



Linux カーネル開発

その開発スピード、開発者、開発過程
および支援企業

Jonathan Corbet, LWN.net
Greg Kroah-Hartman, The Linux Foundation
Amanda McPherson, The Linux Foundation
March 2012



概要

Linux システムの中核を構成する Linux カーネルは、ソフトウェア開発の歴史の中でも特筆すべき大規模な共同作業の成果です。2、3 か月ごとに Linux ユーザーに出荷される安定した公開版には、それぞれに顕著な新機能、新たな装置サポート、性能改善が含まれています。カーネル修正数は増え続けており、最近のカーネル公開版には、毎回 8,000 件から 12,000 件のパッチが含まれています。それぞれのカーネル版の開発には、200 近くの企業の 1,000 人以上の技術者が参加しています。

2005年の調査開始以来、約 800 社の企業に属する 7,800 人以上の技術者がカーネル開発に貢献しています。Linux カーネルは、Linux 開発以外の領域では激しく競争しあっている多くの企業が共同開発した共通リソースです。

今回は、2008 年以降ほぼ毎年公開されているレポートの第 4 版です。Linux カーネル 3.2 公開版までをカバーしており、特に前回の更新からこの公開版まで（つまり 2.6.36 から 3.2 まで）の作業を重点的に取り上げています。この期間は 7 回のカーネル公開版が作成され、多くの重要な修正があり、カーネル開発者/ユーザー コミュニティの継続的な成長が見られた充実した時期でした。

はじめに

Linux カーネルは、Linux システムの基底で動作するソフトウェアです。その役割は、ハードウェアの管理、ユーザープログラムの起動、システムのセキュリティや統合性の確保などです。1991年 Linus Torvalds が最初に公開して以来、飛躍的に開発が進んだ Linux とはこのカーネルのことなのです。カーネルは Linux システム全体から見れば、相対的に小さな部分と言えるでしょう。他の大きなコンポーネントの例としては、GNU プロジェクト、GNOME や KDE のようなデスクトッププロジェクト、X.org プロジェクトなどがあります。しかし、カーネルは、システムを正しく動作させるという点に関しては最も重要な責任があり、また、Linux システムの中で他に代わるべきものではありません。

Linux カーネルというのは、いろいろな点で、研究するに値する興味深いプロジェクトです。カーネルはどのような Linux システムの中においても大きなコンポーネントの一つです。また、カーネルプロジェクトは、変化に素早く対応できる開発プロセスを特徴としており、オープンソースプロジェクトの中でも最大の開発者を擁しています。2005 年以降、カーネルの開発の経緯は、git と呼ばれるソースコード管理システムのおかげで、大変容易にドキュメントにすることができるようになりました。

2011 年のカーネル開発のハイライト

このレポートの中で示されているように、2011 年もまた、カーネル開発コミュニティにとって非常に充実した 1 年でした。この期間には、膨大な修正が行われただけでなく、以下のようにいくつかの注目すべき動きがありました。

- Linux カーネルが 20 周年を迎えました。Linus Torvalds の最初の発表は、1991 年 8 月 25 日に投稿されました。その後まもなく、同年 9 月に、最初の公開版（バージョン 0.01）が発表されました。
- 2004 年から使用されていた 2.6.x というナンバリング体系は、3.0 カーネルの公開により終わりを遂げました。3.0 公開版に特別な意味があったわけではありません。2.6 で始まるバージョン番号は扱いにくくなってきた（この議論は何年も前からありました）ということで、単純に決まったものです。
- Microsoft がカーネル公開版の上位 20 貢献企業に初めて入りました。
- カーネル開発の中央リポジトリ/ディストリビューション サイトである kernel.org が、深刻なセキュリティの侵害を受け、数週間オフラインになりました。その結果、3.1 カーネルの公開が遅れました。詳細な調査の結果、カーネルソースの統合性を危険にさらすような試みは行われなかったことがわかりました（そのような試みがあれば、いずれにせよほぼ確実にすぐさま発見されたでしょう）。この事件により、開発プロセスのセキュリティはさまざまな方法で強化されました。
- よく知られていることですが、カーネルの ARM アーキテクチャ サブツリーが大きくなり過ぎてしまいました。組込み開発コミュニティからの貢献が増えたことにより、ARM サブツリーが複雑になったことは確かです。ある意味、カーネルは自分自身の成功の被害者でもあります。2011 年の年末までには、このコードのクリーンアップ作業がだいぶ進められました。
- 1996 年 6 月の 2.0 公開版では、対称型マルチプロセッシング (SMP) サポートとともに、やっかいなビッグカーネル ロック (BKL: big kernel lock) が追加されました。その 15 年後の 2011 年、2.6.39 カーネル公開版で、このロックを排除するプロセスが完了しました。
- 2011 年の終わりには、“Long-Term Support Initiative (長期サポート構想)” が発表されました。このレポートの中で詳しく説明しますが、この活動は、安定した組込み製品のベースを提供することに重点を置き、特定のカーネル公開版に対する予測可能なサポートを提供するものです。

