

GNU/Linux FAQ by Richard Stallman

To learn more about this issue, you can also read our page on [Linux and the GNU Project](#), our page on [Why GNU/Linux?](#) and our page on [GNU Users Who Have Never Heard of GNU](#).

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?](#)
- [Can another system have “the feel of Linux”?](#)
- [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/”?](#)
- [Unfortunately, “GNU/Linux” is five syllables. People won't use such a long term. Shouldn't you find a shorter one?](#)
- [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 “Foofoo Linux”; doesn't that show it's really Linux?
 - My distro's official name is “Foofoo 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 [sic] 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?](#)

Why do you call it GNU/Linux and not Linux? ([#why](#))

Most operating system distributions based on Linux as kernel are basically modified versions of the GNU operating system. We began developing GNU in 1984, years before Linus Torvalds started to write his kernel. Our goal was to develop a complete free operating system. Of course, we did not develop all the parts ourselves—but we led the way. We developed most of the central components, forming the largest single contribution to the whole system. The basic vision was ours too.

In fairness, we ought to get at least equal mention.

See [Linux and the GNU System](#) and [GNU Users Who Have Never Heard of GNU](#) for more explanation, and [The GNU Project](#) for the history.

Why is the name important? ([#whycare](#))

Although the developers of Linux, the kernel, are contributing to the free software community, many of them do not care about freedom. People who think the whole system is Linux tend to get confused and assign to those developers a role in the history of our community which they did not actually play. Then they give inordinate weight to those developers' views.

Calling the system GNU/Linux recognizes the role that our idealism played in building our community, and [helps the public recognize the practical importance of these ideals](#).

How did it come about that most people call the system "Linux"? ([#howerror](#))

Calling the system "Linux" is a confusion that has spread faster than the corrective

information.

The people who combined Linux with the GNU system were not aware that that's what their activity amounted to. They focused their attention on the piece that was Linux and did not realize that more of the combination was GNU. They started calling it "Linux" even though that name did not fit what they had. It took a few years for us to realize what a problem this was and ask people to correct the practice. By that time, the confusion had a big head start.

Most of the people who call the system "Linux" have never heard why that's not the right thing. They saw others using that name and assume it must be right. The name "Linux" also spreads a false picture of the system's origin, because people tend to suppose that the system's history was such as to fit that name. For instance, they often believe its development was started by Linus Torvalds in 1991. This false picture tends to reinforce the idea that the system should be called "Linux".

Many of the questions in this file represent people's attempts to justify the name they are accustomed to using.

Should we always say "GNU/Linux" instead of "Linux"? ([#always](#))

Not always—only when you're talking about the whole system. When you're referring specifically to the kernel, you should call it "Linux", the name its developer chose.

When people call the whole system "Linux", as a consequence they call the whole system by the same name as the kernel. This causes many kinds of confusion, because only experts can tell whether a statement is about the kernel or the whole system. By calling the whole system "GNU/Linux", and calling the kernel "Linux", you avoid the ambiguity.

Would Linux have achieved the same success if there had been no GNU?

([#linuxalone](#))

In that alternative world, there would be nothing today like the GNU/Linux system, and probably no free operating system at all. No one attempted to develop a free operating system in the 1980s except the GNU Project and (later) Berkeley CSRG, which had been specifically asked by the GNU Project to start freeing its code.

Linus Torvalds was partly influenced by a speech about GNU in Finland in 1990. It's possible that even without this influence he might have written a Unix-like kernel, but it probably would not have been free software. Linux became free in 1992 when Linus rereleased it under the GNU GPL. (See the release notes for version 0.12.)

Even if Torvalds had released Linux under some other free software license, a free kernel alone would not have made much difference to the world. The significance of Linux came from fitting into a larger framework, a complete free operating system:

GNU/Linux.

Wouldn't it be better for the community if you did not divide people with this request? ([#divide](#))

When we ask people to say “GNU/Linux”, we are not dividing people. We are asking them to give the GNU Project credit for the GNU operating system. This does not criticize anyone or push anyone away.

However, there are people who do not like our saying this. Sometimes those people push us away in response. On occasion they are so rude that one wonders if they are intentionally trying to intimidate us into silence. It doesn't silence us, but it does tend to divide the community, so we hope you can convince them to stop.

However, this is only a secondary cause of division in our community. The largest division in the community is between people who appreciate free software as a social and ethical issue and consider proprietary software a social problem (supporters of the free software movement), and those who cite only practical benefits and present free software only as an efficient development model (the open source movement).

This disagreement is not just a matter of names—it is a matter of differing basic values. It is essential for the community to see and think about this disagreement. The names “free software” and “open source” are the banners of the two positions. See [Why Open Source misses the point of Free Software](#).

