etc
- Heard whispered that Ubuntu might provide a map

Koji Version To Git

- So my Koji “bisect” only told me:
- But between git version label v3.16 – v3.17
- git prefixes major version release numbers with “v”
- Now we can use git bisect...

Git Bisect Try #0

- Think bug is in mceusb
- To save time/bisects
- Bisect on just one file, or dir, or dir tree

git reset --hard origin/master

git bisect start drivers/media/rc/mceusb.c

git bisect good v3.16

git bisect bad v3.17

Bisecting: 1 revision left to test after this (roughly 1 step)

[c5540fbb9de39ceec108a889133664a887c2f55a] [media] rc-core: remove protocol arrays

- DEMO

2 Commits On mceusb?

- Just use the manual single-module compile idea above after git checks-out
- But those either don’t compile (support library changes)
- Or they still present the bug!
- Walk up a dir level...

Git Bisect Try #1

- Guessing bug is in a file in drivers/media/rc

git reset --hard origin/master

git bisect reset

git bisect start drivers/media/rc/

git bisect good v3.16

git bisect bad v3.17

- Still single modules (~3 needed)
- Use aforementioned single-module compile method
- All were bad... failure

Git Bisect Try #2

- Move up a dir
- Do complete kernel builds this time

git bisect start drivers/media

git bisect good v3.16

git bisect bad v3.17

Bisecting: 196 revisions left to test after this (roughly 8 steps)

How Do I Fedoraize?

- I’m no kernel hacker
- I don’t want to answer kernel config questions to make .config file
- I want this done fast
- I want this done the Fedora way
- I want the resulting kernel to match Fedora’s stock kernels as closely as possible

Ignatenkobrain

- Fedora guy already solved this:
- http://fedoraproject.org/wiki/User:Ignatenkobrain/Kernel/Bisection
- Helps you do bisection the “Fedora Way”
- Minimal fuss
- Simple

Isn’t Life Grand

$ cd ~

$ git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git

$ git clone http://github.com/ignatenkobrain/kernel-package.git

$ cd linux

$ git bisect start

$ git bisect bad v3.11-rc1

$ git bisect good v3.9

$ ~/kernel-package/kernel-package.py

Sucks To Be Me

```
Traceback (most recent call last):
  File "../kernel-package/kernel-package.py", line 342, in <module>
    main()
  File "../kernel-package/kernel-package.py", line 330, in main
    options = Options(WORK_DIR)
[...]
  File "/usr/lib/python2.7/site-packages/gitdb/db/pack.py", line 71, in _pack_info
    index = item[2](sha)
  File "/usr/lib/python2.7/site-packages/gitdb/pack.py", line 492, in sha_to_index
    return PackIndexFile_sha_to_index(self, sha)
ValueError: Couldn't obtain fanout table
Exit 1
```

Bug #2 … And Counting

- https://bugzilla.redhat.com/show_bug.cgi?id=1090186
- Doh!
- So “easy” Fedora Way is out

Manual “Fedora Way”

- Let Fedora make the .config file for you
- Then steal their .config

```
rpmdev-setuptree

yumdownloader --source kernel

yum-builddep --downloadonly kernel

yum-builddep kernel

cd ~/rpmbuild

# have rpm "prep" the build

cd ~/rpmbuild/SPECS

# normal pkgs:  rpmbuild -bp --target=`uname -m` kernel.spec

# to speed up kernel prep disable what you don't use

rpmbuild -bp --without xen --without kdump --without debug --without debuginfo --target=`uname -m` kernel.spec

cp ~/rpmbuild/BUILD/linux-*/.config ~/linux
```

