

Contribute to Free/Libre and OpenSource Software

HowTo Get Your Things Upstream

Alexander Dahl

<http://www.lespocky.de/>

2017-03-11

What?

Why?

How?

Communicate

Git

Me

Yet another free software developer ...

Background

- ▶ using Free Software since \approx 2001
- ▶ contributing to Free Software since \approx 2003
- ▶ diploma in engineering (mechatronics)
- ▶ working as Embedded Linux developer

Me

Yet another free software developer ...

Background

- ▶ using Free Software since \approx 2001
- ▶ contributing to Free Software since \approx 2003
- ▶ diploma in engineering (mechatronics)
- ▶ working as Embedded Linux developer

Projects

- ▶ fli4l
- ▶ buildroot
- ▶ ptxdist
- ▶ libcgi
- ▶ Freifunk

What?

What?

Why?

How?

Communicate

Git

Free/Libre and OpenSource Software

What Do We Want?

- ▶ Free beer!!1!

Free/Libre and OpenSource Software

What Do We Want?

- ▶ ~~Free beer!!1!~~
- ▶ Free speech!

Free/Libre and OpenSource Software

What Do We Want?

- ▶ Free beer!!1!
- ▶ Free speech!

Freedoms as Defined by Free Software Foundation (FSF)

Freedom 0 The freedom to run the program for any purpose.

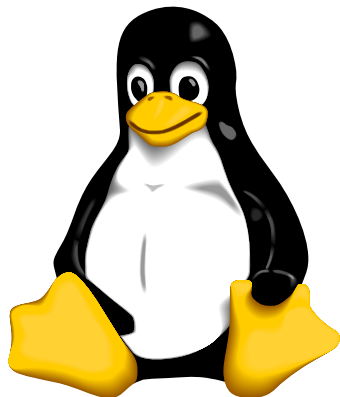
Freedom 1 The freedom to study how the program works, and change it to make it do what you wish.

Freedom 2 The freedom to redistribute and make copies so you can help your neighbor.

Freedom 3 The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

Components of FLOSS

- ▶ Sourcecode
- ▶ Documentation
- ▶ Artwork
- ▶ Community
- ▶ Tests
- ▶ Infrastructure



Why?

What?

Why?

How?

Communicate

Git

Why Contribute?

- ▶ add features
- ▶ fix bugs
- ▶ improve software
- ▶ for fun
- ▶ social responsibility
- ▶ sustainability (use old devices)
- ▶ learning
- ▶ business
- ▶ ...

Why Upstreaming?

- ▶ more people can use it
- ▶ review and improvement of your stuff
- ▶ less future work
- ▶ avoid fragmentation
- ▶ give back to community
- ▶ ...

How?

What?





Why?

How?

Communicate

Git

I Want to Help!

Betreff Re: json-c 0.12.1 release
Von José Bollo <jose.bollo@iot.bzh> 
Absender json-c@googlegroups.com 
An json-c <json-c@googlegroups.com> 
Antwort an json-c@googlegroups.com 
Datum 2016-07-27 14:56

Le mardi 7 juin 2016 06:41:32 UTC+2, Eric a écrit :

Announcing the 0.12.1 release of json-c.

The only changes as compared to 0.12 are to include the couple of minor compile fixes that were already present on the json-c-0.12 branch, and this release is largely just flushing out a couple of tarballs that should have been created long ago.

Note that at some point in the (hopefully) near future, I expect to create an arguably more useful 0.13 release off of the master branch. Volunteers are always welcome to help.

I'm volunteer for improving a lot json-c. I have full ideas and good knowledge.

Eric

--

You received this message because you are subscribed to the Google Groups "json-c" group.
To unsubscribe from this group and stop receiving emails from it, send an email to json-c+unsubscribe@googlegroups.com.
For more options, visit <https://groups.google.com/d/optout>.

I Want to Help!

- > Note that at some point in the (hopefully) near
- > future, I expect to create an arguably more
- > useful 0.13 release off of the master branch.
- > Volunteers are always welcome to help.

I'm volunteer for improving a lot json-c. I have full ideas and good knowledge.

It Depends ...

- ▶ what type of contribution
 - ▶ Sourcecode
 - ▶ Documentation
 - ▶ Artwork
 - ▶ Community
 - ▶ Tests
 - ▶ (Infrastructure)
- ▶ which project/community
- ▶ your preference

Ask and Answer Questions

The community part

- ▶ mailing list
- ▶ forum
- ▶ chat
- ▶ stackoverflow
- ▶ social media
- ▶ usenet
- ▶ ...