The disagreement over values partially aligns with the amount of attention people pay to the GNU Project's role in our community. People who value freedom are more likely to call the system “GNU/Linux”, and people who learn that the system is “GNU/Linux” are more likely to pay attention to our philosophical arguments for freedom and community (which is why the choice of name for the system makes a real difference for society). However, the disagreement would probably exist even if everyone knew the system's real origin and its proper name, because the issue is a real one. It can only go away if we who value freedom either persuade everyone (which won't be easy) or are defeated entirely (let's hope not).

Doesn't the GNU project support an individual's free speech rights to call the system by any name that individual chooses? ([#freespeech](#))

Yes, indeed, we believe you have a free speech right to call the operating system by any name you wish. We ask that people call it GNU/Linux as a matter of doing justice to the GNU project, to promote the values of freedom that GNU stands for, and to inform others that those values of freedom brought the system into existence.

Since everyone knows GNU's role in developing the system, doesn't the “GNU/” in the name go without saying? ([#everyoneknows](#))

Experience shows that the system's users, and the computer-using public in general, often know nothing about the GNU system. Most articles about the system do not mention the name “GNU”, or the ideals that GNU stands for. [GNU Users Who Have Never Heard of GNU](#) explains further.

The people who say this are probably geeks thinking of the geeks they know. Geeks often do know about GNU, but many have a completely wrong idea of what GNU is. For instance, many think it is a collection of [“tools”](#), or a project to develop tools.

The wording of this question, which is typical, illustrates another common misconception. To speak of “GNU's role” in developing something assumes that GNU is a group of people. GNU is an operating system. It would make sense to talk about the GNU Project's role in this or some other activity, but not that of GNU.

Since I know the role of GNU in this system, why does it matter what name I use? ([#everyoneknows2](#))

If your words don't reflect your knowledge, you don't teach others. Most people who have heard of the GNU/Linux system think it is “Linux”, that it was started by Linus Torvalds, and that it was intended to be “open source”. If you don't tell them, who will?

Isn't shortening “GNU/Linux” to “Linux” just like shortening “Microsoft Windows” to “Windows”? ([#windows](#))

It's useful to shorten a frequently-used name, but not if the abbreviation is misleading.

Almost everyone in developed countries really does know that the “Windows” system is made by Microsoft, so shortening “Microsoft Windows” to “Windows” does not mislead anyone as to that system's nature and origin. Shortening “GNU/Linux” to “Linux” does give the wrong idea of where the system comes from.

The question is itself misleading because GNU and Microsoft are not the same kind of thing. Microsoft is a company; GNU is an operating system.

Isn't GNU a collection of programming tools that were included in Linux? ([#tools](#))

People who think that Linux is an entire operating system, if they hear about GNU at all, often get a wrong idea of what GNU is. They may think that GNU is the name of a collection of programs—often they say “programming tools”, since some of our programming tools became popular on their own. The idea that “GNU” is the name of an operating system is hard to fit into a conceptual framework in which that operating system is labeled “Linux”.

The GNU Project was named after the GNU operating system—it's the project to develop the GNU system. (See [the 1983 initial announcement](#).)

We developed programs such as GCC, GNU Emacs, GAS, GLIBC, BASH, etc., because we needed them for the GNU operating system. GCC, the GNU Compiler Collection is the compiler that we wrote for the GNU operating system. We, the many people working on the GNU Project, developed Ghostscript, GNUCash, GNU Chess and GNOME for the GNU system too.

What is the difference between an operating system and a kernel? ([#osvskernel](#))

An operating system, as we use the term, means a collection of programs that are sufficient to use the computer to do a wide variety of jobs. A general purpose operating system, to be complete, ought to handle all the jobs that many users may want to do.

The kernel is one of the programs in an operating system—the program that allocates the machine's resources to the other programs that are running. The kernel also takes care of starting and stopping other programs.

To confuse matters, some people use the term “operating system” to mean “kernel”. Both uses of the term go back many years. The use of “operating system” to mean “kernel” is found in a number of textbooks on system design, going back to the 80s. At the same time, in the 80s, the “Unix operating system” was understood to include all the system programs, and Berkeley's version of Unix included even games. Since we intended GNU to be a Unix-like operating system, we use the term “operating system” in the same way.

Most of the time when people speak of the “Linux operating system” they are using “operating system” in the same sense we use: they mean the whole collection of programs. If that's what you are referring to, please call it “GNU/Linux”. If you mean just the kernel, then “Linux” is the right name for it, but please say “kernel” also to avoid ambiguity about which body of software you mean.

If you prefer to use some other term such as “system distribution” for the entire collection of programs, instead of “operating system”, that's fine. Then you would talk about GNU/Linux system distributions.