Build Your Kernel

- cd ~/linux; make bzImage

```
  HOSTCC  scripts/basic/fixdep
fixdep: error fstat'ing depfile: scripts/basic/.fixdep.d: Value too large for defined data type
scripts/Makefile.host:118: recipe for target 'scripts/basic/fixdep' failed
make[2]: *** [scripts/basic/fixdep] Error 2
/data/Tmp/Linux-git/linux-git/Makefile:443: recipe for target 'scripts_basic' failed
make[1]: *** [scripts_basic] Error 2
  SYSTBL  arch/x86/syscalls/../include/generated/asm/syscalls_32.h
  SYSHDR  arch/x86/syscalls/../include/generated/uapi/asm/unistd_32.h
  SYSHDR  arch/x86/syscalls/../include/generated/uapi/asm/unistd_64.h
  SYSHDR  arch/x86/syscalls/../include/generated/uapi/asm/unistd_x32.h
  HOSTCC  scripts/basic/fixdep
fixdep: error fstat'ing depfile: scripts/basic/.fixdep.d: Value too large for defined data type
scripts/Makefile.host:118: recipe for target 'scripts/basic/fixdep' failed
make[1]: *** [scripts/basic/fixdep] Error 2
Makefile:443: recipe for target 'scripts_basic' failed
make: *** [scripts_basic] Error 2
```

Bug #3 ... And Counting

- scripts/Makefile.host:118: recipe for target 'scripts/basic/fixdep' failed
- fixdep has a bug!
- It breaks when used on a 32-bit system with a 64-bit inode filesystem like XFS and 64-bit inodes are present
- Guess what my system has?
- #@&^%!%*

Gimme Space 40G SSD

- My system has a 40GB SSD as boot, root, swap
- Had to clear up ~10GB for kernel + build
- fixdep now works
- Bonus: SSD speeds instead of RAID6 rust speeds

Build Your Kernel (Redux)

- make bzImage && make modules && make modules_install && make install
- Automagically makes the initramfs for you (dracut?)
- Just works
- reboot
- rm -rf previous attempts in /lib/modules and /boot (but not rpm ones!)

Test: Naughty Or Nice?

- irsend SEND_ONCE dct700 info
- git bisect bad
- git bisect good
- Eventually...
- first bad commit: [ae045e2455429c418a418a3376301a9e5753a0a8] Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net-next

What Did I Just Do?

- git bisect log

git bisect start drivers/media

good: [19583ca584d6f574384e17fe7613dfaeadcdc4a6] Linux 3.16

git bisect good 19583ca584d6f574384e17fe7613dfaeadcdc4a6

good: [19583ca584d6f574384e17fe7613dfaeadcdc4a6] Linux 3.16

git bisect good 19583ca584d6f574384e17fe7613dfaeadcdc4a6

good: [67dd8f35c2d8ed80f26c9654b474cffc11c6674d] Merge branch 'v4l_for_linus' of git://git.kernel.org/pub/scm/linux/kernel/git/mchehab/linux-media

git bisect good 67dd8f35c2d8ed80f26c9654b474cffc11c6674d

[...]

good: [c835a677331495cf137a7f8a023463afd9f032f8] net: set name_assign_type in alloc_netdev()

git bisect good c835a677331495cf137a7f8a023463afd9f032f8

good: [8fd90bb889635fa1e7f80a3950948cc2e74c1446] Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net

git bisect good 8fd90bb889635fa1e7f80a3950948cc2e74c1446

first bad commit: [ae045e2455429c418a418a3376301a9e5753a0a8] Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net-next

That Commit Seems Harmless

- If the commit git bisect says is bad looks OK
- AND
- You are limiting your bisect to a subset (dir)
- Remember there can be, hiding between seemingly adjacent commits in the subdir, many commits outside that subdir
- Your bug is probably outside the subdir

Git Bisect Try #3

- Use the knowledge from Try #2
- git bisect log showed at the end:
- good 8fd90b
- bad ae045e
- So now, NO PATH on start!

git reset --hard origin/master

git bisect reset

git bisect start

git bisect good 8fd90b

git bisect bad ae045e

Twist Ending

- first bad commit: [166afb64511eef08e13331b970c44fe91cea45ef] ktime: Sanitize ktime_to_us/ms conversion
- In include/linux/ktime.h
- Huh?

