Finding bug fixes in mainline

May 2007

Andi Kleen, SUSE Labs
ak@suse.de

Laziness is a virtue!

- Don’t replicate work others did

Overview

- Most work is done in mainline
- Good idea to check if the bug is already fixed
  - and can be just backported

Read-only git I - theory

- A git branch is a collection of changes
  - Each has a commitid
- Distributed
  - But we can get it from git.kernel.org
- Linus tree
  - "mainline"
- Other trees
  - -mm tree (not using git)
    - Available on ftp.kernel.org (or /mounts/mirror/kernel/people/akpm/)
  - maintainer trees
  - stable tree

Read-only git II - useful commands

- http://git.kernel.org/ (gitweb)
  - Preferred for bugzilla because most people don’t use git
- git clone ...
- git pull -u
- git log
- git log dir/file.c
- git show ...
- git blame dir/file.c

Read-only git III - git bisect

- Binary search for buggy commit
  - Only gives you the commit
  - Still have to figure out why the commit broke it
  - Keep notes

- git bisect start
- git bisect good/bad
- git bisect reset
- git bisect visualize
- man git-bisect

More bisect

- Can be also done with quilt for patch series
  - e.g. to find which suse patch broke things
  - Using quilt push / pop
  - But you need to keep track of state manually
- Always double check end result

You got an oops

- Does mainline fix it?
- Identify subsystem
  - e.g. driver or slab or TCP etc.
  - Based on function name or stack trace
  - Best to be as specific possible
- git log dir/filename.c
  - Check changelog
  - Find commit id
- git log | grep -i ... | less
  - Check global changelog quickly

You didn’t find it?

- Has code moved?
  - Often similar function still exists in other file
  - Try grep -r "code pattern"
  - If algorithm changed (rarely) you have to backport the hard way
- Check changelog of file
  - Should have a changeset describing the reorganization

Slow but systematic method

- Can you reproduce it reliably with suse kernel / mainline kernel?
  - Reverse bisect for good commit may help
  - Exchange good/bad
  - Works even without subsystem

How to backport a patch

- Often it just works with minor adjustments
- When a called function name changed
  - Usually has a semantic reason, don’t just replace call
  - Check changelogs and other examples