Contributing to the Open-Source World

Will Webberley

FTS - March 2014

Overview

- Introduction to open-source
- The rise of GNU/Linux (and its role in OSS)
- Packages
- Contribution avenues

1 Introducing open source

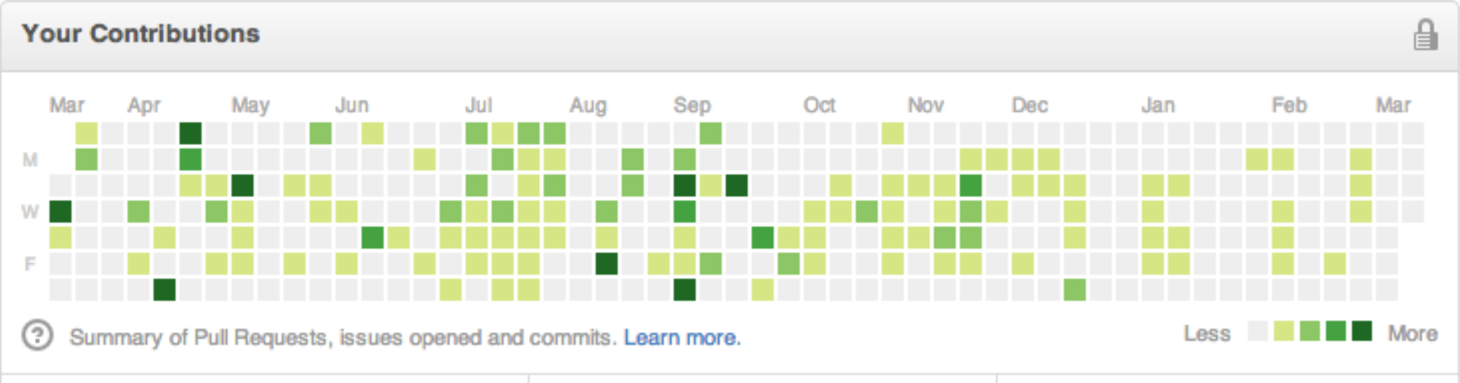
Open-Source

- Release of a program or library’s source code to public
- Encourages growth
- Encourages collaboration
- Increases trustworthiness

Why Open-Source?

- Increases authors’ visibility
- Demonstrates confidence
- Allow others to learn
- Linked to notion of reproducible research

- Many people use public Git repos
- Good to contribute as much as possible
- Can be helpful for employability


Your Contributions

Summary of Pull Requests, issues opened and commits. Learn more.

but…

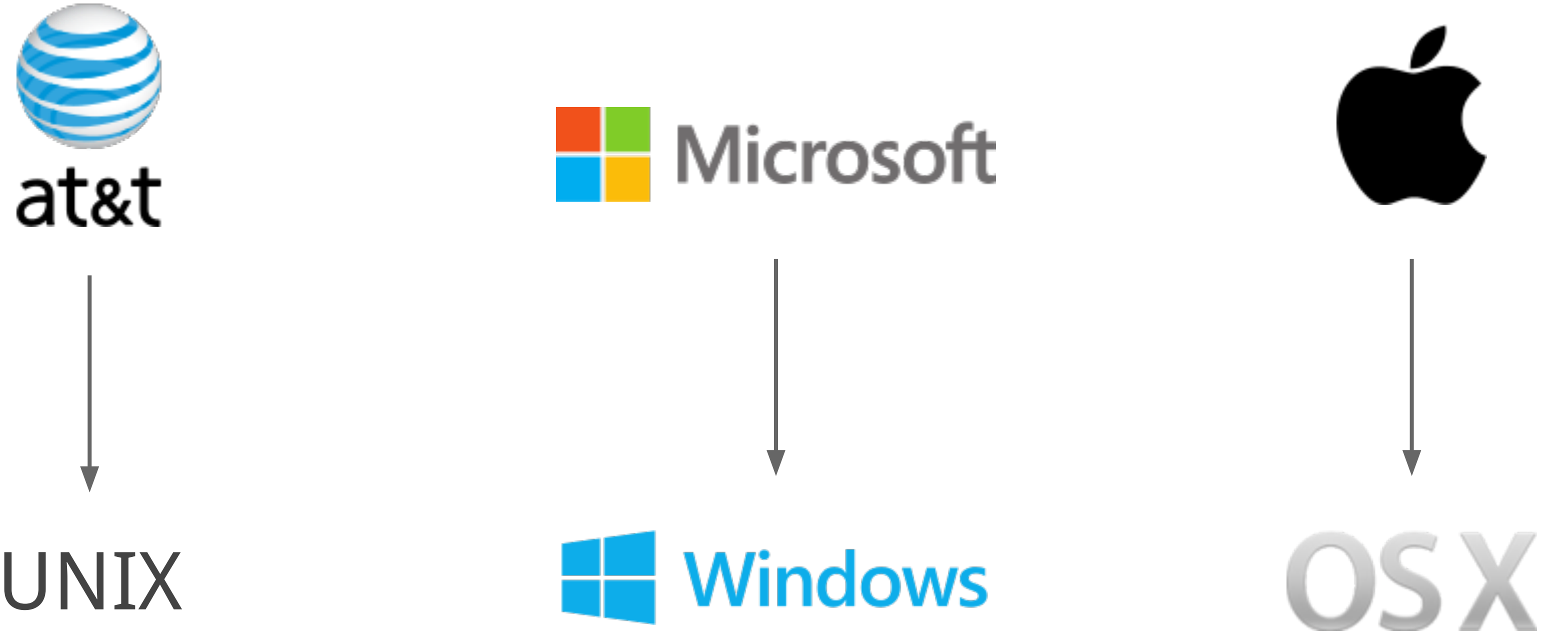
Are you only solving specific problems?

Can you contribute more widely?