Talk, Talk, Talk

It's a trap ...

- ▶ make yourself familiar with community rules
 - ▶ Code of Conduct
 - ▶ non written rules
- ▶ choose the right channel
- ▶ be polite
- ▶ be patient
- ▶ [learn.to/quote](#)

Use the Bugtracker

Which

- ▶ Bugzilla
- ▶ Trac
- ▶ Mantis
- ▶ Redmine
- ▶ GitHub
- ▶ Atlassian Jira

Use the Bugtracker

Which

- ▶ Bugzilla
- ▶ Trac
- ▶ Mantis
- ▶ Redmine
- ▶ GitHub
- ▶ Atlassian Jira

How

- ▶ make sure it's a bug (not a »layer 8 problem«)
- ▶ detailed description
- ▶ reproducible (minimal example)
- ▶ (add a patch)

What's a Patch?

```
From: Alexander Dahl <alex@netz39.de>
Date: Thu, 17 Mar 2016 15:04:09 +0100
Subject: x86: fix platform_export_bootpart() for Xen virtual disks
```

Virtual disk devices in a Xen virtual machine (DomU) can be /dev/xvda, /dev/xvdb and so on with partitions like /dev/xvda1. Devices named like this were not considered before. This resulted in a non working sysupgrade, because the boot partition could not be found.

```
Signed-off-by: Alexander Dahl <alex@netz39.de>
Suggested-by: Matthias Schiffer <mschiffer@universe-factory.net>
```

```
diff --git a/target/linux/x86/base-files/lib/upgrade/platform.sh b/target/linux/x86/base-files/lib/upgrade/platform.sh
index 73ab5ef..adc119c 100644
--- a/target/linux/x86/base-files/lib/upgrade/platform.sh
+++ b/target/linux/x86/base-files/lib/upgrade/platform.sh
@@ -17,7 +17,7 @@ platform_export_bootpart() {
     PARTUUID=[a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9]
     uuid="${disk#PARTUUID=}"
     uuid="${uuid%-02}"
-   for disk in /dev/[hsv]d[a-z]; do
+   for disk in /dev/[hsv]d[a-z] /dev/xvd[a-z]; do
     set -- $(dd if=$disk bs=1 skip=440 count=4 2>/dev/null | hexdump -n 4)
     if [ "$4$3$2$1" = "$uuid" ]; then
         export BOOTPART="${disk}1"
```

What's a Patch?

```
From: Alexander Dahl <alex@netz39.de>
Date: Thu, 17 Mar 2016 15:04:09 +0100
Subject: x86: fix platform_export_bootpart() for Xen virtual disks
```

Virtual disk devices in a Xen virtual machine (DomU) can be /dev/xvda, /dev/xvdb and so on with partitions like /dev/xvda1. Devices named like this were not considered before. This resulted in a non working sysupgrade, because the boot partition could not be found.

```
Signed-off-by: Alexander Dahl <alex@netz39.de>
Suggested-by: Matthias Schiffer <mschiffer@universe-factory.net>
```

```
diff --git a/target/linux/x86/base-files/lib/upgrade/platform.sh b/target/linux/x86/base-files/lib/upgrade/platform.sh
index 73ab5ef..adc119c 100644
--- a/target/linux/x86/base-files/lib/upgrade/platform.sh
+++ b/target/linux/x86/base-files/lib/upgrade/platform.sh
@@ -17,7 +17,7 @@ platform_export_bootpart() {
     PARTUUID=[a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9][a-f0-9]
     uuid="${disk#PARTUUID=}"
     uuid="${uuid%-02}"
-    for disk in /dev/[hsv]d[a-z]; do
+    for disk in /dev/[hsv]d[a-z] /dev/xvd[a-z]; do
         set -- $(dd if=$disk bs=1 skip=440 count=4 2>/dev/null | hexdump -n 4)
         if [ "$4$3$2$1" = "$uuid" ]; then
             export BOOTPART="${disk}1"
```

This fixed sysupgrade for OpenWRT Xen VM. Merged both in gluon (Freifunk) and upstream. See [PR-692](#) on GitHub.

Why Git?

Why?

- ▶ in 2016 most projects start with Git
- ▶ GitHub
- ▶ lots of projects converted their old VCS to Git
- ▶ distributed (offline) work possible

Why Git?

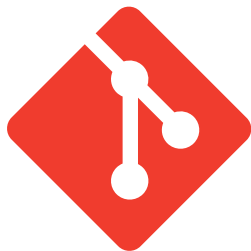
Why?

