



Linux Kernel Development

誰が Linux を開発しているか（第 2 版）

誰が Linux の開発に携わり、どんな企業が支援しているのか、
そして、その開発スピードは？（2009 年 8 月改版）

Greg Kroah-Hartman, SuSE Labs / Novell Inc.

Jonathan Corbet, LWN.net

Amanda McPherson, The Linux Foundation



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 ヶ月毎に Linux ユーザ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。カーネル修正の数はもともと大きな数に上っていましたが、その数はさらに増え続けています。最近のカーネル公開版には 10,000 におよぶ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 1000 人以上の技術者が参加しています。

2005 年以来、500 近くの企業に属する 5000 人以上の技術者がカーネル開発に貢献しました。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、一緒になって開発した共通の資産なのです。

2008 年 4 月に提供した最初の報告からの差異としては以下が挙げられます；

- ・各カーネルのリリースサイクルに参加する開発者の数は約 10% 増加しています。
- ・修正の数も顕著に増えています；カーネルに追加されるコードの量は、一日当たりで見ても 3 倍近くに増えました。
- ・カーネルのコード量全体は、270 万行も増えました。

カーネル開発コミュニティは、その数においても、また、その生産性においても、健全な成長を続けています。

序論

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 あるいは 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 Linux システムの中で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陰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。

本稿は、このソースコード管理システムの力を借りて、ここ 4 年のカーネル開発の歴史を振り返り（カーネル 2.6.11 から 2.6.30）、カーネル開発のプロセスがどのように機能しているかを見て

みます。本稿はこの報告書の第 2 版であり、2008 年 4 月に提供された最初の報告書のフォローアップです。したがって、カーネル 2.6.24 以降の開発をカバーしています。これらの 6 版において、多くはあまり変わっていませんが、変わりつつある点もあります。何と云っても、Linux カーネル開発のペースは継続して速まっています。

開発モデル

Linux カーネル開発は、どちらかと言うと緩い、時間を守って公開版を提供する形で進行しており、2-3 ケ月毎にカーネルの改版が行われます。この、2005 年に決定された開発モデルでは、新機能をメインラインカーネルに取り込んでから、ユーザに提供するまでの時間を最小にすることができます。この開発モデルにより、開発のスピードを上げ、また、ディストリビュータが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、また、ディストリビュータの提供する Linux の品質が上がり、アップストリームカーネルとディストリビューションの差異も少なくなります。

今回の報告に当たり、大きく変化した点は、**linux-next** と呼ばれるカーネルツリーが出来たことです。**linux-next** は、次のカーネル開発サイクルに向けたステージングの役割を担います。本稿を執筆している時点で、2.6.31 は安定化フェーズに入っていますが、**linux-next** は、カーネル 2.6.32 に取り込まれるべき修正を含んだツリーとなっており、開発者にとっては、近い将来の修正を展望し、かつ、次のカーネルの開発サイクルが始まったときに統合上の不具合を最小にするのに役立ちます。**linux-next** は、開発サイクルをスムーズにし、修正量の増大への対策となっています。

メインラインカーネルの正式出荷後、カーネル安定化チーム（現在、Greg Kroah-Hartmann と Chris Wright がリード）は、重要なバグ修正を適用する形で、短期的な保守を行っています。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やユーザに提供され、また、先々のメイン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、最短でも一つの開発サイクルの間行われますが、特定のカーネル版では、もっと長期に渡って行われます。

本稿は、主流の 2.6.x の方に焦点を当て、安定版への修正については除外して議論を進めて行きます。というのは、これらはいずれも小修正であり、かつ、安定版に入った小修正は、次の主要公開版に取り込まれるからです。

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 8-12 週です。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 以降の実際の公開履歴を表 1 に示しています。

Kernel Version	Release Date	Days of Development
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

