

Daily Build Scripts

Hans Verkuil

Cisco Systems Norway

Daily Build Scripts

- git://linuextv.org/hverkuil/build-scripts.git
- These scripts run the daily build compilation and regression tests.
- It builds the `media_stage` master branch for several architectures: `sh`, `arm-at91`, `arm-davinci`, `arm-stm32`, `arm-pxa`, `arm-multi`, `mips`, `powerpc64`, `i686` and `x86_64`.
- It cross compiles `binutils/gcc` so it is easy to upgrade to the latest `binutils/gcc`.
- It checks if all drivers can be compiled with `COMPILE_TEST`.
- It checks if `strcpy`, `strncpy` or `strncpy` are used (`strscopy` should be used instead).
- It builds the various applications (`v4l-utils`, etc.)
- It builds the html documentation (Missing: no pdf at the moment).
- It builds with `CONFIG_PM=n`, `CONFIG_PM_SLEEP=n`, `CONFIG_OF=n` and `CONFIG_DEBUG_FS=n`.
- It builds with `sparse` and `smatch`: warnings from non-media headers are filtered (Missing: needs updating).
- It builds the `kerneldocs`.

Daily Build Scripts

- It checks for unexpected ABI changes. Missing: it doesn't run pahole tests for the codec controls (I have a local test for that, it can be integrated into the build system).
- Finally, it runs regression tests using contrib/test/test-media from v4l-utils, both using 64 bit utilities and with a 32 bit version of the utilities to test the compat32 code.
- These regression tests use the various virtual drivers and a kernel that has lots of debugging enabled.
- I run this build as well when testing Pull Requests using some simple helper scripts where I can just give the branch name to test.
- Missing: it does not test patch-by-patch from a patch series, I have a separate script for that, but it really should be integrated in build.sh.
- Missing: the regression tests rely on a VM, and currently I'm the only one that can run this.

Daily Build Scripts

- It needs more cleanup: probably some of the architecture variants do not need to be compiled anymore (due to improvements in `COMPILE_TEST` support).
- Code cleanup: too much code duplication, can be done a lot nicer.
- Easy to run locally. It does the same tests as the daily build, so no surprises there.
- Regression tests are very useful when changing core code and/or uAPIs.