- ▶ in 2016 most projects start with Git
- ▶ GitHub
- ▶ lots of projects converted their old VCS to Git
- ▶ distributed (offline) work possible

Why not?

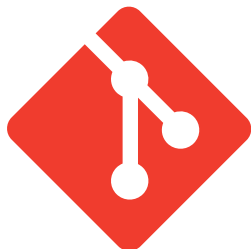
- ▶ the bad things: <https://stevebennett.me/2012/02/24/10-things-i-hate-about-git/>

Learn Git



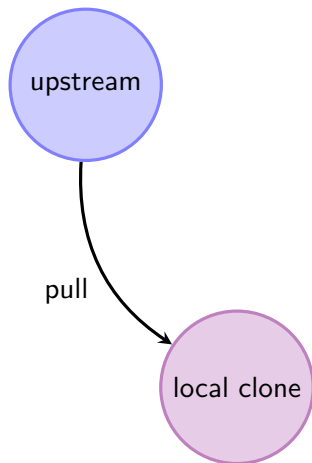
- ▶ Books
 - ▶ e.g. [Pro Git](#)
- ▶ Tutorials
- ▶ Workshops
- ▶ see <https://git-scm.com/doc>

Learn Git



- ▶ Books
 - ▶ e.g. **Pro Git**
- ▶ Tutorials
- ▶ Workshops
- ▶ see <https://git-scm.com/doc>
- ▶ **Use it!**

It's Distributed!



- ▶ add
- ▶ branch
- ▶ checkout
- ▶ commit
- ▶ diff
- ▶ log
- ▶ status

Send Patches

Yes, to mailing lists ...

Do Send

- ▶ clone upstream repository
- ▶ create a local branch
- ▶ work, work ...
- ▶ `git format-patch`
- ▶ `git send-email`

Send Patches

Yes, to mailing lists ...

Do Send

- ▶ clone upstream repository
- ▶ create a local branch
- ▶ work, work ...
- ▶ `git format-patch`
- ▶ `git send-email`

Let Pull

- ▶ host your own git server
- ▶ synchronize with upstream **and** your git server
(`git help remote`)
- ▶ write a mail saying someone should pull from your server

Pull Requests

Get some changes from my tree!

- ▶ `git request-pull`
- ▶ click this GitHub button

Pull Requests

Get some changes from my tree!

- ▶ `git request-pull`
- ▶ click this GitHub button

What does it mean?

I made some changes and pushed it to my public repository. Would you mind pulling from there to review what I did and maybe integrate it upstream?

Pull Requests

Get some changes from my tree!

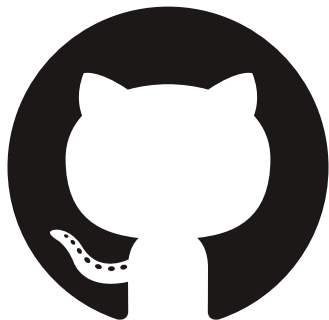
- ▶ `git request-pull`
- ▶ click this GitHub button

What does it mean?

I made some changes and pushed it to my public repository. Would you mind pulling from there to review what I did and maybe integrate it upstream?

- ▶ upstream maintainer should review patch (series)
- ▶ tools like [patchworks](#) help maintainer
- ▶ <http://www.marmalade.de/magazin/2014/12/pullrequests-fluch-und-segen/>

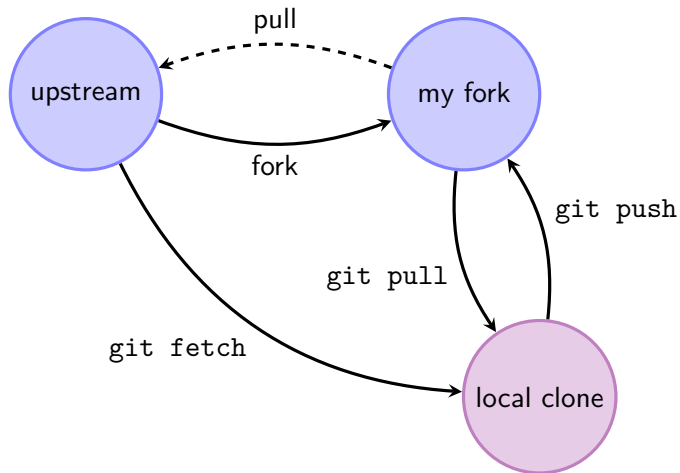
GitHub®



- ▶ Git repository hosting
- ▶ issue tracker
- ▶ wiki
- ▶ easy fork, branch, pull request, merge ...
- ▶ social net (follower)
- ▶ statistics
- ▶ service integration
- ▶ website hosting
- ▶ (enterprise services)

GitHub Workflow

Looks easier at first sight ...