表1 カーネルの公開頻度

これでお分かりの通り、平均の公開間隔は 81 日になっており、12 週の中に収まっています。

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、ちゃんとビルドでき、また、正常に動作しなければなりません。この原則は、カーネル修正がより小さな論理的単位となることを促し、また、コードレビューでの品質確保にも繋がっています。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、表 2 に見られるように大変大きな数になっています。

Kernel Version	Changes (patches)
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

表 2 各公開カーネルの修正数

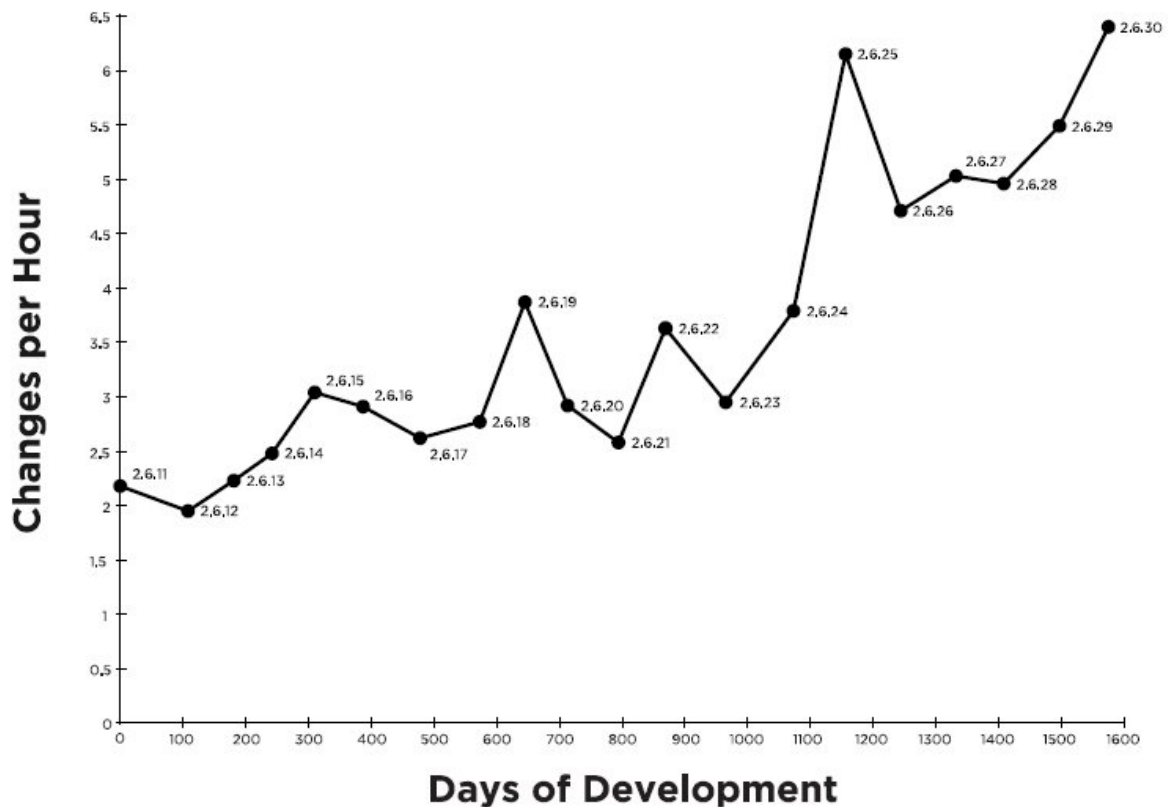
各カーネルの公開までに要した時間と一緒に見ると、1 時間あたりに採用された修正の数は、表 3 のように計算できます。

Kernel Version	Changes per Hour
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

表 3 1 時間あたりの修正数

カーネル 2.6.11 から 2.6.30 に到る 1560 日間に、1 時間当たり平均 3.83 件のカーネルパッチが作成されています。本報告の最初の版が出てから後だけで見ると、この数字は 5.45 件となり更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、この数字でさえも 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るのです。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。



