

From GitHub Projects to the Linux Kernel: How FOSS Contributions Work

Nandakumar Edamana

2025-11-06

Why Contribute

- Everybody gets to use better software (including you)
But why don't keep the change private?
- Meet brilliant people, code, and ideas; learn a lot
Thanks to the source, hacking required, and the public review process
- Make your profile attractive
Doesn't hurt to put a Linux Kernel commit ID in your CV

Be Prepared

Don't focus too much on contributing to others' projects; have personal FOSS projects as well.

Why?

It's their project; you are just a contributor.

- Disagreements, owners changing the license of the project, burnout, etc.
- Less opportunity to gain end-to-end expertise
(Personal projects: programming to branding and packaging)

Just a reminder! Don't back off altogether!

What Can I Contribute?

Does your favourite game need more logic or new **assets**?

What Can I Contribute?

Does your favourite game need more logic or new **assets**?

- The Doom engine was open sourced, but the assets weren't (now there is Freedoom)

What Can I Contribute?

Does your favourite game need more logic or new **assets**?

- The Doom engine was open sourced, but the assets weren't (now there is Freedoom)
- OpenRW - Free re-implementation of the GTA engine; still no assets



Do I need GTA III to play OpenRW?

In short: Yes.

OpenRW is simply a game engine that is compatible with the original game, **it has no assets of its own**. In order to play GTA III using OpenRW you must own the game and have the data installed on your system. If you need a copy of the game it's available on [Steam](#).

It is theoretically possible to develop a completely new game, without any of the assets or content from GTA III, however no such project is currently under way.

What Can I Contribute?

- People say GIMP's internals are powerful enough, but the UI/UX is lacking

What Can I Contribute?

- People say GIMP's internals are powerful enough, but the UI/UX is lacking
- Machine learning. . . alright; more data! Collect and cleanup.

What Can I Contribute?

- People say GIMP's internals are powerful enough, but the UI/UX is lacking
- Machine learning. . . alright; more data! Collect and cleanup.
- Encyclopediae: verification is more important than a rainfall of words

What Can I Contribute?

- People say GIMP's internals are powerful enough, but the UI/UX is lacking
- Machine learning. . . alright; more data! Collect and cleanup.
- Encyclopediae: verification is more important than a rainfall of words
- We miss amazing FOSS programs because they are not packaged for our distro, Flatpak, etc.

Testing: An Underrated Contribution



- rack2009.jpg from openbsd.org
- i386, amd64, VAX, MIPS, SPARC, etc.
- Contains donated machines?

Testing: An Underrated Contribution

- Real world: diverse hardware, so many distros and environments, added diversity with locale, themes, etc.
- Not just infeasible, but impossible
- Solution: a diverse and helpful community
- Everybody learns

Testing: An Underrated Contribution

Remember?

- The xz backdoor incident (2024)
- Hate speech found in Ubuntu 23.10 UI translation

UI Translation

- Translate or review strings one by one
- Many projects use platforms like weblate.org and transifex.com, which are Web-based, interactive, and collaborative

"Welcome!\n" → "സ്വാഗതം!\n"

"Sent %d file(s).\n" → "%d ഫയലുകൾ അയച്ചു.\n"

Surprising Points of Entry

Sure, there are non-code contributions, but what if I wanted to contribute code?

Surprising Points of Entry

Sure, there are non-code contributions, but what if I wanted to contribute code?

- The Linux Kernel tree has userspace components (e.g.: libbpf)
- Contribute to supporting components – tests, build scripts, websites, etc.

Finding Opportunities

- Find good first issues and “help wanted” issues on GitHub
is:issue label:"good first issue"
- Help Wanted/Contribute pages on project websites
- Find bugs by fuzzing or other methods

Where Do I Join?

Not just GitHub.com or GitLab.com.