これらに加え、カーネルの開発と改良が、めまぐるしい速度で続けられました。このレポートでは、その開発プロセスの健全性と、すべてのコードの出自について解説していきます。

開発モデル

Linuxカーネル開発は、どちらかと言うと緩やかに、時間を守って公開版を提供する形で進行しており、2、3か月ごとにカーネルの改版が行われます。2005年に決定されたこの開発モデルでは、新機能をメインラインカーネルに取り込んでからユーザーに提供するまでの時間を最小にすることができます。また、この開発モデルにより、ディストリビューターの開発のスピードを上げ、彼らに加える修正を少なくすることができます。そうすることで、ディストリビューターの提供するLinuxの品質が上がり、ディストリビューション間の差異も少なくなります。

Linux Torvaldsが各カーネルのメジャーリリースを行うと、その後、そのカーネルの“stable team (安定版チーム: 現時点では Greg Kroah-Hartman)”が短期メンテナンスを担当し、重要な修正を適用します。この安定化プロセスのお陰で、重要なバグ修正がディストリビューターやユーザーに提供され、また、先々のメイラインカーネルにも間違いなくそれらの修正が適用されます。この安定化のための保守期間は、最短でも2つの開発サイクルの間続きますが、特定のカーネル版では、もっと長期に渡って行われます。近年は、業界が協力し合い、特定のカーネルを1年間以上メンテナンスする活動が増えています。

公開頻度

開発コミュニティで合意された、主要なカーネル版の望ましい公開間隔は8週から12週です。あまりにも短期間になると、テスターが新しいカーネルの問題を見つける時間がありませんし、期間が長過ぎると、大量の開発成果が滞ってしまいます。しかし、実際の間隔は、各版の修正の量や、最終テストにおける問題の解決の難しさなどに依存して多少変わることもあります。2.6.11以降の実際の間隔を、次の表に示しています。

カーネルバージョン	公開日	開発にかかった日数
2.6.11	2005-03-02	69
2.6.12	2005-05-17	108
2.6.13	2005-08-28	73
2.6.14	2005-10-27	61
2.6.15	2006-01-02	68
2.6.16	2006-03-19	77
2.6.17	2006-06-17	91
2.6.18	2006-09-19	95
2.6.19	2006-11-29	72
2.6.20	2007-02-04	68
2.6.21	2007-04-25	81
2.6.22	2007-07-08	75
2.6.23	2007-10-09	94
2.6.24	2008-01-24	108
2.6.25	2008-04-16	83
2.6.26	2008-07-13	88
2.6.27	2008-10-09	88
2.6.28	2008-12-24	76
2.6.29	2009-03-23	89
2.6.30	2009-06-09	78
2.6.31	2009-09-09	92
2.6.32	2009-12-02	84
2.6.33	2010-02-24	84
2.6.34	2010-05-15	81
2.6.35	2010-08-01	77
2.6.36	2010-10-20	80
2.6.37	2011-01-04	76
2.6.38	2011-03-14	69
2.6.39	2011-05-18	65
3.0	2011-07-21	64
3.1	2011-10-24	95
3.2	2012-01-04	72

ここに見られる日数から計算すると、現在、平均の公開間隔は 80 日になり、ちょうど 12 週の中に収まっています。サイクルの長さは、近年緩やかに短縮されており、多くは 70 日未満です。

修正の速度

開発成果を Linux カーネルに投稿する際、開発者は修正を小さな単位、パッチと呼ばれる修正提案に仕立てます。これらのパッチは、通常、単一の機能のみを実現するようになっています。それらが相互に干渉しないよう工夫して、ソースコードの行修正、行追加、行削除などを行います。各パッチが適用されたとき、当然ながら、カーネルは適切にビルドでき、正常に動作しなければなりません。この原則に従って、カーネル開発者は修正を小さな論理的単位に分割するため、結果的に、コードの品質と正確さを確実にレビューできるようになります。このような背景もあって、各公開カーネルに取り込まれる修正の数は、次の表に見られるように大変大きな数になっています。