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。ご承知の通り、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 の巨大な仕事の一部であり、数えるに値する意味を持っています。

表 4 は、各カーネル版のファイル数とソースコード行数を示しています。

Kernel Version	Files	Lines
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

表 4 各公開カーネルの規模

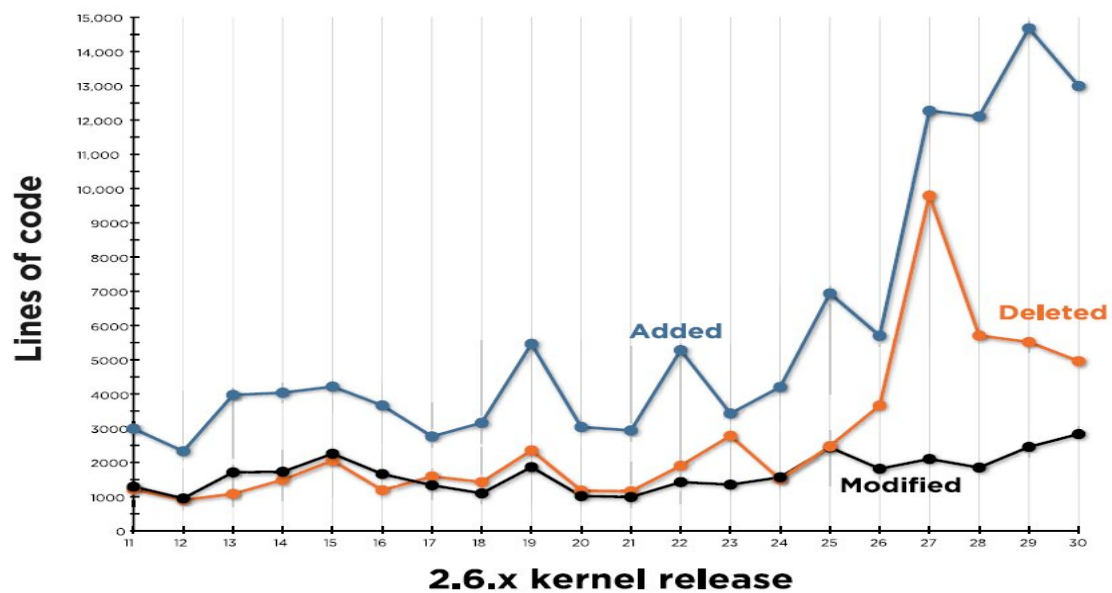
前回の報告時に比べて、カーネルは 270 万行も増えています。従来もカーネル行数の増え方は大きかったのですが、カーネル 2.6.27 以降、特に顕著になりました。主たる理由としてステージングツリーの存在があります。従来メイラインツリーから外れていたコード約 80 万行が、ステージングツリーの仕組みを活用してメインラインに組み込まれました。

しかし、カーネルは単に増えているだけではありません。カーネルのソースツリーに修正が加えられる毎に、コードの修正、追加、削除が行われています。これらの数を 1 日当たりで見ると、カーネルのソースツリーがいかに開発されているかが分かります。表 5 がこのような数字を示しています。

Kernel Version	Lines Added per Day	Lines Deleted per Day	Lines Modified per Day
2.6.11	3,224	1,360	1,290
2.6.12	2,375	951	949
2.6.13	4,443	1,553	1,711
2.6.14	4,181	1,637	1,726
2.6.15	5,614	3,454	2,219
2.6.16	3,853	1,388	1,649
2.6.17	3,635	2,469	1,329
2.6.18	3,230	1,497	1,096
2.6.19	6,013	2,900	1,862
2.6.20	3,120	1,342	1,013
2.6.21	3,256	1,479	982
2.6.22	6,067	2,694	1,523
2.6.23	3,747	3,034	1,343
2.6.24	6,893	4,181	1,563
2.6.25	7,980	3,488	2,430
2.6.26	5,698	3,662	1,815
2.6.27	12,270	9,791	2,102
2.6.28	12,105	5,707	1,850
2.6.29	14,678	5,516	2,454
2.6.30	12,993	4,958	2,830

