Media Subsystem Profile

Overview

The media subsystem covers support for a variety of devices: stream capture, analog and digital TV streams, cameras, remote controllers, HDMI CEC and media pipeline control.

It covers, mainly, the contents of those directories:

- drivers/media
- drivers/staging/media
- Documentation/admin-guide/media
- Documentation/driver-api/media
- Documentation/userspace-api/media
- Documentation/devicetree/bindings/media/[1]
- include/media

[1] Device tree bindings are maintained by the OPEN FIRMWARE AND FLATTENED DEVICE TREE BINDINGS maintainers (see the MAINTAINERS file). So, changes there must be reviewed by them before being merged via the media subsystem's development tree.

Both media userspace and Kernel APIs are documented and the documentation must be kept in sync with the API changes. It means that all patches that add new features to the subsystem must also bring changes to the corresponding API files.

Due to the size and wide scope of the media subsystem, media's maintainership model is to have sub-maintainers that have a broad knowledge of a specific aspect of the subsystem. It is the sub-maintainers' task to review the patches, providing feedback to users if the patches are following the subsystem rules and are properly using the media kernel and userspace APIs.

Patches for the media subsystem must be sent to the media mailing list at linux-media@vger.kernel.org as plain text only e-mail. Emails with HTML will be automatically rejected by the mail server. It could be wise to also copy the sub-maintainer(s).

Media's workflow is heavily based on Patchwork, meaning that, once a patch is submitted, the e-mail will first be accepted by the mailing list server, and, after a while, it should appear at:

- https://patchwork.linuxtv.org/project/linux-media/list/

If it doesn't automatically appear there after a few minutes, then probably something went wrong on your submission. Please check if the email is in plain text[2] only and if your emailer is not mangling whitespaces before complaining or submitting them again.

You can check if the mailing list server accepted your patch, by looking at:

- https://lore.kernel.org/linux-media/

[2] If your email contains HTML, the mailing list server will simply drop it, without any further notice.

Media maintainers

At the media subsystem, we have a group of senior developers that are responsible for doing the code reviews at the drivers (also known as sub-maintainers), and another senior developer responsible for the subsystem as a whole. For core changes, whenever possible, multiple media maintainers do the review.

The media maintainers that work on specific areas of the subsystem are:

- Remote Controllers (infrared):
  Sean Young <sean@mess.org>
- HDMI CEC:
  Hans Verkuil <hverkuil@xs4all.nl>
- Media controller drivers:
  Laurent Pinchart <laurent.pinchart@ideasonboard.com>
- ISP, v4l2-async, v4l2-fwnode, v4l2-flash-led-class and Sensor drivers:
  Sakari Ailus <sakari.ailus@linux.intel.com>
- V4L2 drivers and core V4L2 frameworks:
  Hans Verkuil <hverkuil@xs4all.nl>

The subsystem maintainer is:
Mauro Carvalho Chehab <mchehab@kernel.org>

Media maintainers may delegate a patch to other media maintainers as needed. On such case, checkpatch's `delegate` field indicates who's currently responsible for reviewing a patch.

Submit Checklist Addendum

Patches that change the Open Firmware/Device Tree bindings must be reviewed by the Device Tree maintainers. So, DT maintainers should be Cc:ed when those are submitted via devicetree@vger.kernel.org mailing list.

There is a set of compliance tools at https://git.linuxtv.org/v4l-utils.git/ that should be used in order to check if the drivers are properly implementing the media APIs:

Type	Tool
V4L2 drivers[3]	`v4l2-compliance`
V4L2 virtual drivers	`contrib/test/test-media`
CEC drivers	`cec-compliance`

[3] The `v4l2-compliance` also covers the media controller usage inside V4L2 drivers.

Other compilance tools are under development to check other parts of the subsystem.

Those tests need to pass before the patches go upstream.

Also, please notice that we build the Kernel with:

```
make CF=-D__CHECK_ENDIAN__ CONFIG_DEBUG_SECTION_MISMATCH=y C=1 W=1 CHECK=check_script
```

Where the check script is:

```
#!/bin/bash
/devel/smatch/smatch -p=kernel $@ >&2
/devel/sparse/sparse $@ >&2
```

Be sure to not introduce new warnings on your patches without a very good reason.

Style Cleanup Patches

Style cleanups are welcome when they come together with other changes at the files where the style changes will affect.

We may accept pure standalone style cleanups, but they should ideally be one patch for the whole subsystem (if the cleanup is low volume), or at least be grouped per directory. So, for example, if you're doing a big cleanup change set at drivers under drivers/media, please send a single patch for all drivers under drivers/media/pci, another one for drivers/media/usb and so on.

Coding Style Addendum

Media development uses `checkpatch.pl` on strict mode to verify the code style, e.g.:

```
$ ./scripts/checkpatch.pl --strict --max-line-length=80
```

In principle, patches should follow the coding style rules, but exceptions are allowed if there are good reasons. On such case, maintainers and reviewers may question about the rationale for not addressing the `checkpatch.pl`.

Please notice that the goal here is to improve code readability. On a few cases, `checkpatch.pl` may actually point to something that would look worse. So, you should use good sense.

Note that addressing one `checkpatch.pl` issue (of any kind) alone may lead to having longer lines than 80 characters per line. While this is not strictly prohibited, efforts should be made towards staying within 80 characters per line. This could include using re-factoring code that leads to less indentation, shorter variable or function names and last but not least, simply wrapping the lines.

In particular, we accept lines with more than 80 columns:

- on strings, as they shouldn't be broken due to line length limits;
- when a function or variable name need to have a big identifier name, which keeps hard to honor the 80 columns limit;
- on arithmetic expressions, when breaking lines makes them harder to read;
- when they avoid a line to end with an open parenthesis or an open bracket.

Key Cycle Dates

New submissions can be sent at any time, but if they intend to hit the next merge window they should be sent before -rc5, and ideally stabilized in the linux-media branch by -rc6.

Review Cadence

Provided that your patch is at https://patchwork.linuxtv.org, it should be sooner or later handled, so you don't need to re-submit a patch.

Except for bug fixes, we don't usually add new patches to the development tree between -rc6 and the next -rc1.

Please notice that the media subsystem is a high traffic one, so it could take a while for us to be able to review your patches. Feel free

to ping if you don't get a feedback in a couple of weeks or to ask other developers to publicly add Reviewed-by and, more importantly, `Tested-by:` tags.

Please note that we expect a detailed description for `Tested-by:`, identifying what boards were used at the test and what it was tested.