Patch Improvement

Better Patches

- ▶ change one thing at a time
- ▶ write a good commit message
 - ▶ short first line
 - ▶ one empty line
 - ▶ describe the reason for the change
 - ▶ reference ticket numbers

Patch Improvement

Better Patches

- ▶ change one thing at a time
- ▶ write a good commit message
 - ▶ short first line
 - ▶ one empty line
 - ▶ describe the reason for the change
 - ▶ reference ticket numbers


Patch v2

- ▶ `git format-patch -v4 -M -s -o outgoing origin/master`
- ▶ what can `git rebase` do for you?
- ▶ see [buildroot manual](#) on submitting patches

Contribute Without Git

- ▶ sourcecode and documentation
 - ▶ patch and quilt
 - ▶ other VCS
 - ▶ things not in VCS
 - ▶ ...
- ▶ other categories
 - ▶ you probably won't need a VCS

What Happened Next on the JSON-C Mailing List?

Betreff Re: json-c 0.12.1 release
Von Eric Haszlkiewicz <hawicz@gmail.com> 
Absender json-c@googlegroups.com 
An json-c@googlegroups.com 
Antwort an json-c@googlegroups.com 
Datum 2016-07-27 16:01

On Jul 27, 2016 8:57 AM, "José Bollo" <jose.bollo@iot.bzh> wrote:

I'm volunteer for improving a lot json-c. I have full ideas and good knowledge.

Great! If you have some changes you're thinking of making to json-c feel free to post a message with your idea (w/ a separate subject please). If you have code changes, pull requests are welcome, especially if they include tests and any necessary doc changes.

Eric

--





You received this message because you are subscribed to the Google Groups "json-c" group.
To unsubscribe from this group and stop receiving emails from it, send an email to json-c+unsubscribe@googlegroups.com.
For more options, visit <https://groups.google.com/d/optout>.

What Happened Next on the JSON-C Mailing List?

- > I'm volunteer for improving a lot json-c. I have
- > full ideas and good knowledge.

Great! If you have some changes you're thinking of making to json-c feel free to post a message with your idea (w/ a separate subject please). If you have code changes, pull requests are welcome, especially if they include tests and any necessary doc changes.

What Happened Next on the JSON-C Mailing List?

Betreff going forward
Von José Bollo <jose.bollo@iot.bzh> 
Absender json-c@googlegroups.com 
An json-c <json-c@googlegroups.com> 
Antwort an json-c@googlegroups.com 
Datum 2016-07-27 18:12

Hi all,

It is a pain for me and for you to read/write long mail thus here are many of my proposals. I already implemented most of it.

- specialize internal structures (partly done)
- avoid to serialize when previous result exist (kind of already done)
- use `size_t` where needed (done)
- serialize to/from `bjson` (half done)
- create a new type "custom" for handling serialisation of both json and `bjson`
- make legacy the model of custom serialisation that has serious drawbacks
- better report errors of serialisation (done)
- reduce the cost in memory (done)
- avoid use of function pointer for serialising (done)
- return status when adding items to dictionaries
- make boolean and empty strings uniques (done)
- make possible to handle a dictionary entries to change it quickly (or iterate over their)
- allow any value as dictionary key

work done for people that have time: <https://gitlab.com/jobol/json-c>

I'm working for embedded systems and would really like to reduce the memory footprint and the performance of `json-c`

I think that it is possible and that the API (that is good) can be preserved.

Feel free to comment and send feedback here

Best regards
Jose Bollo

Legal Stuff

The Linux Mascot

Penguin Tux by [Larry Ewing](#) and [The GIMP](#), vectorized by [Simon Budig](#), converted to TikZ by [Stefan Kottwitz](#).

The Git Logo

Git Logo by [Jason Long](#) is licensed under the [Creative Commons Attribution 3.0 Unported License](#).

The GitHub Logo

GITHUB®, the GITHUB® logo design, OCTOCAT® and the OCTOCAT® logo design are exclusive trademarks registered in the United States by GitHub, Inc.

The OCTOCAT design is the exclusive property of GitHub, Inc and has been federally registered with the United States Copyright Office. All rights reserved.

The Last Slide

Contact Me

E-Mail post@lespocky.de

WWW lespocky.de or blog.antiblau.de

Twitter [@LeSpocky](https://twitter.com/LeSpocky)

Slides

- ▶ `hg clone https://bitbucket.org/lespocky/talks`

License

These slides are licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. (CC BY-SA 4.0)

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-sa/4.0/>.