表 5 各公開カーネルの修正行数

この数字を計算してみると、過去 4 年半の間、毎日 6,422 行が追加され、3,285 行が削除され、1,687 行が変更されたということが分かります。カーネル 2.6.24 以降、この数字は、毎日 10,923 行の追加、5,547 行の削除、2,243 行の変更に飛躍しています。もちろん、休日も含んだ計算です。



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、それら開発を援助している会社（会社の数をどのように数えるかについては、次の節で説明しています）の数も増え続けています。表 6 を参照して下さい。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。

Kernel Version	Number of Developers	Number of Known Companies
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	194
2.6.25	1,123	228
2.6.26	1,027	203
2.6.27	1,021	188
2.6.28	1,075	213
2.6.29	1,180	230
2.6.30	1,150	240
All	4,910	532

表 6 開発者の数とその雇用主

この数字は、過去 7 年間の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関わらず、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 4 年半の間、上位 10 人の開発者が、およそ 12% の修正に貢献しています。また、上位 30 人でみると 25% になります。表 7 に、個々の開発者のリスト、彼等が貢献した修正の数、それらが修正全体に占める割合が示されています。

Name	Number of Changes	Percent of Total Changes
David S. Miller	2,239	1.5%
Ingo Molnar	2,125	1.5%
Al Viro	1,981	1.4%
Adrian Bunk	1,883	1.3%
Takashi Iwai	1,801	1.2%
Bartlomiej Zolnierkiewicz	1,651	1.1%
Ralf Baechle	1,471	1.0%
Tejun Heo	1,457	1.0%
Stephen Hemminger	1,408	1.0%
Andrew Morton	1,370	0.9%
Paul Mundt	1,331	0.9%
Russell King	1,173	0.8%
Thomas Gleixner	1,164	0.8%
Alan Cox	1,145	0.8%
Greg Kroah-Hartman	1,100	0.8%
Patrick McHardy	1,087	0.8%
Andi Kleen	1,030	0.7%
Jean Delvare	985	0.7%
Mauro Carvalho Chehab	972	0.7%
Randy Dunlap	942	0.7%
Ben Dooks	906	0.6%
David Woodhouse	889	0.6%
Johannes Berg	880	0.6%
David Brownell	872	0.6%
Hans Verkuil	856	0.6%
Trond Myklebust	844	0.6%
Michael Krufky	814	0.6%
Alexey Dobriyan	806	0.6%
Herbert Xu	805	0.6%

表 7 カーネル開発への貢献者

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。本改版調査期間に合わせて、カーネル 2.6.24 から 2.6.30 の間のコミットの数で調べてみても、表 8 で示すように、傾向は同じですが、まったく同じという訳でもありません。

