Contributing to Jailhouse

=========================

Contributions to Jailhouse are always welcome. This document explains the

general requirements on contributions and the recommended preparation steps. It

also sketches the typical integration process of patches.

Contribution Checklist

----------------------

- use git to manage your changes [\*recommended\*]

- follow Documentation/coding-style.txt coding style [\*\*required\*\*]

- for python code run pep8 coding style checker [\*\*required\*\*]

- add the required copyright header to each new file introduced, see

[licensing information](LICENSING.md) [\*\*required\*\*]

- structure patches logically, in small steps [\*\*required\*\*]

- one separable functionality/fix/refactoring = one patch

- do not mix those three into a single patch (e.g. first refactor, then

add a new functionality that builds onto the refactoring)

- after each patch, the tree still has to build and work, i.e. do not add

even temporary breakages inside a patch series (helps when tracking down

bugs)

- use `git rebase -i` to restructure a patch series

- base patches on top of latest master or - if there are dependencies - on next

(note: next is an integration branch that may change non-linearly)

- test patches sufficiently (obvious, but...) [\*\*required\*\*]

- no regressions are caused in affected code

- seemingly unaffected architectures still build (use Travis CI e.g.)

- static code analyzer finds no new defects (register a github fork with

Travis CI and Coverity for free scanning) [\*recommended\*]

- python code shall be tested with python 2 and 3 [\*\*required\*\*]

- the world is still spinning

- add signed-off to all patches [\*\*required\*\*]

- to certify the "Developer's Certificate of Origin", see below

- check with your employer when not working on your own!

- add Fixes: to all bug-fix commits [\*recommended\*]

- the Fixes: tag format shall be:

Fixes: 12-byte-hash ("subject of bug-introducing commit")

- if you are unsure of the bug-introducing commit do \*not\* add a

Fixes: tag - no Fixes: tag is better than a wrong Fixes: tag.

- post patches to mailing list [\*\*required\*\*]

- use `git format-patch/send-email` if possible

- send patches inline, do not append them

- no HTML emails!

- CC people who you think should look at the patches, e.g.

- affected maintainers (see areas of responsibility below)

- someone who wrote a change that is fixed or reverted by you now

- who commented on related changes in the recent past

- who otherwise has expertise and is interested in the topic

- pull requests on github are only optional

- post follow-up version(s) if feedback requires this

- send reminder if nothing happened after about a week

Developer's Certificate of Origin 1.1

-------------------------------------

When signing-off a patch for this project like this

Signed-off-by: Random J Developer <random@developer.example.org>

using your real name (no pseudonyms or anonymous contributions), you declare the

following:

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I

have the right to submit it under the open source license

indicated in the file; or

(b) The contribution is based upon previous work that, to the best

of my knowledge, is covered under an appropriate open source

license and I have the right under that license to submit that

work with modifications, whether created in whole or in part

by me, under the same open source license (unless I am

permitted to submit under a different license), as indicated

in the file; or

(c) The contribution was provided directly to me by some other

person who certified (a), (b) or (c) and I have not modified

it.

(d) I understand and agree that this project and the contribution

are public and that a record of the contribution (including all

personal information I submit with it, including my sign-off) is

maintained indefinitely and may be redistributed consistent with

this project or the open source license(s) involved.

See also https://www.kernel.org/doc/Documentation/process/submitting-patches.rst

(Section 11, "Sign your work") for further background on this process which was

adopted from the Linux kernel.

Contribution Integration Process

--------------------------------

1. patch reviews performed on mailing list

\* at least by maintainers, but everyone is invited

\* feedback has to consider design, functionality and style

\* simpler and clearer code preferred, even if original code works fine

2. accepted patches merged into next branch

3. further testing done by community, including CI build tests and code

analyzer runs

4. if no new problems or discussions showed up, acceptance into master

\* grace period for master: about 3 days

\* urgent fixes may be applied sooner

github facilities are not used for the review process so that people can follow

all changes and related discussions at a single stop, the mailing list. This

may change in the future if github should improve their email integration.

Areas of responsibility

-----------------------

Jailhouse is rather small. Nevertheless, there are different people involved in

different areas of its code. The following list shall give an overview on who

is working in which area and should be involved when discussing changes:

Jan Kiszka <jan.kiszka@siemens.com>:

- overall Jailhouse maintenance

- committer to official repository

Valentine Sinitsyn <valentine.sinitsyn@gmail.com>:

- AMD64 support

Henning Schild <henning.schild@siemens.com>:

- inter-cell communication

- configuration file generator

Ralf Ramsauer <ralf.ramsauer@oth-regensburg.de>

- uart infrastructure

- inmate library