カーネルバージョン	修正 (パッチ)
2.6.11	3,616
2.6.12	5,047
2.6.13	3,904
2.6.14	3,627
2.6.15	4,959
2.6.16	5,369
2.6.17	5,727
2.6.18	6,323
2.6.19	6,685
2.6.20	4,768
2.6.21	5,016
2.6.22	6,526
2.6.23	6,662
2.6.24	9,836
2.6.25	12,243
2.6.26	9,941
2.6.27	10,628
2.6.28	9,048
2.6.29	11,678
2.6.30	11,989
2.6.31	10,883
2.6.32	10,989
2.6.33	10,871
2.6.34	9,443
2.6.35	9,801
2.6.36	9,501
2.6.37	11,446
2.6.38	9,577
2.6.39	10,269
3.0	9,153
3.1	8,693
3.2	11,881

各カーネルの公開までに要した時間と一緒に見てみると、1時間あたりに採用された修正の数は、次の表のように計算できます。

カーネルバージョン	1時間あたりの修正の数
2.6.11	2.18
2.6.12	1.95
2.6.13	2.23
2.6.14	2.48
2.6.15	3.04
2.6.16	2.91
2.6.17	2.62
2.6.18	2.22
2.6.19	3.87
2.6.20	2.92
2.6.21	2.58
2.6.22	3.63
2.6.23	2.95
2.6.24	3.79
2.6.25	6.15
2.6.26	4.71
2.6.27	5.03
2.6.28	4.96
2.6.29	5.47
2.6.30	6.40
2.6.31	4.93
2.6.32	5.46
2.6.33	5.39
2.6.34	4.86
2.6.35	5.30
2.6.36	4.95
2.6.37	6.28
2.6.38	5.78
2.6.39	6.58
3.0	5.96
3.1	3.81
3.2	6.88

カーネル 2.6.11 から 3.2 に至る 2,581 日間に、1 時間当たり平均 4.3 件のカーネルパッチが作成されています。前回のレポート以降の期間だけを見ると、この数字は 5.64 件となり、さらに大きくなっています。Linux のカーネルの成長とともに、修正の速度も上がっています。

しかし、この数字は Linux カーネル開発の活動のレベルを過小に表していると見るべきです。多くのパッチは、メインラインカーネルで採用されるまでに何度も修正を繰り返しています。また、採用されなかったパッチも多くあります。過去のどんなソフトウェア開発プロジェクトでも、これほどの修正の速度を維持していたものはないでしょう。

安定版の更新 (Stable アップデート)

このレポートの冒頭でも触れたように、カーネル開発はメインラインのリリースのみに留まる活動ではありません。リリースされたカーネル内で、必然的にいろいろな問題が発見され、その問題を修正するためにパッチが作成されます。パッチの適用により、メインラインカーネルと現行リリースの両方を確実に修正できるように、安定版カーネル更新プロセスが設計されました。

これらの安定版の更新に基づいて、多くのディストリビューターのカーネルが作成されています。

次の表は、安定版カーネル更新の履歴を示しています (安定版カーネル更新プロセスは、2.6.11 リリース以降に導入されました)。

カーネルバージョン	安定版の更新回数	修正
2.6.11	12	79
2.6.12	6	47
2.6.13	5	39
2.6.14	7	89
2.6.15	7	103
2.6.16	62	991
2.6.17	14	177
2.6.18	8	232
2.6.19	7	185
2.6.20	21	447
2.6.21	7	155
2.6.22	19	366
2.6.23	17	302
2.6.24	7	243
2.6.25	20	481
2.6.26	8	321
2.6.27	61	1,879
2.6.28	10	611
2.6.29	6	379
2.6.30	10	431
2.6.31	14	819
2.6.32	57	3,315
2.6.33	20	1,877
2.6.34	10	1,323
2.6.35	14	1,609
2.6.36	4	687
2.6.37	6	592
2.6.38	8	634
2.6.39	4	441
3.0	22	1,375
3.1	10	694
3.2	7	389

この表でもわかるように、安定版のカーネルの更新の数は増えています。更新数が増えている主な要因は、開発コミュニティにおいて経験が積まれたことにあります。つまり、パッチを評価し、リリース済みのカーネルに適用可能なパッチをスムーズに特定できるようになったということです。また、安定版の更新が長期にわたって適用されているカーネルがある、というのも要因です。たとえば 2.6.32 カーネルは、現在でもさまざまなディストリビューターによってサポートされ、引き続き更新されています。

2011 年には、安定版更新の管理方法について、いくつかの変更が行われました。通常のカーネル公開版は 2 つのサイクル内で更新されますが、その後はサポートされません。しかし特定版のカーネルは、長期メンテナンスの対象になります。

組込み開発コミュニティは、毎年1つのカーネル公開版を選択し、それを2年間にわたってメンテナンスするというプランを作成しました。そして、3.0カーネルがその第1号に選ばれました。この活動の詳細については、'The embedded long term support initiative' (組込み開発システムの長期サポート構想) を参照してください。