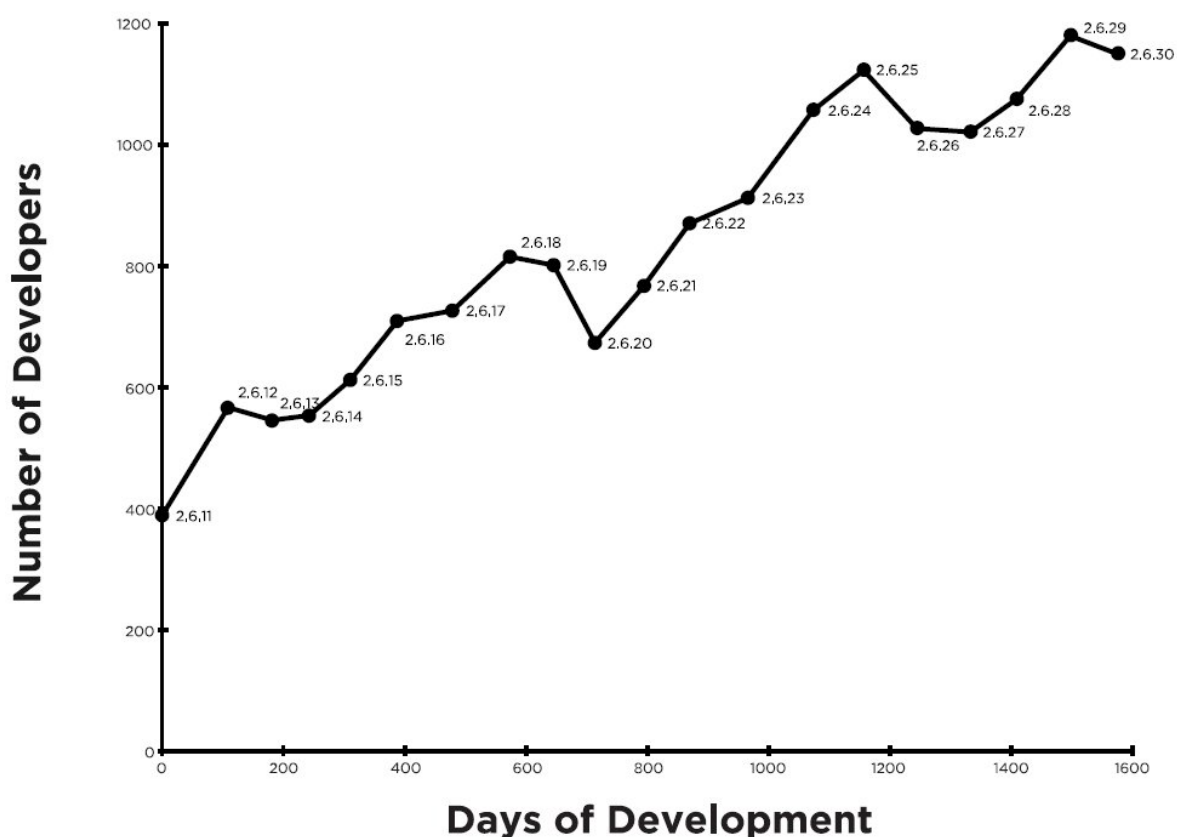
Name	Number of Changes	Percent of Total Changes
Bartlomiej Zolnierkiewicz	1,169	(1.8%)
Ingo Molnar	1,164	(1.8%)
David S. Miller	851	(1.3%)
Chris Mason	717	(1.1%)
Takashi Iwai	711	(1.1%)
Adrian Bunk	708	(1.1%)
Harvey Harrison	669	(1.0%)
Yinghai Lu	609	(0.9%)
Greg Kroah-Hartman	608	(0.9%)
Paul Mundt	570	(0.9%)
Al Viro	550	(0.8%)
Alan Cox	545	(0.8%)
Stephen Hemminger	522	(0.8%)
Hans Verkuil	517	(0.8%)
Steven Rostedt	511	(0.8%)
Mauro Carvalho Chehab	503	(0.8%)
Johannes Berg	483	(0.7%)
Jeremy Fitzhardinge	465	(0.7%)
Ben Dooks	438	(0.7%)
Tejun Heo	430	(0.7%)
Pavel Emelyanov	417	(0.6%)
Mike Frysinger	410	(0.6%)
Glauber Costa	406	(0.6%)
Patrick McHardy	404	(0.6%)
Michael Krufky	398	(0.6%)
Alexey Dobriyan	391	(0.6%)
Mark Brown	390	(0.6%)
Jean Delvare	389	(0.6%)
Thomas Gleixner	388	(0.6%)
Christoph Hellwig	385	(0.6%)