Welcome to
Savannah, the
software forge for
people committed to
free software:

- Some host their on GitLab instances (e.g.: gitlab.gnome.org)
- Linux Kernel: patches and pull requests via email

Do: Locate the Upstream

- Visit the official website, repo, bug tracker, discussion forums, and the mailing list archives
- Example for bug tracker: bugzilla.mozilla.org
- Read the documentation and understand the conventions

Don't: Submit Against a Mirror

- github.com/torvalds/linux is a mirror; not the upstream
- Still thousands of PRs

KernelPRBot commented on Sep 22

Hi @username!

Thanks for your contribution to the Linux kernel!

Linux kernel development happens on mailing lists, rather than on GitHub - this GitHub repository is a read-only mirror that isn't used for accepting contributions.

...

The Workflow

We all know the GitHub/GitLab workflow:

- ① Spot the project
- ② File an issue/pick an issue, get it assigned (optional)
- ③ Fork the project (one-time)
- ④ Make your changes in the fork, in a new branch
- ⑤ Submit a pull request (merge request) to the upstream

The Linux Way

Trees:

- Each subsystem has its own tree (basically a fork)
maintainers can have their own trees
- Hosted at git.kernel.org
- e.g.: bpf, bpf-next, bluetooth-next, etc.
- linux-next
 - Integration of patches from different subsystems
 - Patches to be considered in the next merge window

The Linux Way

Send your patch to the maintainers of the corresponding tree.

What's a Patch? Basically a diff (with more metadata than shown).

```
diff --git a/tools/lib/bpf/libbpf.c b/tools/lib/bpf/libbpf.c
index 194809da5172..1cc87dbd015d 100644
--- a/tools/lib/bpf/libbpf.c
+++ b/tools/lib/bpf/libbpf.c
@@ -2106,7 +2106,7 @@ static int set_kcfg_value_str(struct extern_desc *ext, char *ext_val,
     }

     len = strlen(value);
-    if (value[len - 1] != '\0') {
+    if (len < 2 || value[len - 1] != '\0') {
         pr_warn("extern (kcfg) '%s': invalid string config '%s'\n",
                 ext->name, value);
         return -EINVAL;
```

The Linux Way

Send your patch to the maintainers of the corresponding tree.

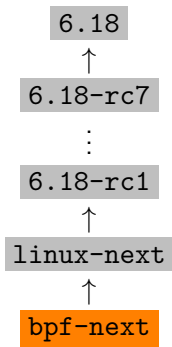
What's a Patch? Basically a diff (with more metadata than shown).

```
diff --git a/tools/lib/bpf/libbpf.c b/tools/lib/bpf/libbpf.c
index 194809da5172..1cc87dbd015d 100644
--- a/tools/lib/bpf/libbpf.c
+++ b/tools/lib/bpf/libbpf.c
@@ -2106,7 +2106,7 @@ static int set_kcfg_value_str(struct extern_desc *ext, char *ext_val,
     }

     len = strlen(value);
-    if (value[len - 1] != '\0') {
+    if (len < 2 || value[len - 1] != '\0') {
         pr_warn("extern (kcfg) '%s': invalid string config '%s'\n",
                 ext->name, value);
         return -EINVAL;
```

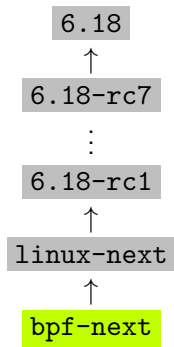
There's `git format-patch` and `git send-email` to automate creating and sending patches.

