

Debian - The Universal Operating System?

Do we provide what users need? Do users get what they really want?

Andreas Tille

22. June 2007

Abstract

Debian GNU/Linux is a Linux distribution that is heavily used as a base for several derivatives - there is no other distribution that has such a large number of derived distributions. This is one reason to make us proud because we are considered solid and rock stable. But the question remains: Why do these people not use plain Debian instead of building their own distribution?

Debian wants to be “The Universal Operating System” - at least the title of the homepage says so. What means universal? One of WordNet answers is “*adapted to various purposes, sizes, forms, operations*”. If we agree to this definition we have to experience: Debian is not really adapted to various purposes but it is perfectly adaptable.

It is not really bad if Debian is adaptable to various purposes and the fact that Debian and even its derivatives are used as a base for adaptations makes us proud. It is a good sign that Debian is an unspecialised base system which is not really modified for a particular purpose or function which makes it attractive for adaptation to any purpose because it is known to be very robust and stable. So in short: We do a quite good job but it does not really qualify for the attribute “*universal*”.

The first question is: Do we really want to be universal? If we answer this with yes, the next question is: Can we as a group of more or less independent maintainers afford this? My answer to this question is yes and the method to do this are Custom Debian Distributions. Currently we leave most of the work that could be perfectly done inside Debian to outsiders. The fact that outsiders constantly are working on enhancing Debian in a certain direction means: There are things missing in Debian and there are things our users are needing that we do not provide.

Over several years I was constantly seeking for reasons, why we seem to fail the task of making adaptations and derivatives superfluous. What makes people keen on changing Debian to fit their needs. The answer to this question goes in two directions:

1. The current structure of Debian does not allow the creation of an universal, multipurpose system.
2. It is impossible to fit all needs - so why not at least making it brain dead easy to adapt Debian.

In the talk I try to discuss issues that might help us to change Debian from a geekish system to something people who consider themselves “only simple user” (whatever this might be) *want* to use instead something they are afraid of. Even if we members of the conference know that there is no reason to be afraid of Debian

...

This is not really a technical paper but it concerns considerations about organisational structure, ways to widen the view from single maintainer - single package (1:1) or maintaining group - single package (n:1) development model to a customisation group - Custom Debian Distribution (n:m) development model to offer our users who need m packages interacting together the optimal system.

1 Different viewpoints of users

1.1 Poor users view

Debian users can be divided into several user types and there are also types of potential users who could be really happy with Debian in principle but either do not know this or have no real chance to learn about the fact that Debian is great because they run into certain trouble before it comes to real usage.

Admittedly there are users who should be referred as “poor users” as a shorthand. For instance the friend of my son. If he would decide to install Debian on similar hardware as my son he would be in vain. It took me several attempts to get the r1000 network adapter up and running (inclusive downloading, compiling and fixing the driver, by asking for help on lists and finally got a working kernel). Moreover the 3D acceleration with the ATI card needed several trials with different kernels and took me finally two days until everything was up and running. So the short answer to the question why the friend of my son does not install Debian GNU/Linux is just because his father is no Debian developer or at least a Linux expert.

I admit, buying by chance a network adapter that is not supported (even if there is a free driver is currently quite seldom and you have to be a very poor user to face this situation. Having some hardware with non-free drivers is not that seldom. The usual advise is to check before buying hardware – but honestly, those users who have no idea how to circumvent a problem with drivers that do not exist in Debian are not aware of the fact that checking before is a good idea. They expect a system to run from scratch and they tend to not accept our reasoning that in most cases we can not do anything to solve this Problem.

1.2 Lucky users view

The lucky user is able to proceed to the whole installation process flawlessly either because he has verified whether his hardware is supported or just was lucky enough to grab fully supported hardware by chance. They reach the point where they are asked to choose a certain task via `tasksel`.

While these users are lucky that their hardware was detected nicely and Debian will probably run flawlessly on their computer they now need a certain knowledge about the tasks they want to solve with this computer. If the user has no idea about the

huge amount of software that comes with Debian he hopefully has an idea about clever strategies how to find the software that is helpful for his day to day work because `tasksel` is a quite raw and there will probably be no user that has to continue the package selection with other tools like `aptitude`, `synaptic` or something like that.