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? ([#house](#))

A kernel is not much like the foundation of a house because building an operating system is not much like building a house.

A house is built from lots of little general parts that are cut and put together in situ. They have to be put together from the bottom up. Thus, when the foundation has not been built, no substantial part has been built; all you have is a hole in the ground.

By contrast, an operating system consists of complex components that can be

developed in any order. When you have developed most of the components, most of the work is done. This is much more like the International Space Station than like a house. If most of the Space Station modules were in orbit but awaiting one other essential module, that would be like the GNU system in 1992.

Isn't the kernel the brain of the system? ([#brain](#))

A computer system is not much like a human body, and no part of it plays a role comparable to that of the brain in a human.

Isn't writing the kernel most of the work in an operating system? ([#kernelmost](#))

No, many components take a lot of work.

How can GNU be an operating system, if I can't get something called "GNU" and install it? ([#notinstallable](#))

Many [packaged and installable versions of GNU](#) are available. None of them is called simply "GNU", but GNU is what they basically are.

We expected to release the GNU system packaged for installation, but this plan was overtaken by events: in 1992 others were already packaging GNU variants containing Linux. Starting in 1993 we sponsored an effort to make a better and freer GNU/Linux distribution, called [Debian GNU/Linux](#). The founder of Debian had already chosen that name. We did not ask him to call it just "GNU" because that was to be the name of a system version with the GNU Hurd kernel—which wasn't ready yet.

The GNU Hurd kernel never became sufficiently ready; we only recommend it to those interested in working on it. So we never packaged GNU with the GNU Hurd kernel. However, Debian packaged this combination as Debian GNU/Hurd.

We are now developing an advanced Scheme-based package manager called GUIX, and this includes repackaging a substantial part of the GNU system.

We never took the last step of packaging GNU under the name "GNU", but that doesn't alter what kind of thing GNU is. GNU is an operating system.

We're calling the whole system after the kernel, Linux. Isn't it normal to name an operating system after a kernel? ([#afterkernel](#))

That practice seems to be very rare—we can't find any examples other than the misuse of the name "Linux". Normally an operating system is developed as a single unified project, and the developers choose a name for the system as a whole. The kernel usually does not have a name of its own—instead, people say "the kernel of

such-and-such” or “the such-and-such kernel”.

Because those two constructions are used synonymously, the expression “the Linux kernel” can easily be misunderstood as meaning “the kernel of Linux” and implying that Linux must be more than a kernel. You can avoid the possibility of this misunderstanding by saying or writing “the kernel, Linux” or “Linux, the kernel.”

Can another system have “the feel of Linux”? ([#feel](#))

There is no such thing as the “feel of Linux” because Linux has no user interfaces. Like any modern kernel, Linux is a base for running programs; user interfaces belong elsewhere in the system. Human interaction with GNU/Linux always goes through other programs, and the “feel” comes from them.

The problem with “GNU/Linux” is that it is too long. How about recommending a shorter name? ([#long](#))

For a while we tried the name “LiGNUx”, which combines the words “GNU” and “Linux”. The reaction was very bad. People accept “GNU/Linux” much better.

The shortest legitimate name for this system is “GNU”, but we call it “GNU/Linux” [for the reasons given below](#).

The problem with “GNU/Linux” is that it is too long. Why should I go to the trouble of saying “GNU”? ([#long2](#))

It only takes a second to say or type “GNU”. If you appreciate the system that we developed, can't you take one second to recognize our work?

Unfortunately, “GNU/Linux” is five syllables. People won't use such a long term. Shouldn't you find a shorter one? ([#long3](#))

Actually, “GNU/Linux” is only four syllables. “Unfortunately” is five syllables, yet people show no sign of reluctance to use that word.

Since Linux is a secondary contribution, would it be false to the facts to call the system simply “GNU”? ([#justgnu](#))

It would not be false to the facts, but it is not the best thing to do. Here are the reasons we call that system version “GNU/Linux” rather than just “GNU”:

- It's not exactly GNU—it has a different kernel (that is, Linux). Distinguishing GNU/Linux from GNU is useful.
- It would be ungentlemanly to ask people to *stop* giving any credit to Linus

Torvalds. He did write an important component of the system. We want to get credit for launching and sustaining the system's development, but this doesn't mean we should treat Linus the same way those who call the system "Linux" treat us. We strongly disagree with his political views, but we deal with that disagreement honorably and openly, rather than by trying to cut him out of the credit for his contribution to the system.

- Since many people know of the system as "Linux", if we say "GNU" they may simply not recognize we're talking about the same system. If we say "GNU/Linux", they can make a connection to what they have heard about.