表 8 最近(2.6.24~30)の貢献者

少し興味深い話は、Linus Torvalds（トータル 729 件、2.6.24 以降では 254 件）が前回の報告以降、トップ 30 リストから落ちたことです。もちろん、Linux 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、また、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。（やや分かりにくい技術的詳細ですが、この数字は、マージコミットの数ではありません。マージコミットというのは、一連の修正が他の修正にマージされるものです。Linux は、大変多くのマージコミットを行っています。もしもマージコミットをちゃんと数えていけば、彼は間違いなくこのリストに載っています。）Linux はまた、非常に多くのレビューやコードのサインオフも行っています。これらに関しては、後述の「誰が修正を承認しているか？」の節をご参照下さい。

Number of developers per release



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。しかし、必ずしも、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とは限りません。これら企業の多くは Linux の現状に満足しているとも考えられますし、また、Linux を特別の方向に引っ張って行こうという意図がないのかも知れません。しかし、実際には表 9 に見えるように、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。

少し細かくカーネル開発者を雇用している企業を見てみましょう。個々の開発者の支援企業を判別するのに以下のようにしました。(1)email アドレスによる企業の識別、(2)投稿したソースコードにおける企業の支援の表示、(3)開発者への問い合わせ。開発者はしばしば雇用先を変えるものですし、また、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、示された数字はおおよそのものだと思って見ていただきたいと思います。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。

Company Name	Number of Changes	Percent of Total
None	26,644	18.2%
Red Hat	17,981	12.3%
Unknown	11,164	7.6%
IBM	11,151	7.6%
Novell	11,046	7.6%
Intel	7,782	5.3%
Consultant	3,657	2.5%
Oracle	3,513	2.4%
Linux Foundation	2,345	1.6%
SGI	2,317	1.6%
Parallels	1,939	1.3%
Renesas Technology	1,925	1.3%
Academia	1,712	1.2%
Fujitsu	1,592	1.1%
MontaVista	1,564	1.1%
MIPS Technologies	1,537	1.1%
Analog Devices	1,467	1.0%
HP	1,415	1.0%
Freescale	1,375	0.9%
Google	1,261	0.9%
linutronix	1,246	0.9%
Astaro	1,109	0.8%
NetApp	1,049	0.7%
Marvell	894	0.6%
Nokia	842	0.6%
Simtec	820	0.6%
QLogic	808	0.6%
Movial	776	0.5%
AMD	775	0.5%
Sun	764	0.5%

表 9 カーネル開発を支援する企業

開発者のかなりの数が、表 9 で、unknown と表示されていますが、彼等の支援企業は確認で

きませんでした。例外もありますが、このカテゴリの開発者は、ここ 3 年間で、大体多くて 10 件くらいの修正を投稿しています。しかしながら、これら開発者の占める貢献の量は トータルとして大変大きなものです。none と表示されているのは、企業からの支援がないことが分かっている開発者です。unknown と none を含めた上位 10 カテゴリがカーネルに対する貢献の 70% を占めています。unknown で示される開発者が全て企業の支援がないとしても、なお、70% のカーネル開発は、企業の正規の仕事としてなされているわけです。

また、ここで見えてくることの一つは、少数の企業がカーネル修正の多くに関わっていることです。また同時に、貢献企業数は、いわゆる、ロングテールというべき様相を呈しています。残念ながらこれら 500 社の名前は上記の表では現れていませんが、顕著な貢献を行っています。このように巨大な共有資産で、このように多くのグループによる共同作業が行われた例は未だ他に類がないでしょう。表 10 に示すように、カーネル 2.6.24 以降の様子は興味深い変化を見せています。

Company Name	Number of Changes	Percent of Total
None	13,850	21.1%
Red Hat	7,897	12.0%
IBM	4,150	6.3%
Novell	4,021	6.1%
Intel	3,923	6.0%
Unknown	2,765	4.2%
Oracle	2,003	3.1%
Consultant	1,480	2.3%
Parallels	1,142	1.7%
Fujitsu	1,007	1.5%
Academia	992	1.5%
Analog Devices	889	1.4%
Renesas Technology	884	1.3%
SGI	755	1.2%
Movial	738	1.1%
Sun	639	1.0%
HP	628	1.0%
Freescall	613	0.9%
Marvell	601	0.9%
MontaVista	574	0.9%
AMD	552	0.8%
Nokia	549	0.8%
Vyatta	513	0.8%
Google	512	0.8%
Atheros Communications	494	0.8%
NTT	445	0.7%
linutronix	445	0.7%
XenSource	432	0.7%
Simtec	414	0.6%
Astaro	411	0.6%