1.3 Upstream developers view

Sometimes upstream developers of a piece of software are using Debian GNU/Linux and sometimes they just use other systems. Also in the latter case they might be called Debian users in a wider sense. If Debian is distributing their software they just gain some profit from it because Debian is serving as vehicle for their code to the end user. This might be a very welcome way to advertise a software if it is distributed via Debian.

Moreover Debian might serve as a quite good quality assurance instance for a project – it is just a good sign if the software installs and runs flawlessly. While commercial developers rely on LSB standard developers of small free projects either have no time or interest in reading all the documents and running certain tests – if inclusion into Debian works a certain level of quality is ensured.

A further point for Debian is the fact that the maintainer is asked for writing some documentation like a man page. This is just being nice to the upstream author.

Finally Debian packages will be auto builded for different architectures which often uncovers porting problems. Once a program is packaged for Debian the author has evidence that the result of his work at least compiles on several hardware architectures.

1.4 Debian Developer view

In the Debian world the maintainer is some kind of *missing link* between upstream developer and user. He is doing the grunt work of obtaining the sources from upstream, building and installing. This process is often boring and boring things are just error prone. So this is the kind of work users can get rid of thanks to the Debian maintainers.

Every Debian developer picks such pieces of Free Software as packaging target that are interesting for his day to day work or his hobbies. In both cases the driving force to maintain a certain software is the fact that it is anyhow useful to solve a certain problem the maintainer has and where he is competent in - at least to a certain level.

This kind of motivation is comparable to the principle that Free Software development is based on: Find a solution for your own problem and share the code to get help of other people who try to solve the same problem. Debian developers just try to solve the problem to develop an operating system instead of writing a certain software for a specific problem. This operating system should meet the interest of as wide as possible user base. The independence of commercial interest of the producer of this system guarantees that it is even possible to support niche groups, that have no large customer base. If a commercial distributor would like to support small specialised user groups this would conflict with the interest to recruit a large customer group. So Debian is able to support those users who might need special applications natively which is not the case by other main distributors.

This feature might lead to Debian developers who act like hunters and collectors of Free Software applications – which is not really a bad thing but to make users really happy it needs more than just a huge collection of any application for certain purposes. It needs some design of a complex and comprehensive system that the user enables to find and use all these applications.

1.5 Derivers view

Debian GNU/Linux has a lot of derivatives for several reasons:

1. Commercial distributions
2. Live CDs
3. “Enhanced” distributions (for instance regarding usability)
4. Specialised distributions

Debian has proven to serve nicely as workpiece that can be modelled to something else. The deriver considers Debian as nice in principle but it does not really fit his needs completely. According to the derivers opinion this is a good reason to take Debian and make something else from it. The quality of Debian will propagate into the new product and thus Debian is considered as a good starting point for a derivative.

2 Universal?

2.1 Nitpicking the term universal

```
Asking WordNet about similarities of the word “universal” we get: $ wordnet
universal -synsa
Similarity of adj universal
3 senses of universal
```

Sense 1

cosmopolitan, ecumenical, oecumenical, general, universal, worldwide, world-wide
=> comprehensive (vs. noncomprehensive)

Sense 2

universal
=> general (vs. specific)

Sense 3

universal
=> adaptable (vs. unadaptable)

It makes no sense to start a nitpicking about word meanings here. The attributes above partially fit very good to `Debian` and others don't. The question is: What do we really want to provide to our users?

For instance *adaptable* is a great feature and we certainly will not change it. But can we go one step further and could provide a system that is not only *adaptable* but even *adapted* to a certain range of tasks our users want to do? Does the feature *general* really exclude that we also can serve very *specific* needs? Does *comprehensive* not include that `Debian` should be prepared for very special needs?