2 GNU/Linux

And the rise of open source

1960s and 70s



at&t → UNIX

Microsoft → Windows

OSX

‘Starting’ the technological open source world

GNU

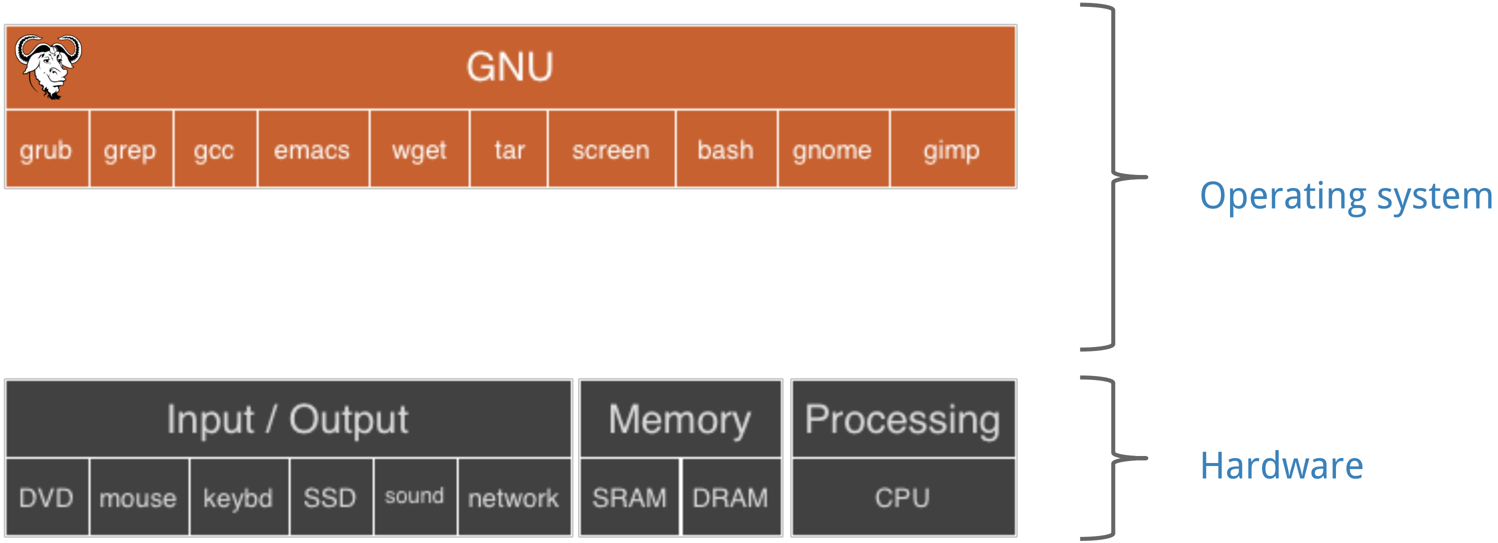
A collection of programs / libraries based on UNIX ones

'Free' & open source

GNU:

- gcc (GNU Compiler Collection)
- Emacs
- BASH (Bourne-Again SHell)
- GNOME
- tar
- screen
- GRUB (Grand Unified Boot Loader)
- grep
- GIMP (GNU Image Manipulation Program)
- wget
- etc., etc...

