

# Demography of Linux Kernel Developers

Timo Aaltonen and Jyke Jokinen

22nd December 2006

## Abstract

Several success stories of open source (OS) products have been seen during last decade. Due to the economical importance of the products, it is important to know who are the ones who have the largest influence to the products. Therefore, studying demography of open source projects is essential. In this paper the aspect is studied with respect to the Linux Kernel. We show that the influence is centered to a small number of core people, and corporates have a large impact to the development. Moreover, we enumerate the most influential companies of the Linux Kernel community. Besides influence, we have touched the nature of development from the point of view of the actual code.

## 1 Introduction

Open source (OS) software development has gained much attention lately. During last decade several success stories, like Apache, Mozilla and Linux, has been seen. Apache is the market leader of the world's web servers [1] having over three times the market share of its next-ranked (proprietary) competitor. Internet Explorer has been losing market share to OS web browser, especially to Mozilla [2]. Linux [3] is a free UNIX-type operating system originally created by Linus Torvalds.

Due to the economical importance of open source, it is important to know, who influences the development. Is it carried out by altruistic individuals and what is the impact of large organizations? By knowing these facts one is able to predict the directions how the products evolve in future. This is essential when choosing between different opens source and proprietary alternatives.

This paper studies the influence of the developers and leaders of the Linux Kernel. The Kernel was chosen because it is the only operating system challenging Microsoft Windows, the available amount of data is large, and the number of people working for the project is numerous. All measurement are applied to data mined from GIT repository, which contains development source code.

The rest of this paper is structured as follows. Section 2 introduces revision control system GIT and discusses how data is mined from it. The applied measures are presented in the next Section 3. In Section 4 measures related to individual stakeholders are applied to the data. Section 5

deals with applying company-related measure, and in Section 6 measures are applied to actual code. Section 7 contains the discussion.

## 2 GIT Repository

### 2.1 GIT

GIT [4] is a revision control system written originally by Linus Torvalds for the use in the Linux Kernel development. Following UNIX tradition, the GIT is a collection of low-level command line utilities implementing a distributed source code management system (SCM). These low-level commands were originally meant to be used as a library for higher level SCM applications. In practice many Linux Kernel developers use the GIT commands directly in their work. Howto-documents also encourage this usage [5].

### 2.2 Generating The Database

Data source used in our work is the GIT repository recommended in “Kernel Hackers’ Guide to git” [5]. First a local working copy of the database is made:

```
git pull \  
git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git
```

Then every entry in the GIT database is listed with command:

```
git-log --pretty=oneline
```

These entries covered GIT database entries from 16 April 2005 to 6 November 2006 (total of 40670 entries). We decided to use one year interval between Jul 1st, 2005 and Jul 1st, 2006. Every log entry with “commit time” between these dates and containing at least one “signed-off-by:”-line [6] were collected (25228 entries). A signed-off-by line has two main meanings in the GIT data: first the original author marks the code copyright by adding the first signed-off-by, later code maintainers mark that they accept the patch by adding their own signed-off-by line. Our database does not separate these two meanings.

#### 2.2.1 Database format

Every GIT database entry contains a 40 character identification string (a SHA1 checksum of the patch-data), which is used as a primary key to identify entries in our database.

GIT has several output formats for patch-entries. We used raw-format by reading GIT database with command:

```
git-show --pretty=raw <gitID>
```

This produces information like:

```
commit db38c179a759a9c4722525e8c9f09ac80e372377  
tree 92edcdcec2fea73cd449a00e6e000ad5e53fec7b  
parent 0f37c6057414fb68024793966b1dcb6a135cb844
```

author Larry Woodman <lwoodman@redhat.com> 1162598745 -0800  
committer David S. Miller <davem@sunset.davemloft.net> 1162764692 -0800

[NET]: \_\_alloc\_pages() failures reported due to fragmentation

We have seen a couple of \_\_alloc\_pages() failures due to fragmentation, there is plenty of free memory but no large order pages available. I think the problem is in sock\_alloc\_send\_skb(), the gfp\_mask includes \_\_GFP\_REPEAT but its never used/passed to the page allocator. Shouldnt the gfp\_mask be passed to alloc\_skb() ?