Journey of an eBPF Patch



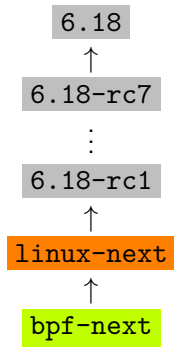
- Send your patch to the BPF maintainers
Cc: `bpf@vger.kernel.org`
- Do self-review based on errors reported by `patchwork.kernel.org`
- Wait for reviews by maintainers of the subtree
- Send revisions

Journey of an eBPF Patch



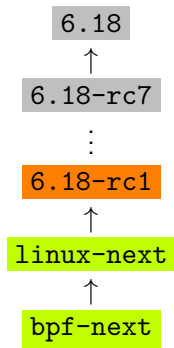
- Your patch is accepted to the subtree
- Your work is done unless something goes wrong in the following stages

Journey of an eBPF Patch



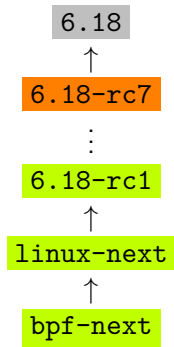
Patch enters `linux-next` as part of integration testing.

Journey of an eBPF Patch



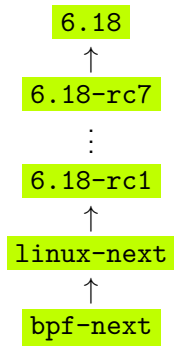
Merge window opens and the patch enters a *release candidate*.

Journey of an eBPF Patch



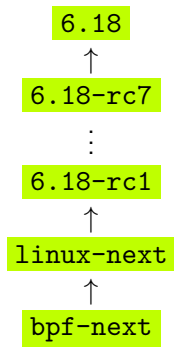
7th week; the patch is still there.

Journey of an eBPF Patch



It's official.

Journey of an eBPF Patch



It's official.

But it may not be the end yet. An important fix could get *backported* to earlier versions.

Patch Acceptance Cheatsheet

No rules of thumb; some observations based on personal experience:

Bad	Good
Big	Small
Overhaul	Isolated
Subjective benefits	Objectively impactful

My patch:

- Improved the general precision of an operation while simplifying the algorithm
- Backed by benchmarks and formal proof

Don't: Submit Cheap PRs

- Fixing typos are important
- A dozen PRs each fixing a single typo in the same file? Bad.
- In other words, avoid submitting PRs just to increase your karma.

Don't: Submit AI Slop

- Autogenerated junk code and bug reports that are false positives
- Maintainers are already burnt out



Ars Technica

<https://arstechnica.com> › gadgets › 2025 › 05 › open-source-project-curl-is-sick-of-users-submitt... ...

Open source project curl is sick of users submitting "AI slop ...

May 7, 2025 · Open source project curl is sick of users submitting "AI slop" vulnerabilities "One way you can tell is it's always such a nice report," founder tells Ars.



It's FOSS News

<https://news.itsfoss.com> › curl-ai-slop ...

Curl is Done With AI Slop - It's FOSS News

May 6, 2025 · The curl project is cracking down on low-quality AI-generated bug reports.



The Register

<https://www.theregister.com> › 2025 › 05 › 07 › curl_ai_bug_reports ...

Curl takes action against time-wasting AI bug reports

May 7, 2025 · Curl project founder Daniel Stenberg is fed up with of the deluge of AI-generated "slop" bug reports and recently introduced a checkbox to screen low-effort submissions that are draining maintainers' time. Stenberg said the amount of time it takes project maintainers to triage each AI-...

Don't: Act Like FOSS Projects Are Public Property

- FOSS is the best mode of running community projects
- But FOSS projects can be highly personal
- Owner can reject contributions
- Owner can even redistribute your code under non-free licenses if it was under a permissive license

Don't: Act Like FOSS Projects Are Public Property



Development

One year ago Redis changed its license – and lost most of its external contributors

By **Tim Anderson** - April 1, 2025

Don't: Act Like FOSS Projects are Public Property



☰ List of formerly open-source or free software

Article [Talk](#)

[Read](#)

From Wikipedia, the free encyclopedia

A Note on Copyright

- Usually: You/your organization owns the copyright (so you can reuse in any way, but the project can as well, based on the license)
- Sometimes: The project asks you to transfer the copyright (you can reuse the code depending on the project's license)

Don't: Pay for Being a Contributor

A potential scam that I'm not aware of.

Do: Have Fun and Learn!