安定版更新シリーズは、メインライン開発の促進を担保しつつ、同時にリリース済みカーネルに対してさらに最終的な修正を行うことにより、その価値を証明し続けています。

カーネルソースコードの規模

Linuxのカーネルソースコードは、Linuxがサポートするハードウェアの数とともに、またその機能の強化とともに、その行数を増やし続けています。以下の数字では、公開されたLinuxのソースパッケージ全体をソースコードとして数えています。ご承知のとおり、これらの中には、少量のビルトスクリプトやドキュメントも含まれています。もちろん、これらもLinuxの巨大な開発成果の一部であり、数えるに値する意味を持っています。

次の表は、カーネルの各バージョンのファイル数とソースコード行数を示しています。

カーネルバージョン	ファイル数	ソースコードの行数
2.6.11	17,090	6,624,076
2.6.12	17,360	6,777,860
2.6.13	18,090	6,988,800
2.6.14	18,434	7,143,233
2.6.15	18,811	7,290,070
2.6.16	19,251	7,480,062
2.6.17	19,553	7,588,014
2.6.18	20,208	7,752,846
2.6.19	20,936	7,976,221
2.6.20	21,280	8,102,533
2.6.21	21,614	8,246,517
2.6.22	22,411	8,499,410
2.6.23	22,530	8,566,606
2.6.24	23,062	8,859,683
2.6.25	23,813	9,232,592
2.6.26	24,273	9,411,841
2.6.27	24,356	9,630,074
2.6.28	25,276	10,118,757
2.6.29	26,702	10,934,554
2.6.30	27,911	11,560,971
2.6.31	29,143	11,970,124
2.6.32	30,504	12,532,677
2.6.33	31,584	12,912,684
2.6.34	32,316	13,243,582
2.6.35	33,335	13,468,253
2.6.36	34,317	13,422,037
2.6.37	36,189	13,919,579
2.6.38	36,868	14,211,814
2.6.39	36,713	14,537,764
3.0	36,788	14,651,135
3.1	37,095	14,776,002
3.2	37,626	15,004,006

カーネルのコードは、初回の報告時から約800万行以上増えており、2010年の更新時と比べても、約150万行増えています。1991年の最初の公開時には、わずか1万行でしたから、カーネルのコードは確実に増え続けています。1つ前の公開版より小さくなったのは、2.6.36カーネルだけです。このバージョンのサイズが小さくなったのは、大量のデフォルトコンフィグレーションファイルを整理したためです。重複コードを削除してカーネルツリーを整備するよう常に努力していますが、ソースコードの縮小は一朝一夕には成功しません。

だれがこの仕事を担っているか

Linux カーネル開発を担っている技術者の数も、また、それらの開発を支援している企業の数も増え続けています。次の表を参照してください。実際、Linux 開発コミュニティの人口は、この3年間で2倍になっています。

カーネルバージョン	開発者の数	既知の企業の数
2.6.11	389	68
2.6.12	566	90
2.6.13	545	94
2.6.14	553	90
2.6.15	612	108
2.6.16	709	111
2.6.17	726	120
2.6.18	815	133
2.6.19	801	128
2.6.20	673	138
2.6.21	767	143
2.6.22	870	180
2.6.23	912	181
2.6.24	1,057	193
2.6.25	1,123	232
2.6.26	1,027	203
2.6.27	1,021	187
2.6.28	1,075	212
2.6.29	1,180	233
2.6.30	1,150	249
2.6.31	1,166	227
2.6.32	1,248	261
2.6.33	1,196	238
2.6.34	1,150	243
2.6.35	1,187	209
2.6.36	1,176	207
2.6.37	1,276	221
2.6.38	1,198	220
2.6.39	1,258	239
3.0	1,131	331
3.1	1,168	212
3.2	1,316	226
以上の全バージョン	7,944	855

この数字は、過去数年間に開発貢献者の数が常に増え続けていることを示しています。

このような多くの開発者の数にもかかわらず、多くの仕事は、比較的少数の個人によってなされています。どの開発サイクルでも、およそ 1/3 の開発者は、パッチ 1 件のみの貢献です。過去 5 年半の間、上位 10 人の開発者が、およそ 9% の修正に貢献しています。また、上位 30 人の貢献が全体の 20% を占めています。次の表は、個々の開発者のリスト、彼らが貢献した修正の数、およびそれらが修正全体に占める割合を示しています。

名前	修正の数	修正の割合
David S. Miller	3,258	1.2%
Al Viro	2,840	1.1%
Takashi Iwai	2,637	1.0%
Ingo Molnar	2,348	0.9%
Tejun Heo	2,235	0.9%