Signed-off-by: Larry Woodman <lwoodman@redhat.com>  
Signed-off-by: David S. Miller <davem@davemloft.net>

```
diff --git a/net/core/sock.c b/net/core/sock.c
index d472db4..ee6cd25 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
...
```

This data is split into several database tables (Figure 1):

**log:** header information: commit id, link to author (person table), link to committer, and UNIX timestamps for these.

**person:** person data from author, committer, or signed-off-by.

**signature:** link to a GIT entry and a person. When one log entry contains several signed-off-by entries, each one has one row in this table.

**diff:** “diff” lines in the GIT entry. Files changed, lines added, and removed are recorded.

### 2.2.2 Person identifications

Person names are found in three places in the GIT data: author name, committer name, and signed-off-by lines. Each containing person’s name and e-mail address. Since person’s names seemed to contain variations, e.g. “Jyke Jokinen”, “Jokinen Jyke”, “Jyke T. Jokinen”, we decided to use the e-mail addresses to identify persons.

In processing a person’s data it is first split into two parts: author name and an e-mail address found between angle brackets (<>). When an e-mail address is used as an unique identifier, users having different hostnames within the same organization would be identified as different users (e.g. ‘torvalds@home.osdl.org’, ‘torvalds@ppc970.osdl.org’, and ‘torvalds@evo.osdl.org’).

To address this problem an e-mail address was further split into username and domainname parts by splitting in at-character location (**username@domain.name**). Domainname parts were converted into sub-domain lists (a dot separating the parts). Only two last parts of this list were used. Exceptions to this rule where country domains ‘jp’, ‘uk’, and ‘tw’ where three parts were used.

Database contains 1722 persons after collecting all person data in the year interval and using this shortened e-mail address as unique identifier.

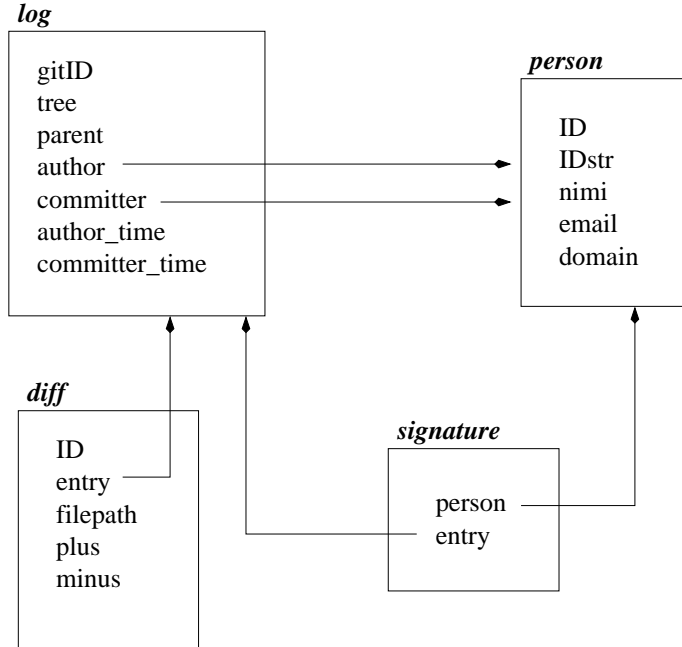


Figure 1: database schema

### 3 Measures

We have developed a set of measures to be applied to our data. The measures are divided into three categories: *personal*, *company-related*, and *code-related*. The personal measures attempt to highlight various aspects of people in the Linux Kernel community:

- **Acceptance spectrum.** Number of signed patches are counted for each person. Then these (person, amount) pairs are sorted in descending order. The measure illustrates how the control and development work is distributed in the community.
- **E-Mail domain distribution.** The Linux Kernel development is highly geographically distributed. This measure shows where and by which kind of organizations does the decision-making takes place.
- **E-Mail taxonomical distribution.** Measure attaches a category to e-mails from taxonomy: corporate, open source project, ISP, e-mail provider, university, personal domain, and other.

The company-related measures attempt to reflect the role of companies in the development:

- **Impact of Companies.** Leaders and developers of the Linux Kernel community signing the patches are related to companies they work for. Then the influence of employers of each company are summed together. This sum is the influence of the company.

The code-related measures attempt to highlight the code units which have been under development during our time window.

- **Impacted directories.** The source code is divided into directories, which in turn might be divided into subdirectories. By studying which directories or subdirectories are impacted by the patches, it can be studied which components have evolved during the time interval.
- **Impact of patches.** The size and nature of the patches has been studied.