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? ([#trademarkfee](#))

There's nothing wrong in calling the system "GNU"; basically, that's what it is. It is nice to give Linus Torvalds a share of the credit as well, but you have no obligation to pay for the privilege of doing so.

So if you want to refer to the system simply as "GNU", to avoid paying the fee for calling it "Linux", we won't criticize you.

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](#))

What we say is that you ought to give the system's principal developer a share of the credit. The principal developer is the GNU Project, and the system is basically GNU.

If you feel even more strongly about giving credit where it is due, you might feel that some secondary contributors also deserve credit in the system's name. If so, far be it from us to argue against it. If you feel that X11 deserves credit in the system's name, and you want to call the system GNU/X11/Linux, please do. If you feel that Perl simply cries out for mention, and you want to write GNU/Linux/Perl, go ahead.

Since a long name such as GNU/X11/Apache/Linux/TeX/Perl/Python/FreeCiv becomes absurd, at some point you will have to set a threshold and omit the names of the many other secondary contributions. There is no one obvious right place to set the threshold, so wherever you set it, we won't argue against it.

Different threshold levels would lead to different choices of name for the system. But one name that cannot result from concerns of fairness and giving credit, not for any possible threshold level, is "Linux". It can't be fair to give all the credit to one secondary contribution (Linux) while omitting the principal contribution (GNU).

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? ([#others](#))

Thousands of projects have developed programs commonly included in today's GNU/Linux systems. They all deserve credit for their contributions, but they aren't the principal developers of the system as a whole, so they don't ask to be credited as such.

GNU is different because it is more than just a contributed program, more than just a collection of contributed programs. GNU is the framework on which the system was made.

GNU is a small fraction of the system nowadays, so why should we mention it? ([#allsmall](#))

In 2008, we found that GNU packages made up 15% of the “main” repository of the gNewSense GNU/Linux distribution. Linux made up 1.5%. So the same argument would apply even more strongly to calling it “Linux”.

GNU is a small fraction of the system nowadays, and Linux is an even smaller fraction. But they are the system's core; the system was made by combining them. Thus, the name “GNU/Linux” remains appropriate.

Many companies contributed to the system as it is today; doesn't that mean we ought to call it GNU/Red Hat/Novell/Linux? ([#manycompanies](#))

GNU is not comparable to Red Hat or Novell; it is not a company, or an organization, or even an activity. GNU is an operating system. (When we speak of the GNU Project, that refers to the project to develop the GNU system.) The GNU/Linux system is based on GNU, and that's why GNU ought to appear in its name.

Much of those companies' contribution to the GNU/Linux system lies in the code they have contributed to various GNU packages including GCC and GNOME. Saying GNU/Linux gives credit to those companies along with all the rest of the GNU developers.

Why do you write “GNU/Linux” instead of “GNU Linux”? ([#whyslash](#))

Following the rules of English, in the construction “GNU Linux” the word “GNU” modifies “Linux”. This can mean either “GNU's version of Linux” or “Linux, which is a GNU package.” Neither of those meanings fits the situation at hand.

Linux is not a GNU package; that is, it wasn't developed under the GNU Project's aegis or contributed specifically to the GNU Project. Linus Torvalds wrote Linux independently, as his own project. So the “Linux, which is a GNU package” meaning is not right.

We're not talking about a distinct GNU version of Linux, the kernel. The free GNU/Linux distros do have a [separate version of Linux](#), since the “standard” version contains non-free firmware “blobs”. If this were part of the GNU Project, it could be considered “GNU Linux”; but we would not want to call it that, because it would be too confusing.

We're talking about a version of GNU, the operating system, distinguished by having Linux as the kernel. A slash fits the situation because it means “combination.” (Think of “Input/Output”.) This system is the combination of GNU and Linux; hence, “GNU/Linux”.

There are other ways to express “combination”. If you think that a plus-sign is clearer, please use that. In French, a hyphen is clear: “GNU-Linux”. In Spanish, we sometimes say “GNU con Linux”.

Why “GNU/Linux” rather than “Linux/GNU”? ([#whyorder](#))

It is right and proper to mention the principal contribution first. The GNU contribution to the system is not only bigger than Linux and prior to Linux, we actually started the whole activity.

However, if you prefer to call the system “Linux/GNU”, that is a lot better than what people usually do, which is to omit GNU entirely and make it seem that the whole system is Linux.

My distro is called “Foofoo Linux”; doesn't that show it's really Linux? ([#distribnames](#))

It means that the people who make the “Foofoo Linux” distro are repeating the common mistake. We appreciate that distributions like Debian, Dragora, Musix, Trisquel, and Venenux have adopted GNU/Linux as part of their official name, and we hope that if you are involved with a different distribution, you will encourage it to do the same.