表 10 カーネル 2.6.24 以降の支援企業

雇用者なし(**none**)の開発者の割合が増えていますが、これは私たちのデータベースがより正確になったことによります。これらの多くが前回報告では不明(**unknown**)と分類されていました。トップ 10 社の名前はほとんど変わっていません。今回もレッドハットは全体をリードしています。しかし、オラクルや富士通がより高いレベルの貢献をしていることが分かります。また、その製品をメインカーネルでサポートすることを決定した **Atheros** のような会社に加わったことも歓迎すべきことです。

誰が修正を承認しているか？

パッチは、普通、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、その前に約 100 位あるサブシステムツリーの一つに統合されます。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。例えば、SCSI ドライバーとか、x86 アーキテクチャー依存コード、ネットワークなどです。それらのサブシステムツリーは、特定のメンテナが管理しています。サブシステムメンテナが、サブシステムツリーに統合することを承認すると、メンテナは、**signed-off-by** というタグ行を付けます。このタグが、当該パッチがカーネルに組み込んでも良いということを表します。一つのパッチがカーネルに組み込まれるのに、一連の **signed-off** 行がつくこともあります。

signed-off 行を分析すると、特に、パッチ作者以外の開発者が付加した **signed-off** 行を見ると、カーネル開発に関する、別の興味深い見方ができます。これらの **signed-off** 行は、サブシステムメンテナのレビューを表しています。**signed-off** 行の分析は、誰がパッチをカーネルに組み込むことを承認するかという仕組みを見せてくれます。

カーネル 2.6.24 以降における **signed-off** 行の数は表 11 に示しています。

Andrew Morton	6,515	10.5%
Ingo Molnar	6,174	9.9%
David S. Miller	5,954	9.6%
John W. Linville	3,733	6.0%
Mauro Carvalho Chehab	3,363	5.4%
Greg Kroah-Hartman	2,394	3.8%
Jeff Garzik	1,879	3.0%
Thomas Gleixner	1,707	2.7%
Linus Torvalds	1,664	2.7%
James Bottomley	1,302	2.1%
Takashi Iwai	1,093	1.8%
Len Brown	1,033	1.7%
Russell King	891	1.4%
Paul Mackerras	876	1.4%
Bryan Wu	806	1.3%
Avi Kivity	739	1.2%
Jaroslav Kysela	660	1.1%
Martin Schwidefsky	504	0.8%
Ralf Baechle	495	0.8%
Bartłomiej Zolnierkiewicz	486	0.8%
Jesse Barnes	477	0.8%
David Woodhouse	474	0.8%
Roland Dreier	472	0.8%
Paul Mundt	467	0.8%
Jens Axboe	402	0.6%
Patrick McHardy	395	0.6%
Lachlan McLroy	383	0.6%
Kumar Gala	381	0.6%
Benjamin Herrenschmidt	378	0.6%
Mark Fasheh	376	0.6%

表 11 signed-off の数

この表から、Linux Torvalds は 3%以下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分かります。

それらの開発者を雇用している企業の名前は、表 12 に示しています。

Red Hat	22,652	36.4%
Google	6,530	10.5%
Novell	5,076	8.2%
None	4,717	7.6%
Intel	3,986	6.4%
IBM	3,321	5.3%
linutronix	1,741	2.8%
Linux Foundation	1,666	2.7%
Consultant	1,213	1.9%
Hansen Partnership	965	1.6%
Analog Devices	851	1.4%
SGI	793	1.3%
Oracle	725	1.2%
MIPS Technologies	497	0.8%
Renesas Technology	486	0.8%
Qumranet	477	0.8%
Cisco	474	0.8%
Unknown	452	0.7%
Freescall	447	0.7%
Astaro	395	0.6%