## 4 Measures for Individuals

### 4.1 Acceptance Spectrum

The acceptance spectrum of the Linux Kernel developers is depicted in Figure 2. The number of signed patches is on the y-axis and individual signers are on the x-axis sorted with respect to the number of signs-offs.

A notable shape of the curve slanting to the left is quite common in open source projects. Actually, the y-axis has been truncated to make the shape of the curve more visible. The curve takes this shape because a small number of core people lead the whole community. In our previous studies we have noticed that a small group of developers contribute more than the rest of the group. For example, 3.4% of the developers of Gnome [7] produce 50% of the code [8]. The same phenomenon is visible in the figure for the Linux Kernel. We call this phenomenon the *flagpole effect*.

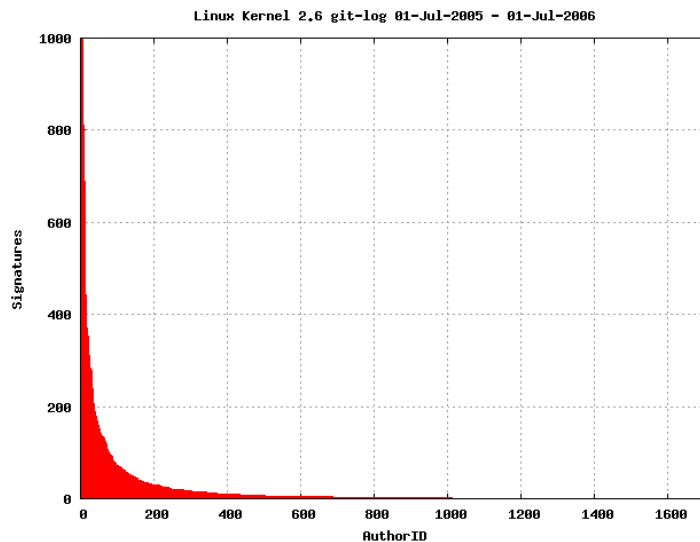


Figure 2: Acceptance spectrum.

To make clearer the strength of the flagpole effect, the acceptance spectrum is redrawn on a logarithmic scale in Figure 3. It is somewhat surprising, that even now, the curve tends to slant to the left so heavily.

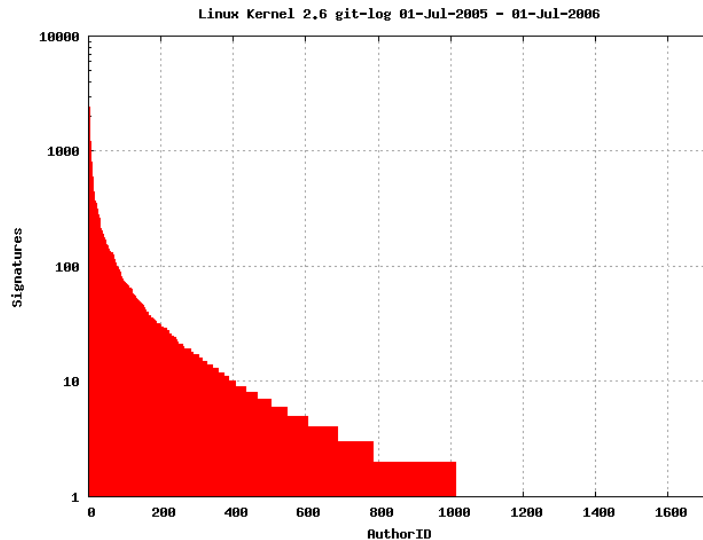


Figure 3: Acceptance spectrum in logarithmic scale.

## 4.2 E-Mail Domain Distribution

The Linux Kernel development is highly distributed. The measure related to distribution is based on studying the e-mail addresses of the leaders who sign the patches. Figure 4 illustrates the distribution with respect to highest level domains. Not surprisingly, *com* domain is the number one in this measure. The second place is taken by *org*, and the third one is occupied by *de* domain, implying that many of the Kernel developers are from Germany.

## 4.3 E-Mail Taxonomical Distribution

Each e-mail address was attached a category from taxonomy: *corporate*, *open source project*, *ISP*, *e-mail provider*, *university*, *personal domain*, and *other*. Google was used manually to attach taxon to the e-mail addresses. The results are illustrated in Figure 5. The distribution has one unexpected result: category *personal domain* taking the second place is somewhat surprising.

