

Ubuntu Kernel Development Process

Pete Graner
Kernel Manager
pgraner@canonical.com

UCLUG 9 Feb 2010



Who Has the Ball?

- Every release has an assigned Kernel Release Manager (KRM)
- KRM is responsible for ensuring a Kernel is ready for each milestone and release
- Additional Responsibilities:
 - Release planning & Reporting
 - Git tree management
 - Kernel upload to the buildsystem
 - Kernel delivery

Beginnings

- Kernel is started at kernel freeze of the previous release
- This involves:
 - Creating a new repository from the release branch
 - We usually do not yet know the name
 - Publishing in a development PPA
- Parallel development
 - Tagging the fork (Ubuntu-lucid-base/Ubuntu-lucid-sync)

First Steps

- Ubuntu current delta review and cleanup
 - Fold down of the debian delta
 - Fold down of the Ubuntu delta and drivers
- Pulling the Ubuntu delta up to the chosen kernel
 - Iteratively pulling patches up to each -rcN release
 - Dropping patches already in upstream
 - Note: rebase tree - one owner
- Updating the configuration
 - Selecting values for new options

Focus & Direction

- Pre Ubuntu Developer Summit (UDS)
 - Selecting broad topics
 - Getting blueprints created
- UDS
 - Recording discussions and decisions
- Post UDS
 - Ensuring specifications are updated
 - Defining workitems and other detail
- Release Reporting
 - <https://wiki.ubuntu.com/KernelTeam/ReleaseStatus/Lucid>

Review

- Reviewing Ubuntu configuration
 - <https://blueprints.launchpad.net/ubuntu/+spec/kernel-lucid-kernel-config-review>
 - What new options have been set?
 - Do the existing selections make sense?
- Reviewing Ubuntu patch delta and drivers
 - <https://blueprints.launchpad.net/ubuntu/+spec/kernel-lucid-review-of-ubuntu-delta>
 - Why is this patch still needed?
 - Should we update or drop this driver?
- Selecting our base Kernel
 - Which kernel will land in time?

Keeping in Sync

- Ongoing updates to the each mainline release
 - Pulling the delta up to each -rcN and mainline release up to the chose release
- Updates based on UDS discussions
 - Selection of upstream features such as filesystem
 - enabling/disabling various options as needed
- Pulling in new in-house development
 - As a service team we are there to provide development support for other teams
 - New features are completed those must be pulled in and uploaded for testing

Stability

- Once we reach the **chosen** kernel version we start to follow the stable branch
 - This typically only occurs just before kernel freeze for non-LTS, this cycle much earlier
 - Management starts to mirror **released** kernel process
 - Generally we simply cherry-pick stable update patches into the branch

Freezing

- Milestone Freezes
 - On the Tuesday of a milestone week the archive is frozen
 - The kernel for the milestone must be **built** and in the archive by then
- Kernel Freeze
 - kernel freezes is about a month before general release freeze
 - Allows time for consumers to consolidate and test

Wrap Up

- Generally we try and upload a new kernel weekly
 - In step with -rc release cycle
 - Maximises bisectability
- Kernels are generated from the git repository
 - <http://kernel.ubuntu.com/git> ubuntu/ubuntu-<series>
 - dpkg-buildpackage -S
 - Make sure it is nice and clean
- Announcements
 - As with all uploads announcements are sent to the **stake holders**