My distro's official name is “Foofoo Linux”; isn't it wrong to call the distro anything but “Linux”? ([#distribnames1](#))

If it's allowed for them to change “GNU” to “Foofoo Linux”, it's allowed for you to change it back and call it “Foofoo GNU/Linux”. This is what you ought to do, because otherwise you'd be propagating their error and unfairness.

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? ([#companies](#))

It isn't a choice of one or the other—we ask companies and organizations and individuals to help spread the word about this. In fact, we have asked all three of those companies. Mandrake said it would use the term “GNU/Linux” some of the time, but IBM and Red Hat were unwilling to help. One executive said, “This is a pure commercial decision; we expect to make more money calling it ‘Linux’.” In other words, that company did not care what was right.

We can't make them do this right, but we're not the sort to give up just because the road isn't easy. You may not have as much influence at your disposal as IBM or Red Hat, but you can still help. Together we can change the situation to the point where companies will make more profit calling it “GNU/Linux”.

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. ([#reserve](#))

The widespread practice of adding non-free software to the GNU/Linux system is a major problem for our community. It teaches the users that non-free software is ok, and that using it is part of the spirit of “Linux”. Many “Linux” User Groups make it part of their mission to help users use non-free add-ons, and may even invite salesmen to come and make sales pitches for them. They adopt goals such as “helping the users” of GNU/Linux (including helping them use non-free applications and drivers), or making the system more popular even at the cost of freedom.

The question is how to try to change this.

Given that most of the community which uses GNU with Linux already does not realize that's what it is, for us to disown these adulterated versions, saying they are not really GNU, would not teach the users to value freedom more. They would not get the intended message. They would only respond they never thought these systems were GNU in the first place.

The way to lead these users to see a connection with freedom is exactly the opposite: to inform them that all these system versions *are* versions of GNU, that they all are based on a system that exists specifically for the sake of the users' freedom. With this understanding, they can start to recognize the distributions that include non-free software as perverted, adulterated versions of GNU, instead of thinking they are proper and appropriate “versions of Linux”.

It is very useful to start GNU/Linux User Groups, which call the system GNU/Linux and adopt the ideals of the GNU Project as a basis for their activities. If the Linux User Group in your area has the problems described above, we suggest you either campaign within the group to change its orientation (and name) or start a new group. The people who focus on the more superficial goals have a right to their views, but don't let them drag you along!

Why not make a GNU distribution of Linux (sic) and call that GNU/Linux? ([#gnudist](#))

All the “Linux” distributions are actually versions of the GNU system with Linux as the kernel. The purpose of the term “GNU/Linux” is to communicate this point. To develop one new distribution and call that alone “GNU/Linux” would obscure the point we want to make.

As for developing a distribution of GNU/Linux, we already did this once, when we funded the early development of Debian GNU/Linux. To do it again now does not seem useful; it would be a lot of work, and unless the new distribution had substantial practical advantages over other distributions, it would serve no purpose.

Instead we help the developers of 100% free GNU/Linux distributions, such as gNewSense and Ututo.

Why not just say “Linux is the GNU kernel” and release some existing version of GNU/Linux under the name “GNU”? ([#linuxgnu](#))

It might have been a good idea to adopt Linux as the GNU kernel back in 1992. If we had realized, then, how long it would take to get the GNU Hurd to work, we might have done that. (Alas, that is hindsight.)

If we were to take an existing version of GNU/Linux and relabel it as “GNU”, that would be somewhat like making a version of the GNU system and labeling it “Linux”. That wasn't right, and we don't want to act like that.

Did the GNU Project condemn and oppose use of Linux in the early days? ([#condemn](#))

We did not adopt Linux as our kernel, but we didn't condemn or oppose it. In 1993 we started discussing the arrangements to sponsor the development of Debian GNU/Linux. We also sought to cooperate with the people who were changing some GNU packages for use with Linux. We wanted to include their changes in the standard releases so that these GNU packages would work out-of-the-box in combination with Linux. But the changes were often ad-hoc and nonportable; they needed to be cleaned up for installation.

The people who had made the changes showed little interest in cooperating with us. One of them actually told us that he didn't care about working with the GNU Project because he was a “Linux user”. That came as a shock, because the people who ported GNU packages to other systems had generally wanted to work with us to get their changes installed. Yet these people, developing a system that was primarily based on GNU, were the first (and still practically the only) group that was unwilling to work with us.

It was this experience that first showed us that people were calling a version of the

GNU system “Linux”, and that this confusion was obstructing our work. Asking you to call the system “GNU/Linux” is our response to that problem, and to the other problems caused by the “Linux” misnomer.