Git Reveals All

- **git diff 166afb64511^ 166afb64511**

```
diff --git a/include/linux/ktime.h b/include/linux/ktime.h

+#if BITS_PER_LONG < 64

+extern u64 ktime_divns(const ktime_t kt, s64 div);

+#else /* BITS_PER_LONG < 64 */

+# define ktime_divns(kt, div)  (u64)((kt).tv64 / (div))

+#endif


 static inline s64 ktime_to_us(const ktime_t kt) {
```

```
-       struct timeval tv = ktime_to_timeval(kt);

-       return (s64) tv.tv_sec * USEC_PER_SEC +
tv.tv_usec;

+       return ktime_divns(kt, NSEC_PER_USEC);

 }

 static inline s64 ktime_to_ms(const ktime_t kt) {

-       struct timeval tv = ktime_to_timeval(kt);

-       return (s64) tv.tv_sec * MSEC_PER_SEC +
tv.tv_usec / USEC_PER_MSEC;

+       return ktime_divns(kt, NSEC_PER_MSEC);
```

Ultimate Test: rpmbuild

- To ensure it works the “Fedora Way”
- Make a patch to undo this simple, small commit
- yumdownloader --source kernel
- rpm -i kernel*.src.rpm
- Add it to the latest Fedora kernel SPEC file from src.rpm and rpmbuild

Make The Patch

```
rm -rf /tmp/S
mkdir /tmp/S
# To work with the after-fedora-patches-applied source (probably what you want), have rpm "prep" the build:
cd ~/rpmbuild/SPECS
# limit what I don't need for build
rpmbuild -bp --without xen --without kdump --without debug --without debuginfo --target=`uname -m` kernel*
mv ~/rpmbuild/BUILD/kernel* /tmp/S/a
cp -a /tmp/S/a /tmp/S/b
# if kernel, reduce the dirs by a level
mv /tmp/S/a/linux*/* /tmp/S/a/linux*/.[A-z]* /tmp/S/a
mv /tmp/S/b/linux*/* /tmp/S/b/linux*/.[A-z]* /tmp/S/b
rm -rf /tmp/S/a/linux* /tmp/S/a/vanilla*
rm -rf /tmp/S/b/linux* /tmp/S/b/vanilla*
# edit the files you want in the /tmp/B source tree
cd /tmp/S/b
# make the patch
(cd /tmp/S; diff -uNr a b > ~/rpmbuild/SOURCES/patchfile)
```

Add Patch to Specfile

* specfile format may have changed a bit; basically you need to reference your patch in 2 diff places in the specfile, just do what the other patches have done

add the patch to the spec file

after SourceX: directives add

Patch1: patchfile

then in the %prep area before the %build add a matching line:

%patch -p 1

could also change version #, but I found that tedious, I just used same version

Build And Install RPM

cd ~/rpmbuild

nice -19 rpmbuild -ba SPECS/specfile

yum reinstall RPMS/kernel*PAE*

(or whatever version you need)

Ensure your grub2 default kernel is the one above

reboot

Hooray!

- Bug is gone in kernel built and installed the “Fedora Way” exactly as Koji would build it
- With all Fedora patches
- With all Fedora custom .config
- Success!
- March 23: two weeks in
- Done yet?...

Now For A Real Fix

- Don’t want to jump through hoops every time a new kernel update is released
- Need to get a real fix made
- And get it in mainline kernel

LKML

- Post to the Linux Kernel Mailing List
- Made a kernel.org bugzilla ticket
- Crickets

Getting Attention

- After no LKML response
- Kept updating RHBZ, sounding desperate
- Josh Boyer of RedHat eventually replied:

"LKML is kind of a dumping ground for patches and if you didn't have the right maintainers CC'd it likely didn't even get read."

Re-Post

- On advice of Josh
- Resent LKML email
- With CC:’s he recommended
- Specific developers

Very Helpful

- The CC’d guys immediately answered
- Provided debug (more printk’s) patches
- Then testing patches
- Then final patches
- Easily compiled/tested following aforementioned rpmbuild instructions
- https://lkml.org/lkml/2015/4/30/5
- https://lkml.org/lkml/2015/5/8/681

Aid The Process

- When a patch is posted to LKML, test it
- Reply with full quoted text
- Right after the > Reported-by:
- Add a new line confirming your test:
- Tested-By: Trevor Cordes <trevor@tecnopolis.ca> [runtime test i686-PAE]
- Automated system tracks all these headers

My Hunch Was Right

- The LKML debug patch output this:
- JDB: ktime_to_us: -20157485 -> divns 18446744073689394 != old method: -20158
- Proof that:
  - Negative numbers passed in
  - Causing two’s-complement signed to unsigned conversion bug
  - University is handy!