名前	修正の数	修正の割合
Thomas Gleixner	2,190	0.8%
Paul Mundt	2,093	0.8%
Russell King	2,083	0.8%
Bartłomiej Zolnierkiewicz	2,074	0.8%
Adrian Bunk	2,042	0.8%
Stephen Hemminger	1,918	0.7%
Johannes Berg	1,915	0.7%
Greg Kroah-Hartman	1,899	0.7%
Mauro Carvalho Chehab	1,879	0.7%
Mark Brown	1,781	0.7%
Ralf Baechle	1,735	0.7%
Christoph Hellwig	1,716	0.7%
Alan Cox	1,703	0.6%
Andrew Morton	1,638	0.6%
Randy Dunlap	1,546	0.6%
Jean Delvare	1,467	0.6%
Joe Perches	1,456	0.6%
Hans Verkuil	1,307	0.5%
Ben Dooks	1,299	0.5%
Trond Myklebust	1,277	0.5%
Patrick McHardy	1,253	0.5%
Eric Dumazet	1,251	0.5%
Peter Zijlstra	1,237	0.5%
Neil Brown	1,182	0.5%
Mike Frysinger	1,163	0.4%

上記の数字は、カーネル 2.6.12 以降の git の履歴から出しています。前回のこの調査 (2.6.35) から 3.2 の間のコミットの数で調べてみても、同様の傾向がありますが、まったく同じというわけでもありません。

名前	修正の数	修正の割合
Mark Brown	887	1.3%
Thomas Gleixner	798	1.1%
Joe Perches	683	1.0%
Chris Wilson	639	0.9%
David S. Miller	636	0.9%
Axel Lin	632	0.9%
Eric Dumazet	614	0.9%
K. Y. Srinivasan	599	0.8%
Johannes Berg	581	0.8%
Al Viro	575	0.8%
Tejun Heo	537	0.8%
Ben Skeggs	536	0.8%
Dan Carpenter	519	0.7%
Takashi Iwai	517	0.7%
Mauro Carvalho Chehab	499	0.7%
Russell King	494	0.7%
Christoph Hellwig	450	0.6%
Jonathan Cameron	439	0.6%
Alex Deucher	422	0.6%
Larry Finger	391	0.6%
Felix Fietkau	382	0.6%
Roland Vossen	377	0.5%
Uwe Kleine-König	356	0.5%

名前	修正の数	修正の割合
Arend van Spriël	355	0.5%
Wey-Yi Guy	352	0.5%
Greg Kroah-Hartman	340	0.5%
Guennadi Liakhovetski	336	0.5%
Vasiliy Kulikov	325	0.5%
Randy Dunlap	324	0.5%
Hans Verkuil	308	0.4%

興味深いのは、Linus Torvalds (トータル 1,113 件、2.6.35 以降では 231 件) が、上位 30 のリストに入っていないことです。もちろん、Linus は、開発プロセスの中で不可欠な役割を継続していますし、彼の貢献は修正の件数だけで測れるものではありません。他のシニア カーネル デベロッパーについても同様です。彼らは他の開発者から提出されたパッチのレビューや管理に多くの時間を割いているため、自分自身のパッチを大量に記述することができません。(少し詳しく説明すると、この数字には、マージ コミットの数が含まれていません。マージ コミットというのは、一連の修正が、別の 1 つの修正にマージされるものです。Linus は、非常に多くのマージ コミットを行っています。マージ コミットの数も含めれば、彼は間違いなくこれらのリストに載っています。

どんな企業が開発を支援しているか

Linux カーネルというのは、たくさんの企業が利用できる共通の資産と言えるでしょう。Linux を利用している企業の多くが、カーネル開発に参加しているわけではありません。これらの企業は Linux の現状に満足しており、Linux 開発を特別の方向に引っ張って行く必要性を感じていないのかも知れません。しかし、下の表でもわかるように、カーネルの強化に向けて協力する企業数は増えています。

カーネル開発者を雇用している企業について、少し細かく見てみましょう。次のような方法で、個々の開発者の支援企業を判別しました。

- (1) email アドレスによる企業の特定
- (2) 投稿したソースコードに含まれている企業スポンサーの情報
- (3) 開発者への直接の問い合わせ

開発者はしばしば雇用先を変えるものですし、中には企業のオフィス外で個人的に開発に参加しているケースなどもありますので、示された数字は概数として見てください。それでも、この数字からいくつかの事実を引き出すことができます。

所属企業を特定することができなかつた開発者も多く存在します。次の表の企業名に "Unknown"(不明) と表示されているのは、そのような開発者のグループです。若干の例外もありますが、このカテゴリの各開発者による修正の投稿数は、ここ 3 年間で多くて 10 件程度です。それでもこれらの開発者の貢献の量を合計すると、膨大なものになります。