Why did you wait so long before asking people to use the name GNU/Linux? ([#wait](#))

Actually we didn't. We began talking privately with developers and distributors about this in 1994, and made a more public campaign in 1996. We will continue for as long as it's necessary.

Should the GNU/*name* convention be applied to all programs that are GPL'ed? ([#allgpled](#))

We never refer to individual programs as “GNU/*name*”. When a program is a GNU package, we may call it “GNU*name*”.

GNU, the operating system, is made up of many different programs. Some of the programs in GNU were written as part of the GNU Project or specifically contributed to it; these are the GNU packages, and we often use “GNU” in their names.

It's up to the developers of a program to decide if they want to contribute it and make it a GNU package. If you have developed a program and you would like it to be a GNU package, please write to [<gnu@gnu.org>](mailto:gnu@gnu.org), so we can evaluate it and decide whether we want it.

It wouldn't be fair to put the name GNU on every individual program that is released under the GPL. If you write a program and release it under the GPL, that doesn't mean the GNU Project wrote it or that you wrote it for us. For instance, the kernel, Linux, is released under the GNU GPL, but Linus did not write it as part of the GNU Project—he did the work independently. If something is not a GNU package, the GNU Project can't take credit for it, and putting “GNU” in its name would be improper.

In contrast, we do deserve the overall credit for the GNU operating system as a whole, even though not for each and every program in it. The system exists as a system because of our determination and persistence, starting in 1984, many years before Linux was begun.

The operating system in which Linux became popular was basically the same as the GNU operating system. It was not entirely the same, because it had a different kernel, but it was mostly the same system. It was a variant of GNU. It was the GNU/Linux system.

Linux continues to be used primarily in derivatives of that system—in today's versions of the GNU/Linux system. What gives these systems their identity is GNU and Linux at the center of them, not particularly Linux alone.

Since much of GNU comes from Unix, shouldn't GNU give credit to Unix by using “Unix” in its name? ([#unix](#))

Actually, none of GNU comes from Unix. Unix was proprietary software (and still is), so using any of its code in GNU would have been illegal. This is not a coincidence; this is why we developed GNU: since you could not have freedom in using Unix, or any of the other operating systems of the day, we needed a free system to replace it. We could not copy programs, or even parts of them, from Unix; everything had to be written afresh.

No code in GNU comes from Unix, but GNU is a Unix-compatible system; therefore, many of the ideas and specifications of GNU do come from Unix. The name “GNU”, which stands for “GNU's Not Unix”, is a humorous way of giving credit to Unix for this, following a hacker tradition of recursive acronyms that started in the 70s.

The first such recursive acronym was TINT, “TINT Is Not TECO”. The author of TINT wrote another implementation of TECO (there were already many of them, for various systems), but instead of calling it by a dull name like “*somethingorother* TECO”, he thought of a clever amusing name. (That's what hacking means: [playful cleverness](#).)

Other hackers enjoyed that name so much that we imitated the approach. It became a tradition that, when you were writing from scratch a program that was similar to some existing program (let's imagine its name was “Klever”), you could give it a recursive acronym name, such as “MINK” for “MINK Is Not Klever.” In this same spirit we called our replacement for Unix “GNU's Not Unix”.

Historically, AT&T which developed Unix did not want anyone to give it credit by using “Unix” in the name of a similar system, not even in a system 99% copied from Unix. AT&T actually threatened to sue anyone giving AT&T credit in that way. This is why each of the various modified versions of Unix (all proprietary, like Unix) had a completely different name that didn't include “Unix”.

Should we say “GNU/BSD” too? ([#bsd](#))

We don't call the BSD systems (FreeBSD, etc.) “GNU/BSD” systems, because that term does not fit the history of the BSD systems.

The BSD system was developed by UC Berkeley as non-free software in the 80s, and became free in the early 90s. A free operating system that exists today is almost certainly either a variant of the GNU system, or a kind of BSD system.

People sometimes ask whether BSD too is a variant of GNU, as GNU/Linux is. It is not. The BSD developers were inspired to make their code free software by the example of the GNU Project, and explicit appeals from GNU activists helped convince them to start, but the code had little overlap with GNU.

BSD systems today use some GNU packages, just as the GNU system and its variants use some BSD programs; however, taken as wholes, they are two different systems that evolved separately. The BSD developers did not write a kernel and add it to the GNU system, so a name like GNU/BSD would not fit the situation.

The connection between GNU/Linux and GNU is much closer, and that's why the name “GNU/Linux” is appropriate for it.

There is a version of GNU which uses the kernel from NetBSD. Its developers call it “Debian GNU/NetBSD”, but “GNU/kernelofNetBSD” would be more accurate, since NetBSD is an entire system, not just the kernel. This is not a BSD system, since most of the system is the same as the GNU/Linux system.