Upshot

- The result from the ktime function
- Probably used for a sleep/delay by lirc
- Negative sleep = no sleep = OK
- Sleep of 18446744073689394ns very bad
- In 584 years the sleep would return
- I’m not that patient

Final Patch & Commit

- 2 weeks later
- commit f7bcb70ebae0dcdb5a2d859b09e4465784d99029

ktime: Fix ktime_divns to do signed division

It was noted that the 32bit implementation of ktime_divns() was doing unsigned division and didn't properly handle negative values.

And when a ktime helper was changed to utilize ktime_divns, it caused a regression on some IR blasters.

See the following bugzilla for details: https://bugzilla.redhat.com/show_bug.cgi?id=1200353

This patch fixes the problem in ktime_divns by checking and preserving the sign bit, and then reapplying it if appropriate after the division, it also changes the return type to a s64 to make it more obvious this is expected.

Nicolas also pointed out that negative dividers would cause infinite loops on 32bit systems, negative dividers is unlikely for users of this function, but out of caution this patch adds checks for negative dividers for both 32-bit (BUG_ON) and 64-bit(WARN_ON) versions to make sure no such use cases creep in.

Headers: I’m Famous

Fixes: 166afb64511e 'ktime: Sanitize ktime_to_us/ms conversion'

Reported-and-tested-by: **Trevor Cordes <trevor@tecnopolis.ca>**

Signed-off-by: John Stultz <john.stultz@linaro.org>

Acked-by: Nicolas Pitre <nicolas.pitre@linaro.org>

Cc: Ingo Molnar <mingo@kernel.org>

Cc: Josh Boyer <jwboyer@redhat.com>

Cc: One Thousand Gnomes <gnomes@lxorguk.ukuu.org.uk>

Cc: <stable@vger.kernel.org>

Link: http://lkml.kernel.org/r/1431118043-23452-1-git-send-email-john.stultz@linaro.org

Signed-off-by: Thomas Gleixner <tglx@linutronix.de>

Final Patch (abbr.)

```
 -extern u64 __ktime_divns(const ktime_t kt, s64 div);
-static inline u64 ktime_divns(const ktime_t kt, s64 div)
+extern s64 __ktime_divns(const ktime_t kt, s64 div);
+static inline s64 ktime_divns(const ktime_t kt, s64 div) {
+       /* Negative divisors could cause an inf loop, so bug out here. */
+       BUG_ON(div < 0);
        if (__builtin_constant_p(div) && !(div >> 32)) {
-               u64 ns = kt.tv64;
-               do_div(ns, div);
-               return ns;
+               s64 ns = kt.tv64;
+               u64 tmp = ns < 0 ? -ns : ns;
+               do_div(tmp, div);
+               return ns < 0 ? -tmp : tmp;
        } else { return __ktime_divns(kt, div); }
 }
```

Now For Fedora

- Fedora took up the kernel version that had the commit 1 week later, on May 28
- Into Fedora testing repo
- Easy to test:
- yum --enablerepo=updates-testing update kernel
- reboot

Aid Fedora

- Karma! Fedora Bodhi
- Essentially a +1/-1 system
- I can test and report success by “leaving karma”
- Command line or web
- RHBZ automatically provides direct link
- https://admin.fedoraproject.org/updates/FEDORA-2015-9127/kernel-4.0.4-202.fc21?_csrf_token=3e28dbc9c4b0575e3c37583afc9b28434166a598#
- Click “add comment” – “solves my bug”

Fedora Updates

- 4 days after Bodhi test build
- Update pushed to normal Fedora Updates
- For one and all!

There!

See, It’s THAT EASY

Lessons Learned #0

- Distro builds (RPMs) are of limited usefulness because of version mismatches
- Mostly because of git
- Non-git programs might be easier

Lessons Learned #1

- Bisecting a subset is risky
- Might end up doing a whole-tree bisect anyhow
- If bisecting but not building all, ensure your starting “good” and “bad” are really so, don’t assume

Lessons Learned #2

- CC: people on LKML manually
- Ask others to suggest likely candidates
- Don’t spam
- Don’t nag

Lessons Learned #3

- Kernel.org bugzilla was completely useless
- No one CC’d, replied, or touched it
- Doubt anyone saw it
- Distro bugzilla far more useful
- But not always a panacea