GNU - something missing



GNU

grub | grep | gcc | emacs | wget | tar | screen | bash | gnome | gimp

Operating system

Input / Output

DVD | mouse | keybd | SSD | sound | network

Memory

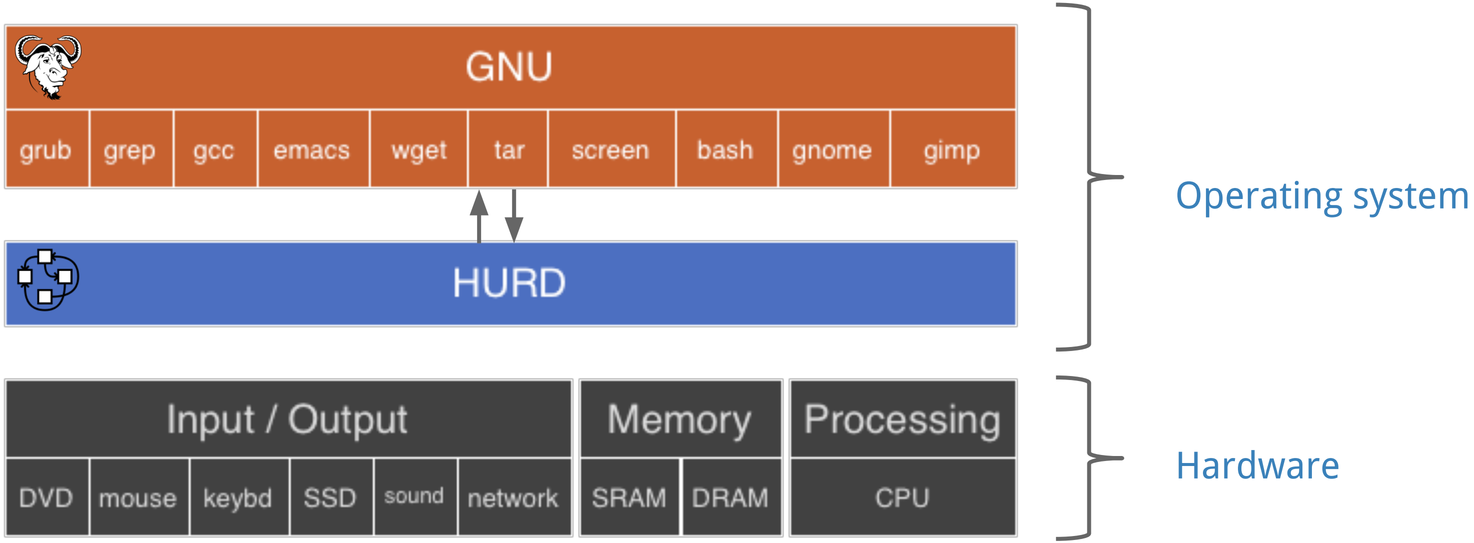
SRAM | DRAM

Processing

CPU

Hardware

GNU Hurd



GNU

grub | grep | gcc | emacs | wget | tar | screen | bash | gnome | gimp

HURD

Input / Output

DVD | mouse | keybd | SSD | sound | network

Memory

SRAM | DRAM

Processing

CPU

Operating system

Hardware

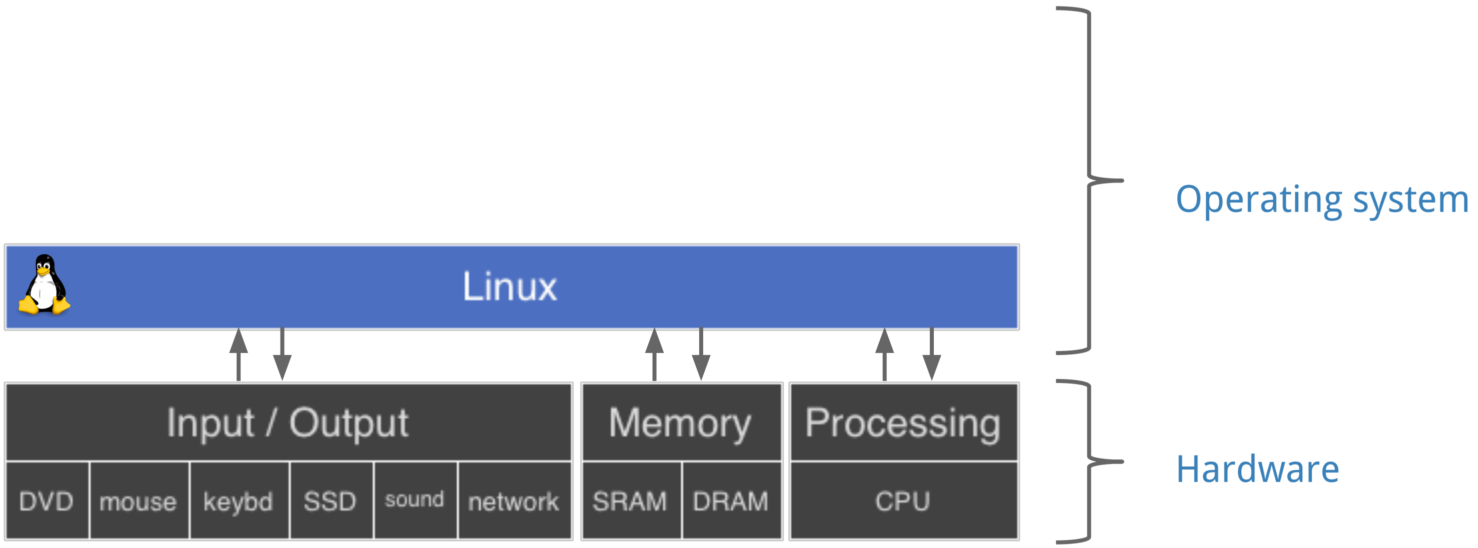
Linux

A kernel:
- handle I/O
- CPU scheduling
- memory management

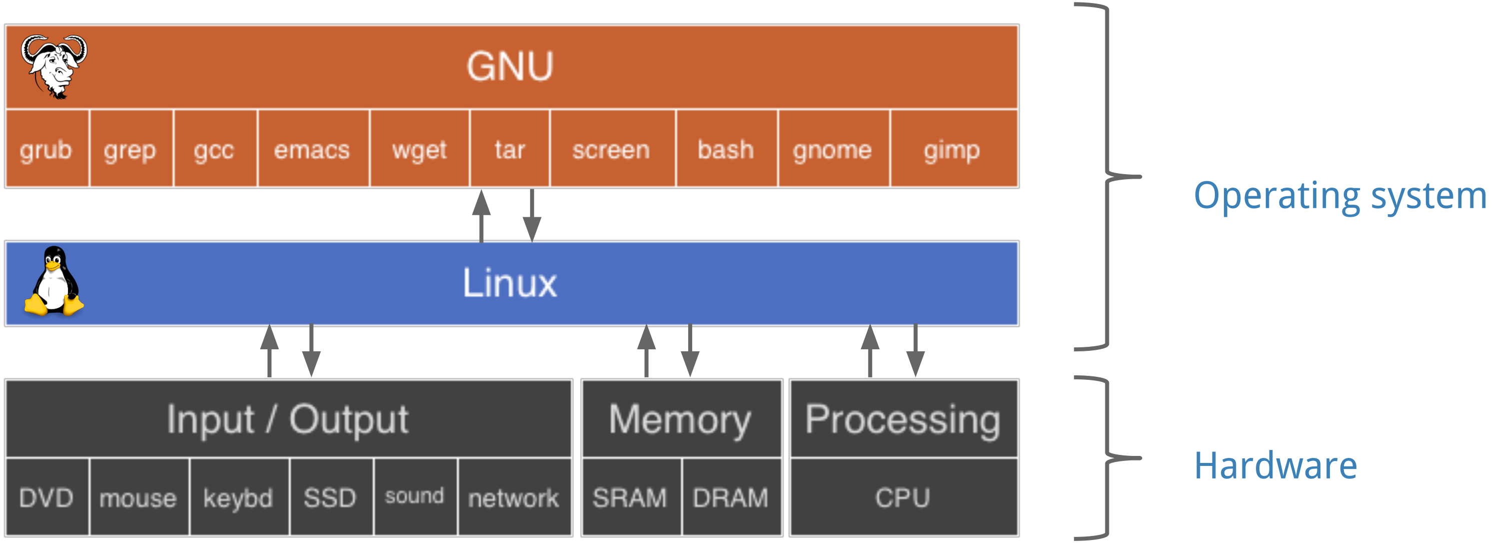
After experiencing UNIX and MINIX

LT more relaxed about FOSS

Linux - something missing



GNU / Linux


GNU
grub
grep
gcc
emacs
wget
tar
screen
bash
gnome
gimp
Linux
Input / Output
DVD
mouse
keybd
SSD
sound
network
Memory
SRAM
DRAM
Processing
CPU
Operating system
Hardware

Controversy

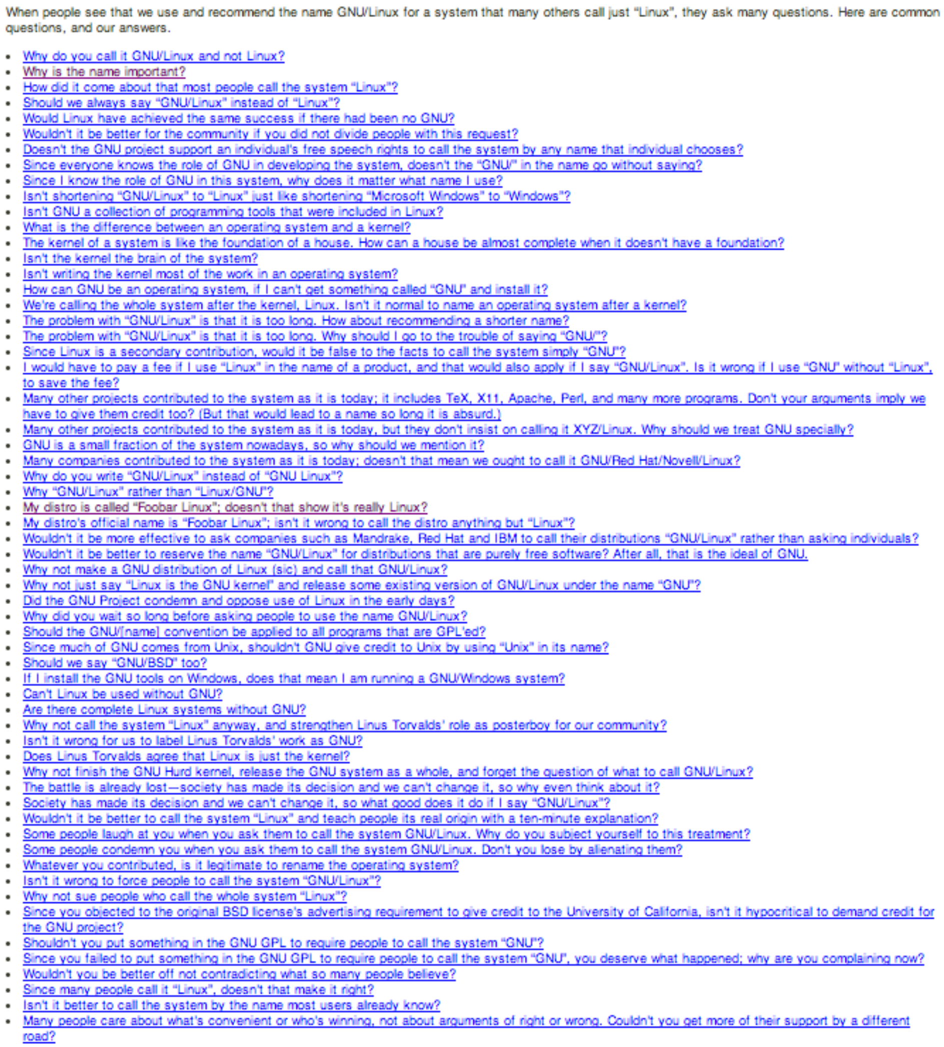
Mainly from RMS:

- Naming conventions (Linux vs. GNU/Linux)

- ‘Freeness’ of software
    - RdRnd in /dev/random

Controversy

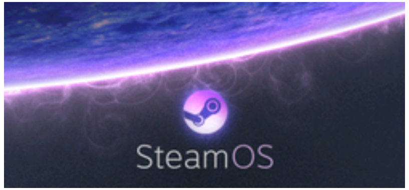
You should call Linux GNU/Linux.