If I install the GNU tools on Windows, does that mean I am running a GNU/Windows system? ([#othersys](#))

Not in the same sense that we mean by “GNU/Linux”. The tools of GNU are just a part of the GNU software, which is just a part of the GNU system, and underneath them you would still have another complete operating system which has no code in common with GNU. All in all, that's a very different situation from GNU/Linux.

Can't Linux be used without GNU? ([#justlinux](#))

Linux is used by itself, or with small other programs, in some appliances. These small software systems are a far cry from the GNU/Linux system. Users do not install them on PCs, for instance, and would find them rather disappointing. It is useful to say that these appliances run just Linux, to show how different those small platforms are from GNU/Linux.

Are there complete Linux systems [sic] without GNU? ([#linuxsyswithoutgnu](#))

There are complete systems that contain Linux and not GNU; Android is an example. But it is a mistake to call them “Linux” systems, just as it is a mistake to call GNU a “Linux” system.

Android is very different from the GNU/Linux system—because the two have very little code in common. In fact, the only thing they have in common is Linux.

If you call the whole GNU/Linux system “Linux”, you will find it necessary to say things like, “Android contains Linux, but it isn't Linux, because it doesn't have the usual Linux [sic] libraries and utilities [meaning the GNU system].”

Android contains just as much of Linux as GNU/Linux does. What it doesn't have is the GNU system. Android replaces that with Google software that works quite differently. What makes Android different from GNU/Linux is the absence of GNU.

Why not call the system “Linux” anyway, and strengthen Linus Torvalds' role as posterboy for our community? ([#helplinus](#))

Linus Torvalds is the “posterboy” (other people's choice of word, not ours) for his goals, not ours. His goal is to make the system more popular, and he believes its value to society lies merely in the practical advantages it offers: its power, reliability and easy availability. He has never advocated [freedom to cooperate](#) as an ethical principle, which is why the public does not connect the name “Linux” with that principle.

Linus publicly states his disagreement with the free software movement's ideals. He developed non-free software in his job for many years (and said so to a large audience at a “Linux”World show), and publicly invited fellow developers of Linux, the kernel, to use non-free software to work on it with him. He goes even further, and rebukes people who suggest that engineers and scientists should consider social consequences of our technical work—rejecting the lessons society learned from the development of the atom bomb.

There is nothing wrong with writing a free program for the motivations of learning and having fun; the kernel Linus wrote for those reasons was an important contribution to our community. But those motivations are not the reason why the complete free system, GNU/Linux, exists, and they won't secure our freedom in the future. The public needs to know this. Linus has the right to promote his views; however, people should be aware that the operating system in question stems from ideals of freedom, not from his views.

Isn't it wrong for us to label Linus Torvalds' work as GNU? ([#claimlinux](#))

It would be wrong, so we don't do that. Torvalds' work is Linux, the kernel; we are careful not to attribute that work to the GNU Project or label it as “GNU”. When we talk about the whole system, the name “GNU/Linux” gives him a share of the credit.

Does Linus Torvalds agree that Linux is just the kernel? ([#linusagreed](#))

He recognized this at the beginning. The earliest Linux release notes said, [“Most of the tools used with linux are GNU software and are under the GNU copyleft. These tools aren't in the distribution - ask me \(or GNU\) for more info”](#).

Why not finish the GNU Hurd kernel, release the GNU system as a whole, and forget the question of what to call GNU/Linux? ([#finishhurd](#))

We would like credit for the GNU operating system no matter which kernel is used with it.

Making the GNU Hurd work well enough to compete with Linux would be a big job,

and it's not clearly necessary. The only thing ethically wrong with Linux as a kernel is its inclusion of firmware “blobs”; the best fix for that problem is [developing free replacement for the blobs](#).

The battle is already lost—society has made its decision and we can't change it, so why even think about it? ([#lost](#))

This isn't a battle, it is a campaign of education. What to call the system is not a single decision, to be made at one moment by “society”: each person, each organization, can decide what name to use. You can't make others say “GNU/Linux”, but you can decide to call the system “GNU/Linux” yourself—and by doing so, you will help educate others.

Society has made its decision and we can't change it, so what good does it do if I say “GNU/Linux”? ([#whatgood](#))

This is not an all-or-nothing situation: correct and incorrect pictures are being spread more or less by various people. If you call the system “GNU/Linux”, you will help others learn the system's true history, origin, and reason for being. You can't correct the misnomer everywhere on your own, any more than we can, but you can help. If only a few hundred people see you use the term “GNU/Linux”, you will have educated a substantial number of people with very little work. And some of them will spread the correction to others.