"None"(なし) と表示されているのは、企業からの支援がないことがわかっている開発者です。

Unknown と None を含めた上位 10 社が、カーネルに対する貢献の 60% 以上を占めています。注目すべきなのは、Unknown で示される開発者のすべてが企業の支援を受けていないとみなしても、カーネル開発の 75% 以上は、企業の正規の仕事として行われているということです。

企業名	修正の数	全体に占める割合
None (なし)	46,982	17.9%
Red Hat	31,261	11.9%
Novell	16,738	6.4%
Intel	16,219	6.2%
IBM	16,073	6.1%
Unknown (不明)	13,342	5.1%
Consultant (コンサルタント)	7,986	3.0%
Oracle	5,542	2.1%
Academia (学術関連)	3,421	1.3%
Nokia	3,272	1.2%

企業名	修正の数	全体に占める割合
Fujitsu	3,156	1.2%
Texas Instruments	2,982	1.1%
Broadcom	2,916	1.1%
Linux Foundation	2,890	1.1%
Google	2,620	1.0%
Analog Devices	2,595	1.0%
SGI	2,578	1.0%
AMD	2,510	1.0%
Parallels	2,419	0.9%
Freescale	2,265	0.9%
Cisco	2,259	0.9%
HP	2,158	0.8%
Renesas Technology	2,092	0.8%
MontaVista	2,019	0.8%
Atheros Communications	1,960	0.7%
Wolfson Microelectronics	1,952	0.7%
Marvell	1,752	0.7%
NetApp	1,746	0.7%
Linutronix	1,656	0.6%
Samsung	1,650	0.6%

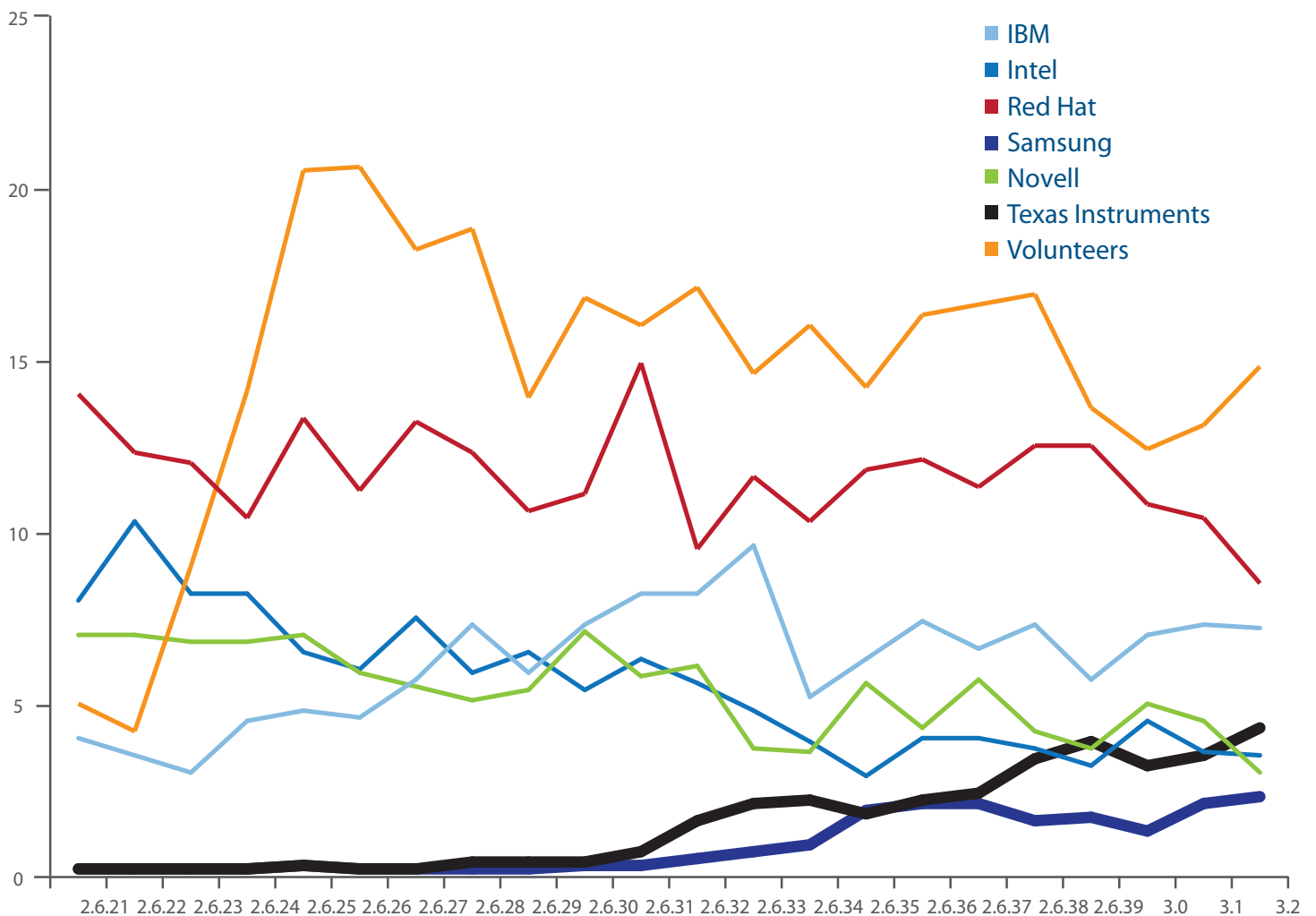
これを見てもわかるように、少数の企業がカーネル修正の多くに関わっています。しかし、重要な貢献を行っている企業数は「ロングテール」の様相を呈しており、この表に含まれていない企業が700社以上あります。このように巨大な共有資産で、これほど多くのグループによる共同作業が行われた例は、他に類を見ないでしょう。

