

5 OSS 事例： Debian

鵜飼 文敏

ukai@debian.or.jp

Debian Project /

特定非営利活動法人フリーソフトウェアイニシアティブ

Debian とは？

Debian は、1993 年 8 月に Linux と GNU の精神に則りオープンに開発される新しい Linux ディストリビューションとして、Ian Murdock が開発を始めたものである。それまでのディストリビューションとは異なり、特定の個人や企業によって開発をすすめるのではなく、それぞれのソフトウェアに詳しい人が開発に参加できるようにして、それらを積み上げていくことでディストリビューションをつくっていくことにしたのだ。そのような構造を持つ Debian には世界中の多くのフリーソフトウェア開発者、オープンソースソフトウェア開発者が参加するようになり、現在では最も巨大なディストリビューションとなっている。商業ディストリビューションに劣らない品質のよさと多くのユーザの支持を得るディストリビューションになった。サポートしているアーキテクチャも非常に多く、最近では Linux 以外のカーネルへの対応も行われつつある。また、Debian を元にした派生ディストリビューションと呼ばれるものも数多く作られており、たとえば Ubuntu や Knoppix などよくできたディストリビューションとして多くのユーザを得るようになってきている。

Debian における開発のすすめかた

□ パッケージによる分散開発

Linux の初期の頃は、特定の個人もしくは企業が、インターネット上で配布されている Linux で利用できるフリーソフトウェアを集めて使えるようにした「ディストリビューション」を作成・配布することで Linux の普及がすすんでいた。しかしながら、すべてのフリーソフトウェアに詳しい人はいないため、特定の個人でディストリビューションを作成しようとする、どうしても品質に劣る部分が出てしまいがちであった。企業が作成する場合は、その企業の意向に左右されてしまい、フリーソフトウェアで最も重要な「自由である」という点に問題が出てきてしまうことが懸念されていた。

そこで Debian では、ある分野の専門的知識を持つ人が、その分野を含む個々の構成要素である「パッケージ」を開発したり保守したりすることができるようにすることで、それらの問題を解決しようとした。その分野に詳しい人自身が開発および保守ができるようにすることで、それぞれの部分の品質が高くなることが期待できるので、全体として高い品質のソフトウェアの集合としてのディストリビューションを生み出すことを狙ったのである。

□ ポリシーによる暗黙の了解の明文化

パッケージの集合をシステムとして使うためには、単にパッケージを集めるだけではだめで、なんらかの方法でパッケージ間の関係をうまく調整する必要がある。パッケージ間には依存関係や競合関係などさまざまな関係が存在するからである。このような関係を調整するために開発されたのが dpkg と呼ばれるパッケージ管理システムである。パッケージの数が増えてくるにつれ dpkg では管理が複雑になってきたのでさらに高度なパッケージ管理システムとして apt が開発された。パッケージ管理システムを有効に使えるようにするためには個々のパッケージはそのパッケージ管理システムに対応して作成される必要がある。またパッケージを開発・作成するときには、他のパッケージがどのように作られているのかを知る必要があるが、パッケージの数が増えてくると他のパッケージをすべて調べることは不可能になってくる。そのような場合に必要になるのが、パッケージを作る際のガイドラインとして定められている Debian Policy Manual である。Debian Policy Manual にはパッケージが従うべき指針が書かれており、システム全体としてどのようになるかといった合意のようなもの