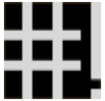


When people see that we use and recommend the name GNU/Linux for a system that many others call just “Linux”, they ask many questions. Here are common questions, and our answers.

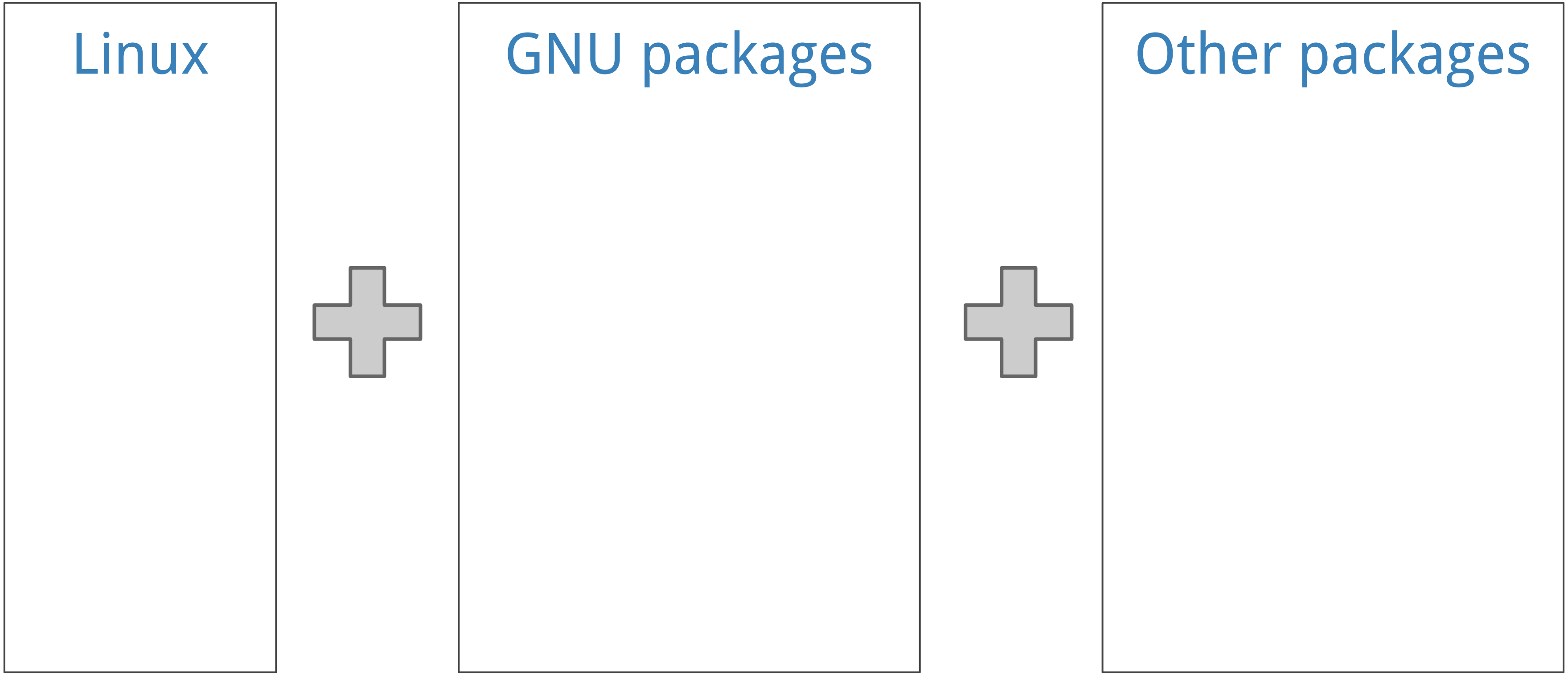
- Why do you call it GNU/Linux and not Linux?
- Why is the name important?
- How did it come about that most people call the system “Linux”?
- Should we always say “GNU/Linux” instead of “Linux”?
- Would Linux have achieved the same success if there had been no GNU?
- Wouldn't it be better for the community if you did not divide people with this request?
- Doesn't the GNU project support an individual's free speech rights to call the system by any name that individual chooses?
- Since everyone knows the role of GNU in developing the system, doesn't the “GNU/” in the name go without saying?
- Since I know the role of GNU in this system, why does it matter what name I use?
- Isn't shortening “GNU/Linux” to “Linux” just like shortening “Microsoft Windows” to “Windows”?
- Isn't GNU a collection of programming tools that were included in Linux?
- What is the difference between an operating system and a kernel?
- The kernel of a system is like the foundation of a house. How can a house be almost complete when it doesn't have a foundation?
- Isn't the kernel the brain of the system?
- Isn't writing the kernel most of the work in an operating system?
- How can GNU be an operating system, if I can't get something called “GNU” and install it?
- We're calling the whole system after the kernel, Linux. Isn't it normal to name an operating system after a kernel?
- The problem with “GNU/Linux” is that it is too long. How about recommending a shorter name?
- The problem with “GNU/Linux” is that it is too long. Why should I go to the trouble of saying “GNU/”?
- Since Linux is a secondary contribution, would it be false to the facts to call the system simply “GNU”?
- I would have to pay a fee if I use “Linux” in the name of a product, and that would also apply if I say “GNU/Linux”. Is it wrong if I use “GNU” without “Linux”, to save the fee?
- Many other projects contributed to the system as it is today; it includes TeX, X11, Apache, Perl, and many more programs. Don't your arguments imply we have to give them credit too? (But that would lead to a name so long it is absurd.)
- Many other projects contributed to the system as it is today, but they don't insist on calling it XYZ/Linux. Why should we treat GNU specially?
- GNU is a small fraction of the system nowadays, so why should we mention it?
- Many companies contributed to the system as it is today; doesn't that mean we ought to call it GNU/Red Hat/Novell/Linux?
- Why do you write “GNU/Linux” instead of “GNU Linux”?
- Why “GNU/Linux” rather than “Linux/GNU”?
- My distro is called “Foobar Linux”; doesn't that show it's really Linux?
- My distro's official name is “Foobar Linux”; isn't it wrong to call the distro anything but “Linux”?
- Wouldn't it be more effective to ask companies such as Mandrake, Red Hat and IBM to call their distributions “GNU/Linux” rather than asking individuals?
- Wouldn't it be better to reserve the name “GNU/Linux” for distributions that are purely free software? After all, that is the ideal of GNU.
- Why not make a GNU distribution of Linux (sic) and call that GNU/Linux?
- Why not just say “Linux is the GNU kernel” and release some existing version of GNU/Linux under the name “GNU”?
- Did the GNU Project condemn and oppose use of Linux in the early days?
- Why did you wait so long before asking people to use the name GNU/Linux?
- Should the GNU/[name] convention be applied to all programs that are GPL'ed?
- Since much of GNU comes from Unix, shouldn't GNU give credit to Unix by using “Unix” in its name?
- Should we say “GNU/BSD” too?
- If I install the GNU tools on Windows, does that mean I am running a GNU/Windows system?
- Can't Linux be used without GNU?
- Are there complete Linux systems without GNU?
- Why not call the system “Linux” anyway, and strengthen Linus Torvalds' role as posterboy for our community?
- Isn't it wrong for us to label Linus Torvalds' work as GNU?
- Does Linus Torvalds agree that Linux is just the kernel?
- Why not finish the GNU Hurd kernel, release the GNU system as a whole, and forget the question of what to call GNU/Linux?
- The battle is already lost—society has made its decision and we can't change it, so why even think about it?
- Society has made its decision and we can't change it, so what good does it do if I say “GNU/Linux”?
- Wouldn't it be better to call the system “Linux” and teach people its real origin with a ten-minute explanation?
- Some people laugh at you when you ask them to call the system GNU/Linux. Why do you subject yourself to this treatment?
- Some people condemn you when you ask them to call the system GNU/Linux. Don't you lose by alienating them?
- Whatever you contributed, is it legitimate to rename the operating system?
- Isn't it wrong to force people to call the system “GNU/Linux”?
- Why not sue people who call the whole system “Linux”?
- Since you objected to the original BSD license's advertising requirement to give credit to the University of California, isn't it hypocritical to demand credit for the GNU project?
- Shouldn't you put something in the GNU GPL to require people to call the system “GNU”?
- Since you failed to put something in the GNU GPL to require people to call the system “GNU”, you deserve what happened; why are you complaining now?
- Wouldn't you be better off not contradicting what so many people believe?
- Since many people call it “Linux”, doesn't that make it right?
- Isn't it better to call the system by the name most users already know?
- Many people care about what's convenient or who's winning, not about arguments of right or wrong. Couldn't you get more of their support by a different road?