One more query to WordNet says:

```
$ wordnet universal -over
```

```
...
```

```
The adj universal has 3 senses (first 1 from tagged texts)
```

1. cosmopolitan, ecumenical, oecumenical, general, universal, worldwide, world-wide – (of worldwide scope or applicability; "an issue of cosmopolitan import"; "the shrewdest political and ecumenical comment of our time"- Christopher Morley; "universal experience")
2. universal – (applicable to or common to all members of a group or set; "the play opened to universal acclaim"; "rap enjoys universal appeal among teenage boys")
3. universal – (*adapted to various purposes, sizes, forms, operations*; "universal wrench", "universal chuck"; "universal screwdriver")

While item 1 does perfectly fit as a description of `Debian` item 2 is a little bit weak. Looking at the viewpoints of potential `Debian` users we have one group that definitely is not covered: `Debian` is not applicable for the group of derivers. That is kind of a trivial conclusion if we say that derivers are now `Debian` users (per definition). But is this really something that we want? To answer this question we will go below more into the detail of the reasons for deriving `Debian`.

Finally item 3 depends from the viewpoint how somebody looks at a computer as a tool to solve the daily work. Surely `Debian` is an operating system that can be used for any task – there is no doubt about that from the developers view. But is it enough from a user point of view. Is `Debian` a system that is ready to run for every task after pressing a view keys?

For instance look at a teacher: The main focus of a teacher is not to run a specific operating system. He rather wants to have tools to

- handle student accounts
- enable students to run educational software
- teach students basic skills in information science
- ...

A teacher will choose the tool or rather the tool-set that serves the features above while being stable, reliable and keeps him free of headaches because students try to demonstrate their ability to destroy things. So the motivation for using `Debian` in the flavour

of Debian-Edu is not *because* he wants a cool operating system but because he wants the perfect tools for his day to day work ready to run after pressing a few keys.

So looking from the perspective of a teacher Debian provides an universal tool because he is able to do everything he wants with a minimum effort. If our user is a scientist he can find a lot of very interesting applications packaged in Debian. So in principle Debian could provided some kind of scientific workbench a feature that is rare amongst other distributions. But this workbench is kind of served in pieces and the scientist has to build it up before he can start working. So what Debian really serves is comparable to a discounter that leaves the work to assemble the workbench to the customer instead of delivering a full featured, ready to run workbench right into the workshop of the customer. At least if you follow the [Debian-Science mailing list](#) you have the impression that there are solutions for nearly everybody (or at least people working on packaging the missing ones) but there is a lack of concept to guide scientific users through the Debian universe of packages to let them find the things they need easily.

The situation is slightly better for biologists, because the [Debian-Med project](#) is covering a large amount of biological applications and more importantly has a way to install these with one single command or mouse click. But it would be way to much to compare the state of Debian-Med with Debian-Edu which is perfectly usable in real live.

The best operating system is a system that just works and makes no trouble - most users do not want to notice their operating system (and many of them do not even know that something like an operating system does exists).

2.2 Limits to Growth

While we proudly pronounce that Debian is the largest distribution we should have a look whether Debian scales well in all directions we are growing. What are the directions in which the Debian universe is growing?

Number of people involved in development which are not only the official developers but also sponsees who are mostly sitting anywhere in the new maintainer queue

Number of packages is constantly growing and the structure inside this growing number is quite flat. Structuring elements are on one hand the `Priority` control field and on the other hand the `Section` control field. Both are not really effective any more to give a good overview. There is a bright light at the horizon that is called `DebTags` that has a high potential to give a fine grained structure but currently it has kind of a Debian insider status and is not really known to most of our users.

Number of architectures is the dimension with the lowest rate of growth but every additional architecture might add quite complex problems.

Number of bugs is increasing because there is no balance between the number of closed and opened bugs. Moreover the complexity how to handle bugs is growing (tagging for different versions / suite).

Number of users is also constantly growing – which is just a good sign and a good reason to be proud of. On the other hand this increased number is also a task: Make all these users with very different needs happy – or at least try to do so.

Number of derivatives is also growing and mostly not even recognised. The reasons and whether it makes sense is discussed below

Number of flame wars is the not so cool coordinate in the Debian universe and it would be really good if we would be able to stop this growth. It just takes time from developers and sometimes it takes developers from Debian.