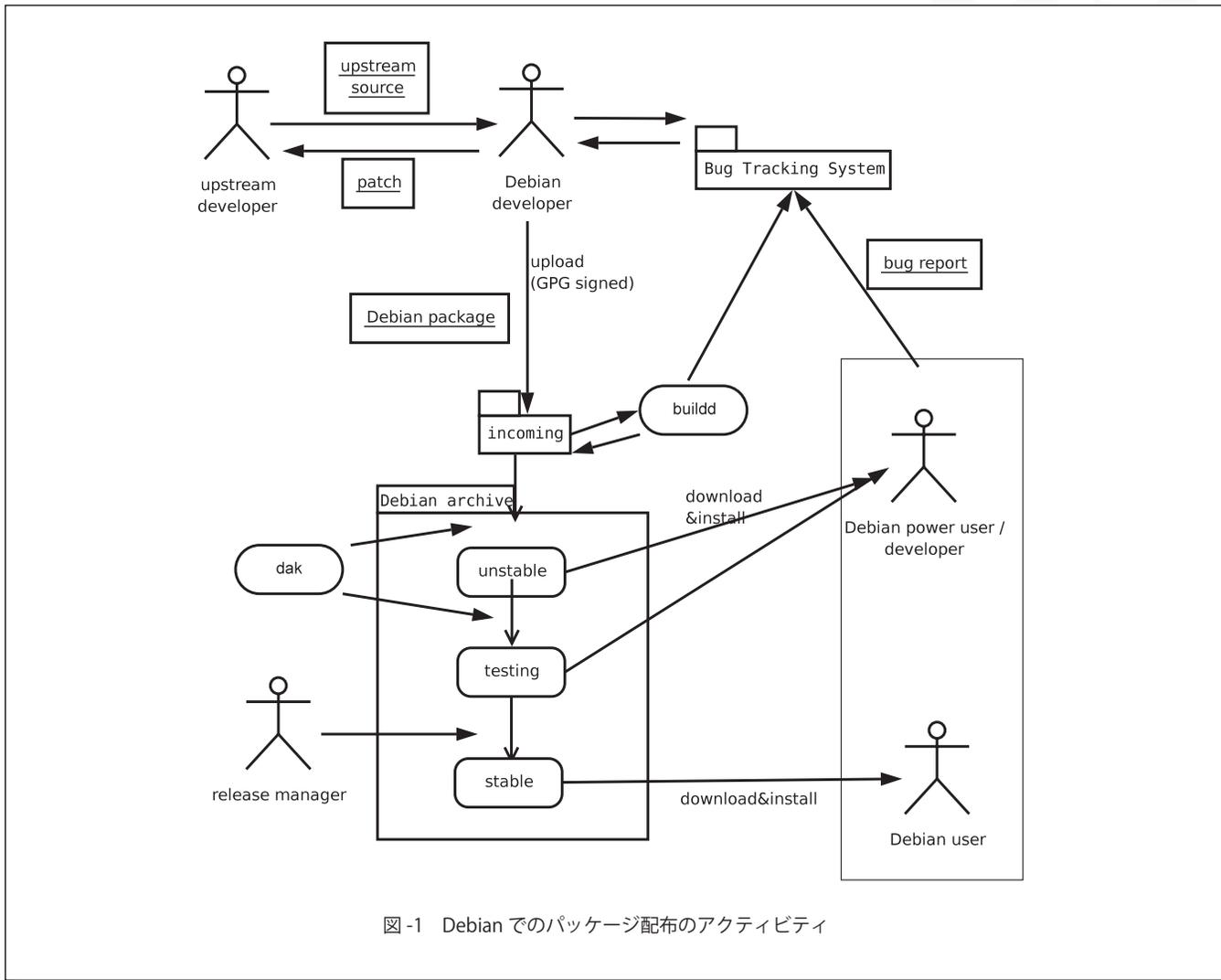


図-1 Debian でのパッケージ配布のアクティビティ

が文書化されていると言える。パッケージ管理システムと Policy Manual が補完しあって、パッケージの数が膨大になっても全体として整合性のとれたシステムが構築できるようになっている。

Debian での開発は、基本的にはソフトウェアのパッケージを Debian に向けて作成することである。Debian のパッケージシステムや Policy Manual に合うように修正を加えていくことになる。Policy Manual などあまり細かい点については記述されていない場合もあるので、そのようなところは他のパッケージ開発者と議論し調整する場合もある。場合によっては小さなジャンルで Policy を策定し、それに関係するパッケージはその Policy に従うようにしているものもある。

品質管理とリリース

Debian の unstable とよばれる版は、開発者がパッケージを更新したものを日々反映したものになっており最新のパッケージが利用できるようになっている(図-1)。そのようなパッケージはやはりバグが存在す