GNU/Linux over the years





GNU/Linux distributions

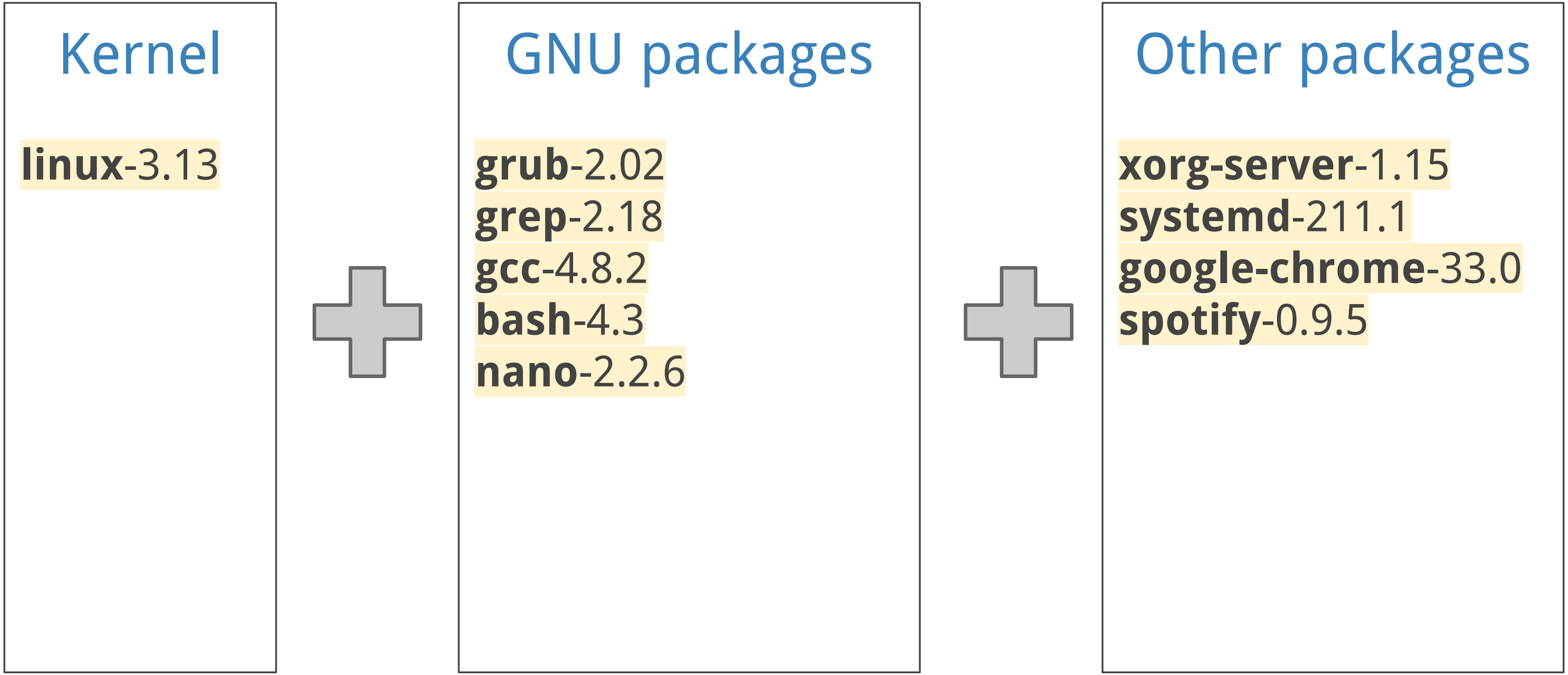


Linux

GNU packages

Other packages

Custom distribution



Kernel

linux-3.13

+

GNU packages

grub-2.02
grep-2.18
gcc-4.8.2
bash-4.3
nano-2.2.6

+

Other packages

xorg-server-1.15
systemd-211.1
google-chrome-33.0
spotify-0.9.5

3 Packages

Packages

- Format varies between distributions

- Can provide a library, an app, a script, etc.