Number of attendees of DebConf which is just nice. ☺

It is a well known fact that a system that grows constantly changes over time into a system with new features (*“Change of quantity into quality; Hegel”*). Those Debian developers who joined in the middle of nineties last century might remember that their hobby horse was some nice system that was developed and used by some interested geeks who were clever enough to work around certain problems and Debian was famous to be “administrators choice” but “not usery friendly”. Well sometimes you meet people who never tried Debian and keep this precondition alive – but Debian has changed. At least since Sarge Debian reached a state that everybody (except “poor users”) is able to get it up and running quite flawlessly.

But when happened this change? *“To determine at the right moment the critical point where quantity changes into quality is one of the most important and difficult tasks in all the spheres of knowledge.”* (Trotzki) Well, it is useless to find the exact point in time once we just passed it – obviously it was a positive change looking a the features of Debian. The main point is that we should actively analyse the situation and find out whether there is something to do for the future to be prepared for the next changes that will come.

A nice description of the current organisation of Debian was given by Manoj Srivastava (8 Oct 2003):

For the most part, Debian is a Bazaar of Cathedrals; with a few procedures in place to override the low level cathedral in exceptional situations. Each developer has, within reason, full control over his packages, modulo following Debian Technical policy, thus creating the low level cathedral. The Technical committee, and the General Resolution Protocol offer means of overriding developer decisions about their own packages.

This model seems to be quite effective but it establishes a 1:1 relation between maintainer and package. The fact that more complex software projects needed more man power led to the establishment of packaging groups which also has a better “run-over-by-bus” safety. Most of these groups work effectively with a n:1 maintainer to package relation. To be correct in many cases it is not just a single package this group of maintainers is caring for. In most cases it is a set of strongly connected packages for instance the X strike force deals with the software that is issued by X.Org. This group maintenance is a reliable and practical way to deal with complex packaging tasks and is based on a technical level to organise the packaging. It is not really focussed on complete suites covering all tasks a user wants to solve with his computer.

But there is a solution inside `Debian` that offers complete suites for defined groups of users: Custom Debian Distributions (CDD). The idea is to establish a n:m maintainer to package relation. This means a group of maintainers is busy building a `Debian` internal subsystem for special use cases. This approach solves different problems

Number of people As it was demonstrated above the number of `Debian` developers is constantly growing. While this is a nice thing in general it makes it harder to focus many people onto a common goal. So picking a special field and unifying those developers with a special interest in this field builds a task force of manageable size.

Number of packages While the number of packages in `Debian` is constantly growing more or less according to the principle: if a maintainer has time and interest to care about a certain piece of Free Software it will be included. In a CDD there is a certain road map about wanted and missing applications for the target users purpose which is the cause for some kind of coordinated growing of the number of packages in a CDD.

Number of bugs The existence of certain CDDs do not really reduce the number of bugs in `Debian` itself but the people working on the CDD are keen on keeping the number of bugs that directly or indirectly has an influence on the CDD. That means that not only a single package maintainer is looking at the bugs but a larger group with a common interest.

Number of users As user centric projects CDDs just address specific user groups it is easier to concentrate on the more or less clearly defined needs of these users and thus in fact the number of users is reduced to a closer circle.

Number of derivatives The main goal of CDD and derivatives of `Debian` are often quite similar. Derivatives are often started by people who try to do something in some respect differently than main `Debian` is doing. The reasons to derive from `Debian` might be very different and will be discussed later but if a CDD would fit the scope of a derivative perfectly at least the technical reason to do something else than original `Debian` would vanish. If there are non-technical reasons at least the workload of the deriver can be reduced.

Release date The Etch release was the first one that has broken the rule of constantly growing time spans between release dates. Moreover this last point in this list is a little bit weak because there are just random ideas about separate releases of certain CDDs even if a CDD is completely integrated into `Debian`. The rationale behind this is that for certain applications certain fixed release dates are needed and thus the CDD approach might lead in the far future to releases of these parts of `Debian` separately from main `Debian`. But there is not even an idea how to implement this practically and thus this last point is rather a wild guess than a solution for the problem of long time spans in between `Debian` releases.

3 Deriving versus staying Debian internal

According to [Distrowatch](#) Debian is the distribution that has the most derivatives namely 129. It is followed by Fedora that has less than half this number (63) and other distributions are listed with less than 30 derivatives. It is not really bad if Debian is adaptable to various purposes and the fact that Debian and even its derivatives are used as a base for adaptations makes us proud.