カーネル 2.6.36 以降の数値を見てみましょう。興味深い変化が見られます。

企業名	修正の数	全体に占める割合
None (なし)	11,413	16.2%
Red Hat	7,563	10.7%
Intel	5,075	7.2%
Novell	3,050	3.3%
Unknown (不明)	2,998	4.3%
IBM	2,638	3.7%
Texas Instruments	2,124	3.0%
Consultant (コンサルタント)	1,859	2.6%
Broadcom	1,780	2.5%
Nokia	1,367	1.9%
Samsung	1,195	1.7%
Oracle	1,102	1.6%
Google	1,054	1.5%
Wolfson Microelectronics	1,005	1.4%
AMD	980	1.4%
Academia (学術関連)	882	1.3%
Fujitsu	854	1.2%
Pengutronix	733	1.0%

企業名	修正の数	全体に占める割合
Atheros Communications	726	1.0%
Freescape	712	1.0%
Microsoft	688	1.0%
ST Ericsson	663	0.9%
Wind River	645	0.9%
MiTAC	632	0.9%
Soc. Francaise de Radiotelephone	614	0.9%
Analog Devices	611	0.9%
tglx PITA	591	0.8%
Linaro	527	0.7%
QLogic	526	0.7%
Marvell	465	0.7%

上位 10社 の名前はほとんど変わっていません。今回も Red Hat は全体をリードしています。ただし次の図を見ると、興味深い傾向があることがわかります。



この図は、2007 年の 2.6.20 公開版以降に、活動が顕著な企業が貢献した修正の割合を表したものです。この期間の貢献の増減は、ほとんどの企業が似かよっています。つまり、カーネル開発の速度が上がるのに比例して、彼らの投稿数も増えています。

しかし下の方のラインには、異なる動きが見られます。この2本のラインは、モバイル/組込み開発企業である Samsung と Texas Instruments による投稿数を表しています。最近、この分野の企業からの参加が急激に増えています。これらの企業は、カーネルへのハードウェア サポートに力を入れるだけでなく、コア カーネル領域の改良にも責任を負うようになっています。

誰が修正を承認しているか？

パッチは、通常、いきなりメインライン カーネルに統合されるのではなく、その前に 100 以上のサブシステムツリーの 1 つに統合されます。各サブシステム ツリーは、カーネルのどこかの部分に対応しています。たとえば、SCSI ドライバー、x86 アーキテクチャ依存コード、ネットワーキングなどです。それらのサブシステム ツリーは、特定のメンテナーが管理しています。サブシステム メンテナーは、サブシステム ツリーへのパッチ統合を承認すると、“Signed-off-by” 行を追加します。この署名行は、当該パッチをカーネルに組み込んでも良いということを示します。1 つのパッチがカーネルに組み込まれるのに、一連の署名行が付加されることもあります。

署名行の中でも、特にパッチ作者以外の開発者が付加した署名行に注目すると、カーネル開発の興味深い一面を見ることができます。これらの署名行は、サブシステム メンテナーのレビューを表しています。署名行を分析すると、パッチをカーネルに組み込むことを誰が承認しているかがわかります。