- May contain source or compiled binaries (depending on distro)

- Usually a compressed directory containing files and installation routines

Distribution packages

- Can be exclusive

- Stored in repositories to be downloaded the by distro’s users

- Distributors **not** usually the *author* of the packages

- Distributors merely maintain a version for their distro

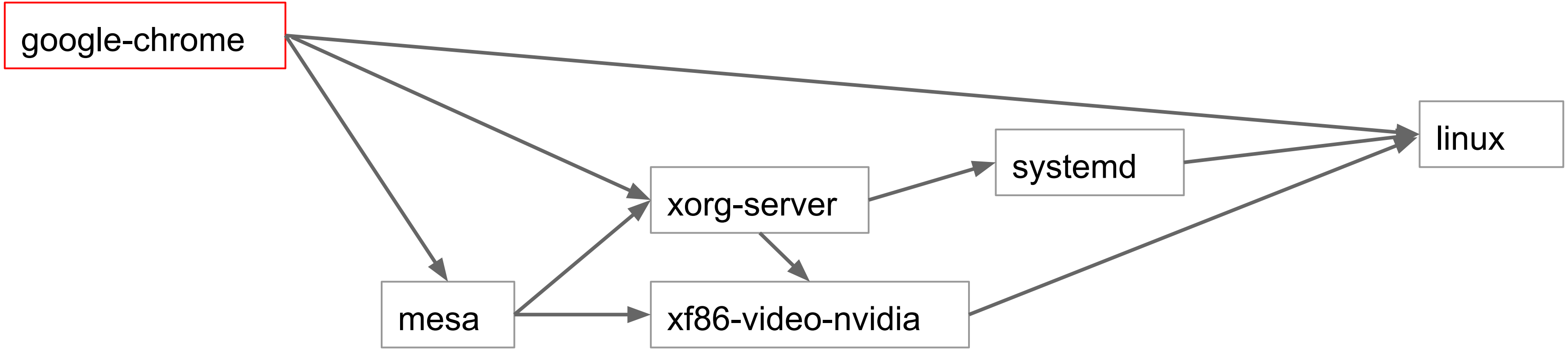
- If package not in your distro’s repositories, can build it yourself!

- But be wary of dependencies

Package dependencies

Some packages rely on others in order to work

For example, Google Chrome:



Package dependencies

For example, if your package contains a Java program, you’ll likely need the `jdk` and `jre` packages.

Can get confusing to manage!

Package managers

Most distributions come with a package manager as a default package

debian ubuntu #!	Aptitude	`apt-get install firefox`
fedora	yum	`yum install firefox`
archlinux A simple, lightweight linux distribution.	pacman	`pacman -S firefox`
	Google Play	

Package managers

They'll download your package and its dependencies

… from your distro's repositories

… and in the right order!

Bringing ‘downstream’

- Packages not usually written by distributors

- Instead they are brought downstream from the source (usually Git)

- Can then be modified to suit the distro

- Packaged up for installation by distro’s users

- Distributor *maintains* the downstream version for the distro

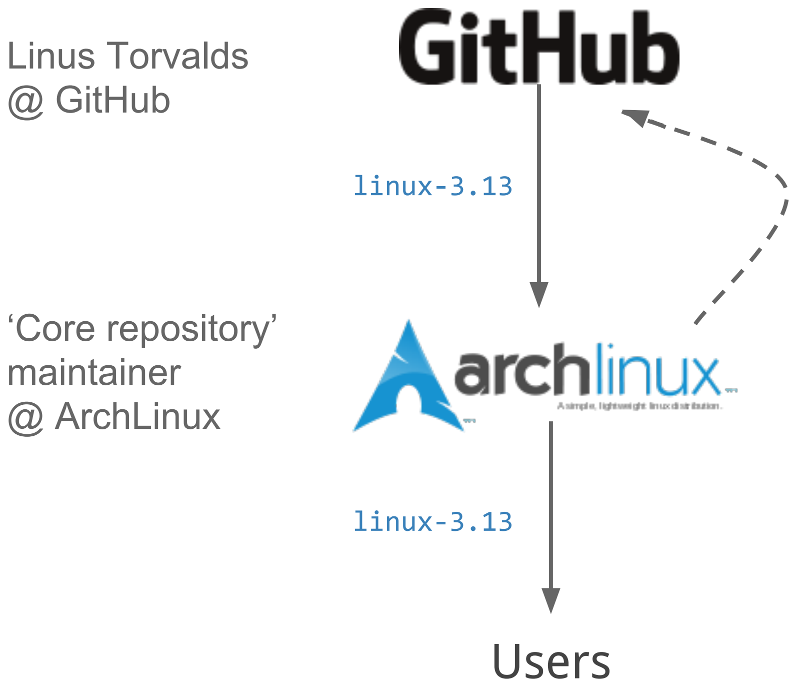
Pushing ‘upstream’

- If distributer has to fix a particular issue on every update, can push upstream

- Future versions by the author will then have this patch

Upstream and downstream

Typically, all GNU/Linux packages work in this way



Linus Torvalds
@ GitHub

linux-3.13

‘Core repository’
maintainer
@ ArchLinux

linux-3.13

Users

4 Contributing

Contributors

- Open source movement is largely powered by volunteers

- Community of hobbyists

- Relies on people giving up some time

Why contribute?

- If you use OSS, your chance to give back!

- Belong to a community

- Gain respect / kudos

- Become an expert

- Shows evidence for employability

- You’re passionate

Where to start?

- Can be daunting
- But there are many avenues for contribution:
    - Contribute documentation
    - Join a *useful* project
    - Maintain a package

Contributing documentation

Non-coding contributors: **as important as code!**

- Wiki (do you know something that may be useful to others?)

- Articles / tutorials (you may want to share how you solved a problem)

- Artwork (graphic designers are expensive)

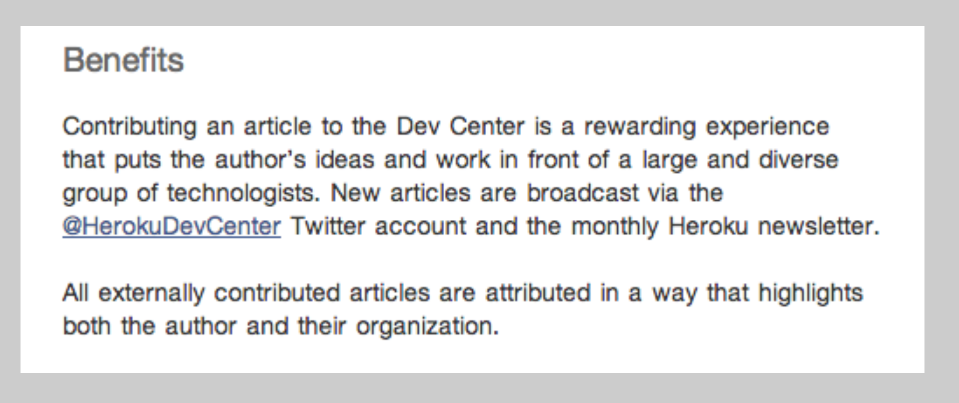
- Manuals (man-pages)

- etc.