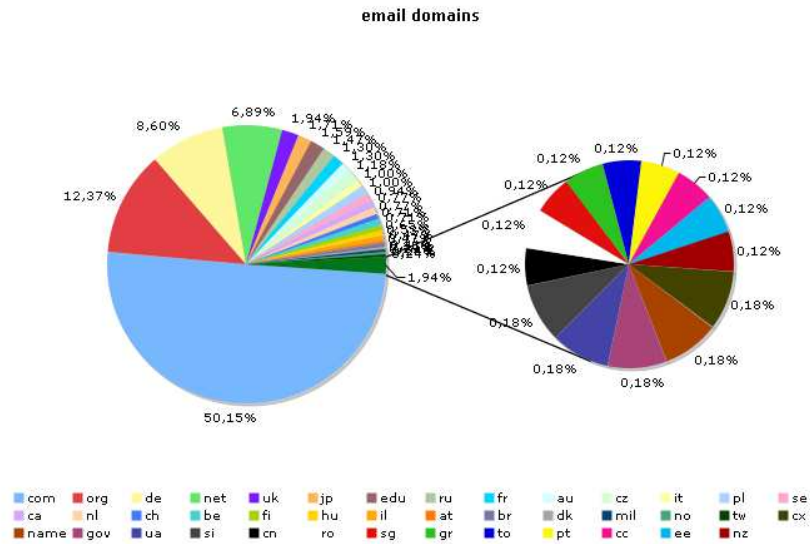


Figure 4: The e-mail domains of patch .

Category	Number
corporate	342
personal domain	207
other	200
university	114
ISP	110
open source project	78
e-mail provider	21

Figure 5: The taxonomy of e-mail addresses.

## 5 Measures for Companies

We took a closer at the top 100 signers according to criterium of most signed patches, and used Google search engine to study whether the top 100 leaders were employed by some organization. Then we were able to calculate the size of the impact of the organizations to the Linux Kernel development.

The search techniques we used were various. We had two obvious starting points: a name and an e-mail address. If a developer had a company-related e-mail then it is quite obvious that she works for the company. Few developers had their CV on www, which was easy to find with a simple search. Book publishers and organizer of open-source-related conferences maintain lists of their contributors with a small description of people's careers on www. Often, these people were among the top 100 leaders to the Linux Kernel. One surprisingly fruitful technique was search with the name part from an e-mail address. People seem to preserve their original e-mail names in their e-mail addresses. This way the employer was joined to a set of contributors. Some people were found from Wikipedia [9]. Moreover, several creative searches were carried out.

The results of *Impact of Companies* measure are shown in Figure 6. The company with the largest impact during our time interval has been SteelEye Technology. Actually, all 928 signatures related to the company have been signed by a single person. Obviously SteelEye Technology has been very active during our time window, and perhaps all patches from the company are signed by the person. After SteelEye Technology, the next companies should not be a surprise. Google's rank has been improved by Andrew Morton's migration to the company.

## 6 Measures for the Code

The top-level directory view to the Linux Kernel source code roughly divides the Kernel into subsystem categories. The directories include *arch* for code-related to hardware architectures, *drivers* for device drivers, *Documentation*, *fs* for file system, *include* for *c* language header files, *net* for high-level networking and 11 other directories.

Figure 7 illustrates the top-level directories affected by patches during our time window. More than one third of all patches are targeted to device drivers, one fourth are for hardware architectures, every seventh patch deals with *c* header files.

The top-level directories are again divided into subdirectories. A closer look was taken in the two most patched directories in Figures 8 and 9. The figure 8 reveals that most of the driver patches are targeted to media, networking and SCSI related devices. All in all, patches are targeted to 64 different device categories.

Taking a closer look at the architecture directory shows that PowerPC, Arm and MIPS architectures have been under heaviest development during the year interval. All in all, 25 different architectures were under development in the time period.