Besides having a valid reason to be proud we should not ignore the fact that if somebody takes over the workload to start a derivative he must have some reason. In other words: There are a lot people out there who do not like Debian as it is – if they would they could save their time and take Debian as it is. So what might be the reasons for deriving?

3.1 The poor, diligent deriver

Sometimes people have some misconceptions about Debian. It just happens that people work happily with Debian while ignoring ways of feedback like `reportbug`, mailing lists or IRC channels completely. This kind of users tend to ignore that Debian is keen on getting user response to make a better system that would also fit their special needs. They just regard Debian as one-way street as they know it from proprietary systems where user response is something that just causes trouble.

Some other users might also have learned from bad experience that a maintainer has not behaved nicely and ignored bug reports or even worse responded unfriendly. Such cases just happen – well also Debian maintainers are human beings.

Those kind of users might think that they could easily reach their goal by downloading (parts of) the Debian archive, change the bits they need to reach their goal, build installer disks etc. While this is perfectly OK from a licensing point of view – it is just Free Software. The question is whether the effort to keep the system up to date, following security and other bug fixes etc. will not be too high to keep the high quality standards of plain Debian. If the derived project is able to cope with this issue it probably needs so much man power that the question remains, whether the effort, to make the necessary changes inside plain Debian.

3.2 The impatient, desperate deriver

As stated above Debian reached a size that in some aspects does not allow some flexible changes and that's why an impatient user might face show stoppers he can not bear with or can not wait for. OK, what to do if there is a time line the user has to reach with his solution and it becomes clear that the change that is needed for his task will not be realised timely. He desperately decides not to wait until Debian has implemented what he wants and just builds a “fixed for my purpose Debian”.

Debian might be a little bit more helpful to this kind of people in teaching one key feature of the Debian organisation: It is a so called [DoOcracy](#), which means in short the doer decides what becomes done. This principle allows people to contribute to Debian if they have learned some rules which the first ones are: Be sustained and patient. Show some code that is able to convince people.

If the deriver is really impatient he should make sure that he is keeping the diff against plain `Debian` remains small. This would ensure that once the reason for the need of deriving might have vanished switching back to original `Debian` is easily possible.

3.3 The authority employed deriver

Sometimes authorities decide to apply Linux on their computers. It seems to be a tendency that in this cases an own distribution is developed ([LiMux](#), [Wienux](#), [LinEx](#), [Lliurex](#), ...). Its probably a feature of authorities that they are convinced that they need something else than any other authority. An alternative way would be to start working in common on the [Debian-eGov](#). This CDD was started in a diploma work but is stalled now. It would make real sense to find the common things in all these authority related distributions and try to implement a common base in `Debian-eGov` to use this as source for a derivative. This enable to join forces and to reduce the workload to those things that are really special.

The case is similar in official sponsored projects. For instance the [DeMuDi](#) distribution which was funded by the European Union. These people had to get their own product out but tried to contribute their work back to `Debian`.

3.4 The lucky live CD creator

Live CDs are a cool thing for some reasons like demonstrating / showing off your nice product to people who do not run Linux as preferred system, testing your hardware, boot your favourite OS on any random computer, run a kiosk system, run a Linux course in a computer lab with computers featuring other operating system etc. The “mother of these” live CDs was probably the famous Knoppix. Many people regarded Knoppix as so extraordinary cool that they started deriving from Knoppix for several purposes. Well, deriving from a derivative of `Debian` is not a bad idea - the problem is to keep this system up to date. Especially the read-only feature of a CD makes distribution of changes harder than just a simple `apt-get` call.

The lesson that `Debian` has learned is that there are users out there who are really keen on producing live CDs and the most clever way would be to just use the creation of a live CD a feature of `Debian`. Try `live-helper` and have fun!

There are several projects outside `Debian` that tried in the past and are busy trying to find a general solution for the problem: How to obtain your own live CD ISO image. The message to all people who try to create their own live CD distribution is: Just look what is inside native `Debian` and see whether you can save a lot of time by the available tools.