Contributing documentation

e.g. Heroku Dev Center

devcenter.heroku.com/articles/open-topics



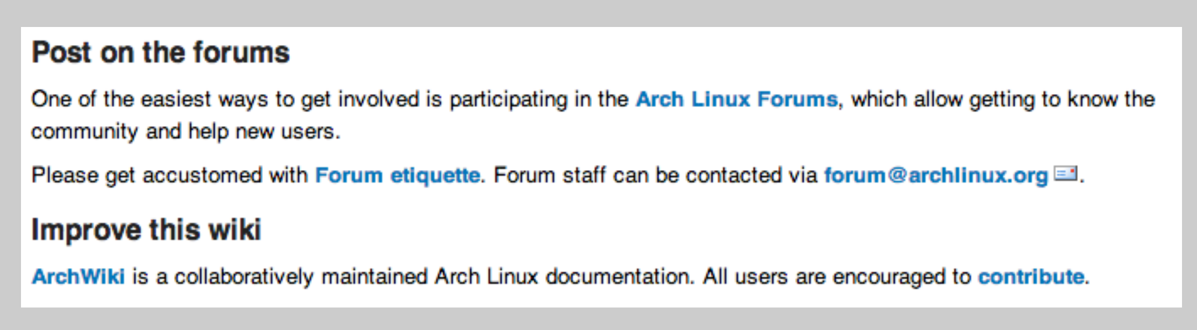
Benefits

Contributing an article to the Dev Center is a rewarding experience that puts the author's ideas and work in front of a large and diverse group of technologists. New articles are broadcast via the @HerokuDevCenter Twitter account and the monthly Heroku newsletter.

All externally contributed articles are attributed in a way that highlights both the author and their organization.

e.g. Arch Linux

wiki.archlinux.org/index.php/Getting_involved



Post on the forums

One of the easiest ways to get involved is participating in the Arch Linux Forums, which allow getting to know the community and help new users.

Please get accustomed with Forum etiquette. Forum staff can be contacted via forum@archlinux.org.

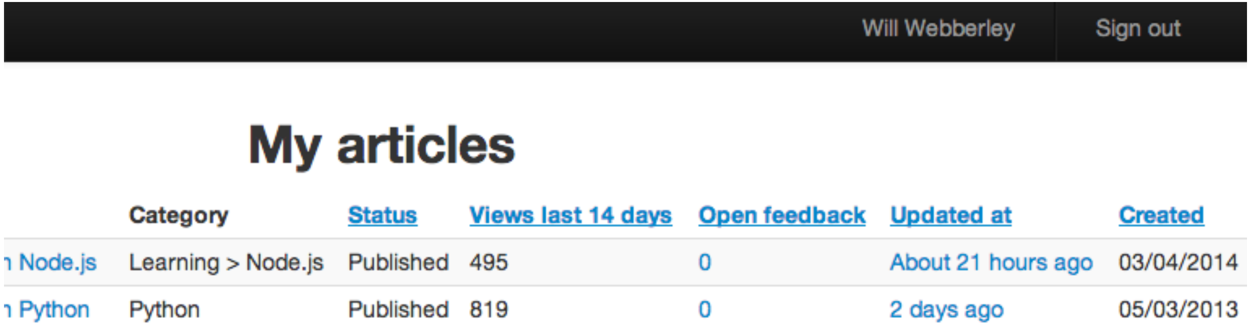
Improve this wiki

ArchWiki is a collaboratively maintained Arch Linux documentation. All users are encouraged to contribute.

Contributing documentation

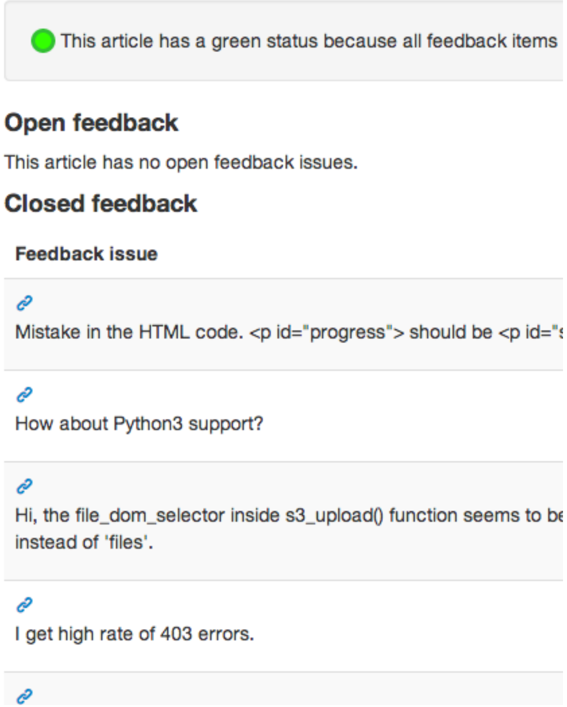
- Quickly gain visibility

- Keep up-to-date



My articles

	Category	Status	Views last 14 days	Open feedback	Updated at	Created
Node.js	Learning > Node.js	Published	495	0	About 21 hours ago	03/04/2014
Python	Python	Published	819	0	2 days ago	05/03/2013



● This article has a green status because all feedback items

Open feedback

This article has no open feedback issues.

Closed feedback

Feedback issue

Mistake in the HTML code. <p id="progress"> should be <p id="

How about Python3 support?

Hi, the file_dom_selector inside s3_upload() function seems to be instead of 'files'.

I get high rate of 403 errors.

Contribute to the Linux kernel?

- Probably not a good place to start

- Harder!

- You have to write in C

- You have to deal with Linus Torvalds (very strict)

But, if you do want to...

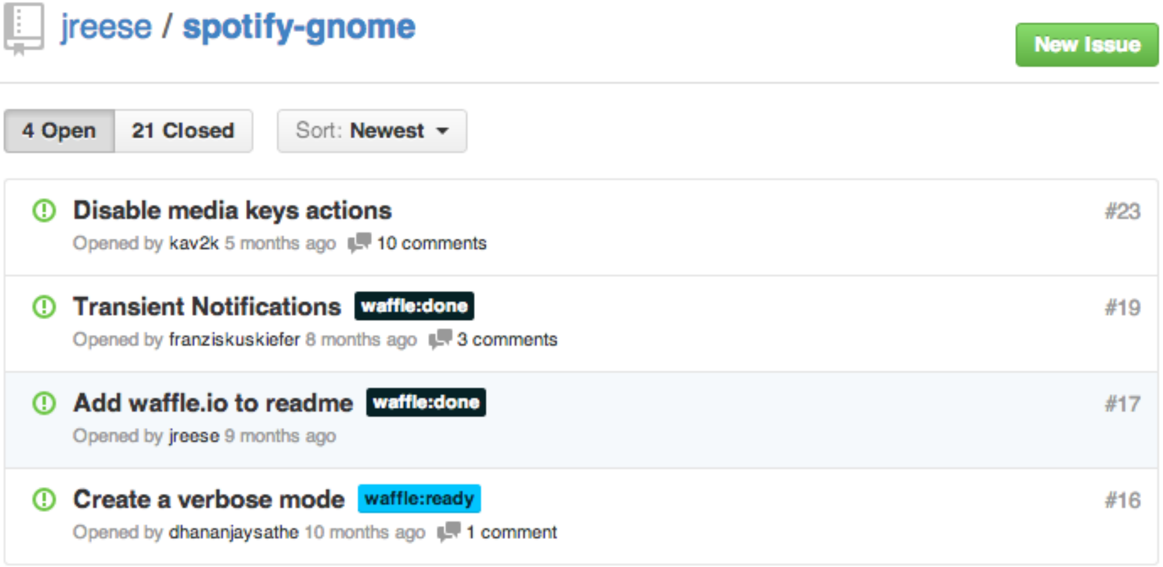
- Start by checking for bugs (or typos): bugzilla.kernel.org

Join a project

- If you can fix an issue, do it!