る可能性も高い。バグを発見した場合は Bug Tracking System に報告するようになっており、過去に報告されたバグやその経緯がすべて記録されている。バグ報告の中には、動作しないといった問題から要望までいろいろあり、バグ報告への対処がパッケージ開発者の最も重要な作業の1つである。バグ報告しないことには開発者に問題の存在が伝わらない。開発者が対処するために必要な情報をきちんとバグ報告に含めることが重要である。場合によっては元々のソフトウェアで修正した方がいい場合もある。そのような修正は元々のソフトウェアにフィードバックした方が継続したメンテナンスを考えると楽なことが多い。バグに対処しはじめるとバグ報告が増えるという現象もある。開発者がアクティブであるほうがバグの報告がされやすいという傾向があるように思われる。

Debian では、サポートしているアーキテクチャすべてでパッケージが作れないと、そのパッケージは正式リリースに含めないことになっている。つまり、パッケージにするソフトウェアにかなりの移植性が要求されてい

ることになる。開発者がパッケージをアップロードすると、そのソースパッケージを元に各アーキテクチャ用のパッケージを作成する buildd というインフラストラクチャが構築・運用されており、移植性の問題などをとらえやすくしている。

パッケージの更新、バグ報告、その修正というサイクルを毎日のように繰り返していくことでパッケージの品質を高くしていくことができる。このような開発を経て十分に安定してきた頃に stable として Debian のリリースが行われる。stable に含まれているパッケージは十分なテストが行われていることが期待できるために高い安定性を提供することができている。

Debian とオープンソース

Debian はフリーソフトウェアだけからなるディストリビューションであるが、何をもち「フリー」とするかについては、暗黙の了解はあったが明確な線引きといったものは存在しなかった。そこでプロジェクト内での議論を経て、何を「フリー」と見なすかについて「Debian Free Software Guidelines」という文書にまとめあげた。このガイドラインにそったものを Debian は「フリー」なソフトウェアとして扱うように決めたのである。

後に、この「Debian Free Software Guidelines」の Debian 固有の条項をより一般的になるように変更を加えたものが「Open Source Definition」となった。つまり、Debian によって定めた「フリー」と見なすものを、「オープンソース」と呼ぶようになったのである。オープンソースを一般に広めるきっかけになったのは Netscape 社によるソース公開であるが、オープンソースという概念に明確に定義を与えたのは Debian だったのである。

Debian をよりよくしてみたい人へ

Debian 開発者になるのが最も直接的に Debian を改善していく作業をすることができるが、Debian 開発者

になるためには長時間のプロセスが必要になっている。Debian 開発者は Debian の一部としてパッケージを作る責任を負っているので、Debian の理念や手続きをちゃんと理解しているかどうかを確認するプロセスが決められている。Debian 開発者にならなくてもできる活動がたくさんあるのでそれらを継続していくことが重要である。それらを継続してできないようでは、Debian 開発者になっても継続して活動できないと思われるため、そのような人は Debian 開発者にふさわしくないとされる。Debian のようなプロジェクトでは継続して活動すること、他の開発者と協調して作業をすすめていくことができることなどがメンバの資質として求められている。

Debian に含むことができるようなすばらしいフリーソフトウェアを開発することも Debian をよりよくしていくことの手助けになる。その場合ライセンスの選択には注意が必要だ。独自ライセンスよりも既存のライセンスのどれかを適用するほうがよい。他のライブラリなども利用する作りになっている場合は、ライセンスの矛盾に陥らないようにすることも考慮しなければならない。ソフトウェアは動くところまで作るところが一番おもしろいところではあるが、それをメンテナンスしていくことがより重要なことである。機能追加などを行わなくても、まわりのソフトウェアの変化に合わせて修正が必要になることがよくある。それらの変化に対応して生き残っていくソフトウェアを開発することが大事である。

参考文献

- 1) Murdock, I. A. : The Debian Manifesto, <http://www.debian.org/doc/manuals/project-history/ap-manifesto.en.html> (1994).
- 2) Debian Project : Debian Social Contract & Debian Free Software Guidelines, http://www.debian.org/social_contract (1997, 2004).
- 3) Debian Project : Debian Policy Manual, <http://www.debian.org/doc/debian-policy/> (1996-2006).
- 4) Debian Project : Buildd: Debian Package Auto-builder, <http://build.debian.org/>
- 5) 武藤健志 : Debian GNU/Linux 徹底入門第 3 版 Sarge 対応, 翔泳社 (2005).
- 6) やまだあきら : [入門]Debian パッケージ, 技術評論社 (2006).
(平成 18 年 7 月 10 日受付)