Wouldn't it be better to call the system “Linux” and teach people its real origin with a ten-minute explanation? ([#explain](#))

If you help us by explaining to others in that way, we appreciate your effort, but that is not the best method. It is not as effective as calling the system “GNU/Linux”, and uses your time inefficiently.

It is ineffective because it may not sink in, and surely will not propagate. Some of the people who hear your explanation will pay attention, and they may learn a correct picture of the system's origin. But they are unlikely to repeat the explanation to others whenever they talk about the system. They will probably just call it “Linux”. Without particularly intending to, they will help spread the incorrect picture.

It is inefficient because it takes a lot more time. Saying and writing “GNU/Linux” will take you only a few seconds a day, not minutes, so you can afford to reach far more people that way. Distinguishing between Linux and GNU/Linux when you write and speak is by far the easiest way to help the GNU Project effectively.

Some people laugh at you when you ask them to call the system GNU/Linux. Why do you subject yourself to this treatment? ([#treatment](#))

Calling the system “Linux” tends to give people a mistaken picture of the system's history and reason for existence. People who laugh at our request probably have picked up that mistaken picture—they think our work was done by Linus, so they laugh when we ask for credit for it. If they knew the truth, they probably wouldn't laugh.

Why do we take the risk of making a request that sometimes leads people to ridicule us? Because often it has useful results that help the GNU Project. We will run the risk of undeserved abuse to achieve our goals.

If you see such an ironically unfair situation occurring, please don't sit idly by. Please teach the laughing people the real history. When they see why the request is justified, those who have any sense will stop laughing.

Some people condemn you when you ask them to call the system GNU/Linux. Don't you lose by alienating them? ([#alienate](#))

Not much. People who don't appreciate our role in developing the system are unlikely to make substantial efforts to help us. If they do work that advances our goals, such as releasing free software, it is probably for other unrelated reasons, not because we asked them. Meanwhile, by teaching others to attribute our work to someone else, they are undermining our ability to recruit the help of others.

It makes no sense to worry about alienating people who are already mostly uncooperative, and it is self-defeating to be deterred from correcting a major problem lest we anger the people who perpetuate it. Therefore, we will continue trying to correct the misnomer.

Whatever you contributed, is it legitimate to rename the operating system? ([#rename](#))

We are not renaming anything; we have been calling this system “GNU” ever since we announced it in 1983. The people who tried to rename it to “Linux” should not have done so.

Isn't it wrong to force people to call the system “GNU/Linux”? ([#force](#))

It would be wrong to force them, and we don't try. We call the system “GNU/Linux”, and we ask you to do it too.

Why not sue people who call the whole system “Linux”? ([#whynotsue](#))

There are no legal grounds to sue them, but since we believe in freedom of speech, we wouldn't want to do that anyway. We ask people to call the system “GNU/Linux” because that is the right thing to do.

Shouldn't you put something in the GNU GPL to require people to call the system "GNU"? ([#require](#))

The purpose of the GNU GPL is to protect the users' freedom from those who would make proprietary versions of free software. While it is true that those who call the system "Linux" often do things that limit the users' freedom, such as bundling non-free software with the GNU/Linux system or even developing non-free software for such use, the mere act of calling the system "Linux" does not, in itself, deny users their freedom. It seems improper to make the GPL restrict what name people can use for the system.

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? ([#BSDlicense](#))

It would be hypocritical to make the name GNU/Linux a license requirement, and we don't. We only ask you to give us the credit we deserve.

Please note that there are at least [two different BSD licenses](#). For clarity's sake, please don't use the term "BSD license" without specifying which one.

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? ([#deserve](#))

The question presupposes a rather controversial general ethical premise: that if people do not force you to treat them fairly, you are entitled to take advantage of them as much as you like. In other words, it assumes that might makes right.

We hope you disagree with that premise just as we do.

Wouldn't you be better off not contradicting what so many people believe? ([#contradict](#))

We don't think we should go along with large numbers of people because they have been misled. We hope you too will decide that truth is important.

We could never have developed a free operating system without first denying the belief, held by most people, that proprietary software was legitimate and acceptable.

Since many people call it "Linux", doesn't that make it right? ([#somanyright](#))

We don't think that the popularity of an error makes it the truth.

Isn't it better to call the system by the name most users already know? ([#knownname](#))

Users are not incapable of learning. Since “GNU/Linux” includes “Linux”, they will recognize what you're talking about. If you add “(often erroneously referred to as ‘Linux’)” once in a while, they will all understand.

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? ([#winning](#))

To care only about what's convenient or who's winning is an amoral approach to life. Non-free software is an example of that amoral approach and thrives on it. Thus, in the long run it would be self-defeating for us to adopt that approach. We will continue talking in terms of right and wrong.

We hope that you are one of those for whom right and wrong do matter.