- (even if only fixing readme typos)

- Join discussions



jreese / spotify-gnome

Disable media keys actions #23
Opened by kav2k 5 months ago 10 comments

Transient Notifications waffle:done #19
Opened by franziskuskiefer 8 months ago 3 comments

Add waffle.io to readme waffle:done #17
Opened by jreese 9 months ago

Create a verbose mode waffle:ready #16
Opened by dhananjaysathe 10 months ago 1 comment

Maintain a package

- Become a maintainer for a package

- Maintain an interesting package for a distro that you use

2 main ways to do this:

- Submit your own package

- Adopt an orphaned package

Submitting a package to maintain

Your own code (game, library, utility, etc.)

- e.g. A project you worked on that'd be useful to others

Someone else's code

- e.g. A public project
- Author may not have time to package it for your distro
- Don't submit a package that already exists in some form

Adopting a package to maintain

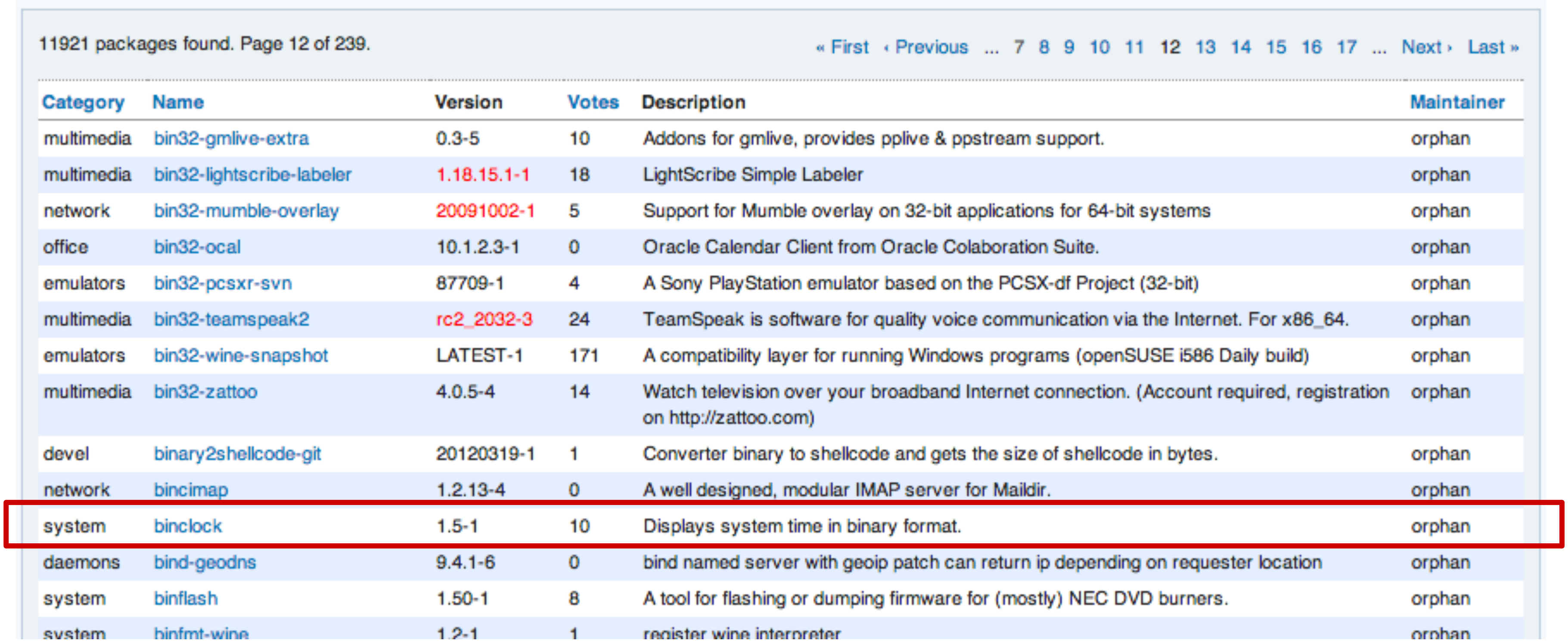
- A good place to start!

- People unable to continue maintaining can orphan their packages

- Good opportunity

- Many distributions encourage adoption and list them

Adopting a package to maintain



11921 packages found. Page 12 of 239.

« First ‹ Previous ... 7 8 9 10 11 12 13 14 15 16 17 ... Next › Last »

Category	Name	Version	Votes	Description	Maintainer
multimedia	bin32-gmlive-extra	0.3-5	10	Addons for gmlive, provides pplive & ppstream support.	orphan
multimedia	bin32-lightscribe-labeler	1.18.15.1-1	18	LightScribe Simple Labeler	orphan
network	bin32-mumble-overlay	20091002-1	5	Support for Mumble overlay on 32-bit applications for 64-bit systems	orphan
office	bin32-ocal	10.1.2.3-1	0	Oracle Calendar Client from Oracle Colaboration Suite.	orphan
emulators	bin32-pcsxr-svn	87709-1	4	A Sony PlayStation emulator based on the PCSX-df Project (32-bit)	orphan
multimedia	bin32-teamspeak2	rc2_2032-3	24	TeamSpeak is software for quality voice communication via the Internet. For x86_64.	orphan
emulators	bin32-wine-snapshot	LATEST-1	171	A compatibility layer for running Windows programs (openSUSE i586 Daily build)	orphan
multimedia	bin32-zattoo	4.0.5-4	14	Watch television over your broadband Internet connection. (Account required, registration on http://zattoo.com)	orphan
devel	binary2shellcode-git	20120319-1	1	Converter binary to shellcode and gets the size of shellcode in bytes.	orphan
network	bincimap	1.2.13-4	0	A well designed, modular IMAP server for Maildir.	orphan
system	binclock	1.5-1	10	Displays system time in binary format.	orphan
daemons	bind-geodns	9.4.1-6	0	bind named server with geoip patch can return ip depending on requester location	orphan
system	binflash	1.50-1	8	A tool for flashing or dumping firmware for (mostly) NEC DVD burners.	orphan
system	binfmt-wine	1.2-1	1	register wine interpreter	orphan

Adopting a package to maintain

- Usually just needs updating from upstream

- Just package it up for your distro when upstream author updates

- Otherwise, just keep the code up-to-date with dependencies

- When popular enough, can get moved to official repositories

When you have a package

- Either through adoption or submission

- Keep using it yourself

- Be enthusiastic

- Watch feedback

- Be interactive with your users!

5 Closing remarks

Moving on...

When more confident...

- Contribute more documentation
- Adopt / submit more packages
- Takes as little as 1-2 hours / month

The more you do…

- You keep up-to-date with advancements
- Moves you on to contributing to some official repositories

Contributing to the Open-Source World

Thank you

Will Webberley