表 12 signed-off を行う開発者の雇用企業

signed-off の分析は、レビューの様子はやや荒い評価ということになります。それでも、サブシステムメンテナーが、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分かります。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。

なぜ企業はカーネル開発を支援するのか

Linux カーネル開発を支援する企業のリストは、多くの有力 IT 企業を含んでいます。これら企業は慈善事業のようなことを行っているわけではありません。各ケースにおいて、これら企業はカーネルの強化がその企業の市場における競争力強化に繋がると見ているわけです。いくつか例を挙げてみましょう；

・IBM、インテル、SGI、MIPS、フリースケール、HP、富士通、その他は、皆、それら企業のハードウェア製品上でより良く **Linux** を動作させることを狙っています。そうすることで、これら企業の製品が **Linux** ユーザにとって魅力的になり、ビジネスに寄与すると考えるわけです。

・レッドハット、ノベル、モンタビスタのようなディストリビュータは、**Linux** を強化することに明確な意思を持っています。これらディストリビュータは、対顧客では競争していますが、**Linux** カーネルをよくすることに関しては相互に協力しています。

・ソニー、ノキア、サムソンのような企業はビデオカメラ、テレビ、携帯電話などのコンポネントとして **Linux** を採用しています。**Linux** 開発プロセスへの参加は、**Linux** を将来にわたって製品の重要な部品とすることに対する保障となります。

・IT 産業でないような企業にとっても **Linux** 開発への参加には意味があります。カーネル 2.6.25 には、PF_CAN ネットワークプロトコルという機能が実装されましたが、これはフォルクスワーゲン社の貢献です。また、2.6.30 には、Quantum Controls DV 社のパッチが採用されました。これは、ヨットの航海システムを実現するものです。これらの企業は **Linux** が信頼性のあるプラットフォームとなり、彼らの製品の基礎となることを望んでいます。また、**Linux** が将来にわたり、彼らの要件に適合し続けることができるよう、彼ら自身も **Linux** への貢献を継続しています。他のオペレーティングシステムでは、将来の開発にユーザが強い影響力を与えることはできません。

このように、企業が **Linux** カーネルの開発を支援するのにはいろいろな理由があるのです。結果として、**Linux** は広範な支持基盤をもち、特定少数の企業に支配されない仕組みとなっています。たとえ、大きく貢献している開発者が明日から開発貢献を打ち切ると宣言しても、**Linux** カーネルは、堅固な開発コミュニティのおかげでその歩みを止める心配はありません。

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、また、開発者の数も、また、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、ある日突然、開発支援企業が支援を打ち切ったとしても、心配ないレベルに達しています。**Linux** はサーバ、デスクトップ、組み込みシステムの市場で広く利用されていますが、現在の支援企業の数、また、貢献する開発者の数は、今後ますます多くなると考えることができます。カーネル開発コミュニティは新規開発者の参入を常に歓迎しています。**Linux** カーネルに貢献しようとする意図のある個人や企業は、是非、**How to Participate in the Linux Community**、(和訳)「**Linux** カーネル開発への参加方法」を一読下さるか、本報告者の著者または **The Linux Foundation** にお問い合わせ下さい。

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。これらの方々がいなければ、本稿のような試みは誰の興味も惹きません。

参考

本稿に出て来るいろいろな統計数字を抽出するのに、Jonathan Corbet の gitdm ツールを使用いたしました。gitdm は、GNU GPL 下で配布できます。次の web サイトで入手できます；

`git://git.lwn.net/gitdm.git`

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。これらに使用したログ情報、スクリプト、スプレッドシートは、以下に置いています。

http://www.kernel.org/pub/linux/kernel/people/gregkh/kernel_history