3.5 The clever, commercial deriver

The fact that `Debian` is free allows to take it and turn it into a product that can be sold to customers. Several companies like Linspire, Mephis, Progeny, Ubuntu, Xandros and others are doing so. The idea is to add some value and sell some service to customers because you can not buy any service contract with `Debian` as an organisation. The

only service you get from `Debian` is help over the usual channels like mailing lists or IRC. While this support works really good it is completely volunteer based and enterprises need just some 24/7 contract or something like that.

So building a commercial distribution that is based on `Debian` makes perfectly sense because this also opens some opportunities that `Debian` does not have like for instance adding non-free software like drivers for certain hardware that can not be supported to full extend in plain `Debian`. So one purpose of these commercial derivatives might be to help the poor users mentioned in the beginning of this paper.

So if the derivative is kind of a re-branding of `Debian` that adds value that can not be added by `Debian` for licensing reasons (non-free drivers) or its pure volunteer organisation (24/7 support) it is perfectly reasonable. But the company that runs this business is well advised to keep the diff to plain `Debian` small for the same reasons as above: It just saves man power that can be spend to enhance the source of their business: Give back to `Debian`.

Believe it or not `Debian` tries to make the work of this deriviers as easy as possible because developers are constantly busy to make the best operation system they can afford. If you just can take the best that 1000+x people are able to produce and make a product from it a high quality product can be obtained for less. On the other hand `Debian` tries to make it hard to compete - for the same reason as above.

3.5.1 Derivers lessons to learn

Summarising this you can find the simple formula

$$\Delta(\textit{Debian} - \textit{Derivative}) < \varepsilon \forall t$$

which means: Keep the diff *always* small.

The second idea is: We are on your side. Talk to us.

4 Supporting very special applications

There are several reasons for and against adding special applications to the `Debian` pool. One drawback is penalty is that `Debian` becomes larger and larger regarding the number of packages and the question raises: How many packages are good for `Debian`? Currently there is no reasonable answer to this question and the rule for adding packages is currently: As long as one person is interested enough in a software to maintain the package it will be included.

From the upstream maintainers point of view it is a very interesting method to bring special applications under quality control. It often happens that `Debian` maintainers add reasonable patches and `Debian` users might file interesting bug reports that lead to further enhancements of the software.

As mentioned above `Debian` can serve as a vehicle for upstream software because `Debian` users just stumble upon some programs they did not know before when they are installing `Debian`. So integrating software into `Debian` just increases the user base of the software.

This turns in to the question: How many packages are good for our users? If our users on one hand learn about interesting applications and on the other hand are enabled

to install these applications flawlessly ready to run this is exactly what we want.

Related to the packaging of special applications is the question about packaging special data. Some applications need large data sets to work reasonably and it is not really an option to bundle all these into normal packages because this would drive mirror maintainers crazy in terms of bandwidth and disk space. There would be other solutions to discuss like providing a separate data archive which seems not to be happen according to [#38902 wont-fix](#) or alternatively find some common tool that downloads data according to certain rules.

So the basic goal of Custom Debian Distributions is to enable the user to focus on the packages that are really needed for his day to day work leading him friendly through the jungle of Debian's > 15000 packages. A user that is working in a certain field is only interested in a defined *subset* of packages and the CDD that is concerned about this field tries to prepare the computer optimally to install this subject with adapted configuration and easily accessible applications. So CDDs are taking care of groups specialised users turning Debian into a useful tool adapted to their requirements for day to day work and making it to the distribution of choice for their use cases. It should enable and easy installation and automatically configuration whenever possible to make the needed work to fit the intended purpose as small as possible

While Debian as a distribution stays a general collection of Free Software ready to install it supports specialists as well which is kind of unique in the distribution market. By attracting specialists Debian might become more attractive for a larger user base especially those users that are basically left alone by other distributions. So the basic idea of Custom Debian Distributions is

Do not make a separate distribution but make
Debian fit for special purpose instead.

5 Summary

Custom Debian Distributions might solve future structural problems of Debian while fitting better user interests. If done the right ay it makes Debian stronge and is sometimes refered to as: *“The last, final step towards Total World Domination.”*

From a Debian developers point of view we are really *universal*. From a random users point of view we are not – but we are perfectly able to reach this goal.