次の表は、カーネル 2.6.35 以降、作者以外の署名行を多く記述した開発者のリストです。

名前	Sign off の行数	全体に占める割合
Greg Kroah-Hartman	7,848	5.8%
David S. Miller	6,246	4.6%
John W. Linville	4,146	3.1%
Linus Torvalds	3,266	2.4%
Mauro Carvalho Chehab	3,253	2.4%
Andrew Morton	2,687	2.0%
Mark Brown	2,131	1.6%
James Bottomley	1,609	1.2%
Takashi Iwai	1,282	0.9%
Russell King	1,245	0.9%
Ingo Molnar	1,216	0.9%
Thomas Gleixner	1,088	0.8%
Paul Mundt	1,029	0.8%
Dave Airlie	1,003	0.7%
Chris Wilson	944	0.7%
Al Viro	915	0.7%
Kukjun Kim	818	0.6%
Wey-Yi Guy	787	0.6%
Avi Kivity	762	0.6%
Artem Bityutskiy	761	0.6%

この表から、Linux Torvalds は 2% 強のパッチの組み込みに関与しているのみで、その他の多くはサブシステムメンテナーによって実施されていることがわかります。

それらの開発者を雇用している企業の名前は、次の表のとおりです。

企業名	Sign off の行数	全体に占める割合
Red Hat	26,225	37.7%
Novell	13,722	13.4%
None (なし)	13,587	9.2%

企業名	Sign off の行数	全体に占める割合
Intel	9,876	6.6%
IBM	4,801	4.8%
Google	4,057	3.5%
Texas Instruments	3,744	2.4%
Linux Foundation	3,270	1.8%
Unknown (不明)	3,161	1.4%
Consultant (コンサルタント)	2,899	1.3%
Samsung	2,463	1.1%
Wolfson Microelectronics	2,250	1.1%
Broadcom	2,198	1.1%
Microsoft	2,174	1.1%
Nokia	2,108	0.9%
Oracle	1,817	0.9%
Pengutronix	1,420	0.9%
Wind River	1,285	0.7%
Academia (学術関連)	1,126	0.5%
AMD	1,101	0.5%

ただし署名行の数は、レビューのようすをおおまかに示しているだけです。参考程度に見る必要があるでしょう。それでも、サブシステムメンテナが、カーネル開発者達よりも少ない企業に集中していることがわかります。半分以上のパッチがわずか3社によって雇用された開発者の手でカーネルへの組み込みを果たしているのです。

結論

Linux カーネルは、オープンソースソフトの歴史の中でも、最も大規模で、最も大きな成功を収めたプロジェクトの一つです。その修正の速度と貢献者の数は、力強く、活発なコミュニティの存在を証明しています。Linux が使用される環境の広がりに応じて、Linux カーネルは、絶え間なく進化を続けています。Linux の修正速度は、継続して増加中であり、開発者の数も、開発のプロセスに関与する企業の数も増加中です。Linux の開発のプロセスは、その開発速度をさらに上げることに何ら問題がないことを証明しています。

Linux の開発を支援する企業の数も十分に大きな数に上っており、多くの企業が Linux に貢献すれば利益を享受できる可能性があるにもかかわらず、未だ何も貢献していない状況でも心配ないレベルに達しています。サーバー、デスクトップ、および組み込み分野における昨今の Linux の普及を考えれば、貢献企業や個人的開発者の数は、間違いなく増え続けるでしょう。カーネル開発コミュニティは、新しい開発者の参加を歓迎しています。Linux カーネルへの貢献作業に興味のある方および企業は、ぜひ [“How to Participate in the Linux Kernel Community”](#) (「Linuxカーネル開発への参加方法」) をご覧ください。また詳細については、このレポートの著者が The Linux Foundation までご連絡ください。

謝辞

本稿の著者はともどもに、何千にも及ぶカーネル貢献者の皆様に感謝の意を表します。こうした方々の協力がなければ、このようなレポートは興味深いものになりません。

参考

このデータの数値の多くは、Jonathan Corbet が作成した “gitdm” ツールで作成されたものです。gitdmは、GNU GPL下で配布できます。 [git://git.lwn.net/gitdm.git](https://git.lwn.net/gitdm.git) から入手することができます。

本稿の元となる情報は、kernel.org サイトに掲載された公開カーネル、および、git カーネルリポジトリから引き出しました。ただし、email アドレスの変更や開発者情報の誤記を直すために、一部の git リポジトリのログ情報に手修正を加えています。統計値の計算には、スプレッドシートを使用しました。これらに使用したログ情報、スクリプト、およびスプレッドシートは、<https://github.com/gregkh/kernel-history> に置いています。

The Linux Foundation は
Linux の普及促進、保護、ならびに発展に取り組み
Linux/OSS がクローズドなプラットフォームに対抗するのに必要とされる
統合されたリソースとサービスを提供します。

The Linux Foundation、オープン コンプライアンス プログラム
およびその他の活動については、下記を参照してください。

<http://www.linuxfoundation.org/>

<http://www.linuxfoundation.jp/>

