

A day in the life of a Linux kernel hacker... Why who does what when and how!

John W. Linville

Linux Symposium

17 July 2009

Who am I?



Why is this interesting?

Why is this talk worth an hour of my life?

- Linux is a huge project...and it works!
- Are you (or might you be) a user with problems?
- ...a commercial developer?
- ...a community developer?
 - Scratch an itch...
 - Save the world?
 - Resume builder!

What is so different?

What makes Linux development different from traditional software development?

- Profit not (necessarily) the main motive
- No inherent authority (not even Linus!)
- Meritocracy
 - Market for usefulness
 - Code talks!

People

The Linux community is comprised of people from all over the world...

- Wide variety of motivations
- A number of different roles

Motivations

Why do people get involved?

- Personal “itch” to scratch
- Internal/commercial project
- Work for hire
- Sponsorship
- Altruism

Roles

What jobs do people perform in the community?

- Bug reporter
- Tester
- Coder
- Reviewer
- Maintainer
- Technical Writer
- Journalist

Tools

A variety of tools make things possible...

- Communications tools facilitate information flow
- Development tools manage changesets and code distribution

Communications Tools

Communication is key!

- E-mail
- Bugzilla, etc.
- IRC
- Wikis, etc.
- Gitweb
- Patchwork
- What is missing...?

Development Tools

Preparing and posting patches...

- mutt (or other non-braindead MUA)
- checkpatch.pl
- Sparse
- Git

Git

Git is a distributed revision control system

- Distributed means no central repository
 - No central authority!
 - Easier offline usage
 - Easy to fork a project
- Really good at merging
 - Coordination only needed “after the fact”
 - Easier to rejoin (or refresh) forked projects
- Structured around commits (i.e. patches)
 - Tools for identifying problem commits (i.e. git bisect)
 - Tools for restructuring branches w/ specific commits

Patches

Fundamental unit of work is the patch...

- Identifies your exact set of changes
- Encapsulates changes to all modified files
- Resilient across changes to underlying files

Patch Example

```
diff --git a/drivers/net/wireless/airo.c b/drivers/net/wireless/airo.c
index fc4322c..0c7aa61 100644
--- a/drivers/net/wireless/airo.c
+++ b/drivers/net/wireless/airo.c
@@ -4686,7 +4686,7 @@ static int proc_stats_rid_open( struct inode *inode,
     StatsRid stats;
     int i, j;
     __le32 *vals = stats.vals;
- int len = le16_to_cpu(stats.len);
+ int len;

     if ((file->private_data = kzalloc(sizeof(struct proc_data ), GFP_KERNEL)) == NULL)
         return -ENOMEM;
@@ -4697,6 +4697,7 @@ static int proc_stats_rid_open( struct inode *inode,
     }

     readStatsRid(apriv, &stats, rid, 1);
+ len = le16_to_cpu(stats.len);

     j = 0;
     for(i=0; statsLabels[i]!=(char *)-1 && i*4<len; i++) {
```

Identify A Need

Identifying a development need...

- Bug report
 - Mailing list
 - Bugzilla
 - IRC
- External project requirement
- Some other OS is doing it...
- “Wouldn’t it be cool if...?”

Development Cycle

Iterative process once real development begins...

- Post
- Review
- Revise
- Repeat

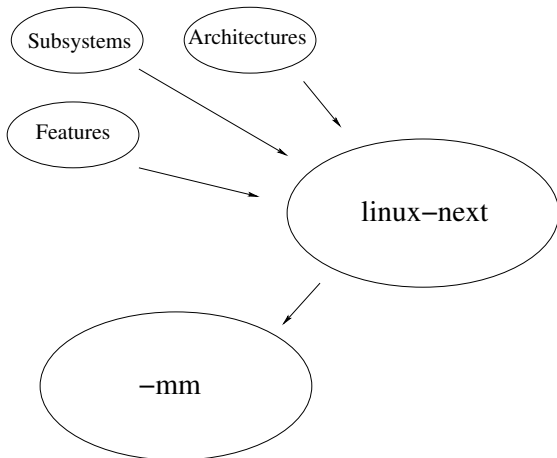
Above is unavoidable, so don't develop in a cave!

Source Trees

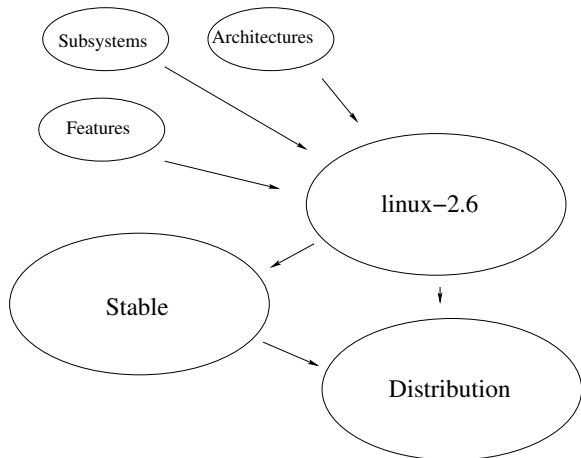
Once a patch is acceptable, it moves through a sequence of source trees...

- Maintainers
 - Subsystems (e.g. networking, SCSI, PCI, etc)
 - Features (e.g. realtime, SELinux, etc)
 - Architectures (e.g. MIPS, SPARC, Blackfin, etc)
- linux-next / -mm
- linux-2.6
- Stable
- Distribution

Development Cycle



Release Cycle



Distributions

Distribution kernel processes have different influences than upstream

- Community distros (Fedora, Debian, Gentoo, etc.)
 - Less review – trusted committers
 - Emphasis on bug fixing and stability, but...
 - Some willingness for experimental features
- Enterprise distros (RHEL, SLES, etc.)
 - Enable customer-driven features
 - Priorities driven by sales and marketing concerns
 - Long-term stability is foremost importance

The Staging Tree

“It’s too hard to get code into Linux!”

- Looser standards for inclusion (i.e. must compile)
- Limited community support
- Intended to provide reference material and/or base for porting
- Not the end goal!

How do you get involved?

Join our community!

- Kernel Newbies (<http://kernelnewbies.org/>)
- The Linux Driver Project (<http://www.linuxdriverproject.org/>)
- Just jump in! (i.e Shut-up and code!)
 - Linux Device Drivers (<http://lwn.net/Kernel/LDD3/>)
 - Understanding The Linux Kernel

Questions?



Contact

Feel free to contact me!

- Email linville@tuxdriver.com
 - ...@redhat.com
 - ...@gmail.com
 - ...@kernel.org
- IRC linville on FreeNode, OFTC, and LinuxNET
- Facebook as “John W. Linville”

Slides available:

<http://www.kernel.org/pub/linux/kernel/people/linville/ls2009/>