<b>Company</b>	<b>impact</b>
SteelEye Technology	928
IBM	924
Google	759
Intel	742
Novell	665
OSDL	588
UNKNOWN	453
Cisco (Topspin)	421
Debian	376
Alcatel	322
Red Hat	302
Netfilter (not a company, but a project)	293
Linutronix	283
Conectiva	280
Ameritech (American Information Technologies)	260
Dunvegan Media	184
Simtec Electronics	165
Wise Riddles Software	164
SGI	155
Levanta (previously Linuxcare)	138
Oracle	136
Symantec	135
Academic (all universities)	135
MISC (creative way for living)	133
Broadcom	131
Deep Blue Solutions Limited	121
QLogic	114
CoopTel	107
MontaVista Software	105
Freescale	98
Hewlett-Packard	94
Network Appliance	92
Circle Computer Resources	86
Mellanox Technologies	85
Ultra	79
Toshiba	77
Motorola	74

Figure 6: Companies and the number of patches signed by the personnel.



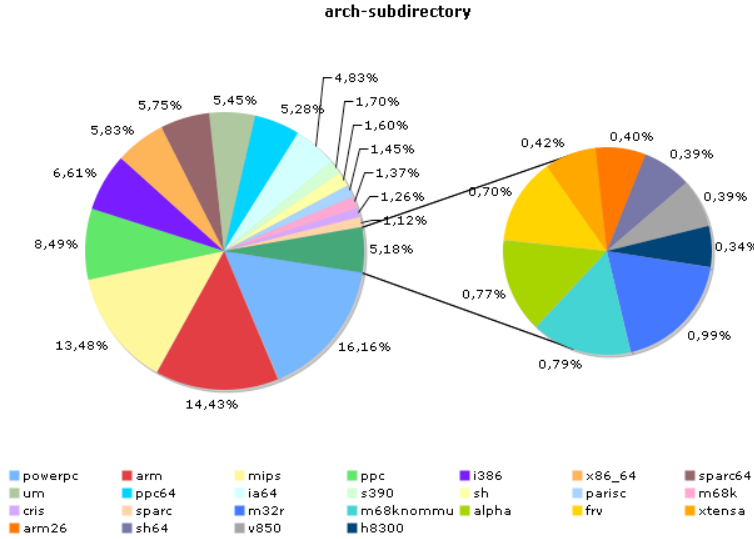


Figure 9: The impact of patches to subdirectories of arch.

## 6.1 Impact of Patches

In our time window 78106 files were changed. Total number of lines added was 2.289.399, and the number of removed lines was 1.490.408. The reader is reminded of the fact that these measures are gathered from data created with *diff* command, which can describe both adding and removing lines. Figure 10 includes data about the sizes of delivered patches.

## 7 Discussion

We studied the Linux Kernel development based on patch signature information. The information was mined from the GIT repository. Six measures was applied to the mined data.

The measure was divided into three categories: personal, company, and code-related. The personal measures show that control in the community is heavily concentrated to a small group of people. Similar results have been reported earlier in [8]. Open source communities have been described by the onion model [10], in which *project leader* is in the center, *core developers* form the next layer, then are *active developers* and so on. The flagpole effect somewhat reveals similar characteristics of the Linux community. If signing a patch means having influence on the community, we can claim that only 12 leaders out of 1722 have 50% of the influence. In other words 0.7% of the leaders have a half of the influence.

E-Mail domain distribution and taxonomical distribution show that the Linux Kernel is mostly developed in western countries and corporations have half of the control. Idealistic thoughts of Linux being a large

item	size
patches only adding new lines	17143
patches only removing lines	12999
patches adding and removing	47964
patches adding one line	17724
patches adding two lines	8675
patches adding 3-5 lines	24127
patches adding 6-10 lines	8111
patches adding >10 lines	19469
patches removing one line	21298
patches removing two lines	8466
patches removing 3-5 lines	27137
patches removing 6-10 lines	6823
patches removing >10 lines	14382

Figure 10: Statistical data of the impact to the Linux Kernel.

volunteer community consisting of altruistic programmers can be abandoned. Besides corporations, universities have a quite large impact.

Referring to code-related measures, typical patches are quite small only adding or removing few lines. Actually, breaking a large patch into several smaller ones are encouraged by the community [11], since smaller changes are easier to grasp.

Quantitative measurement of open source has been published earlier. In [12] project from the SourceForge repository have been examined. In [13], the geographical distribution and personal background of open source developers is documented.

## 7.1 Future Work

In this paper we studied patch data with respect to signing information, which is related to making decisions. Therefore, the personal- and company-related measures measure the distribution of influence in the community. The patch data includes also *author* and *committer* information. Applying the measures of this paper to data mined with respect to former would measure similar aspects from the actual programmers, and the latter would give a different view to influence. These two types of measurement are left as future work.

We have formulated three hypotheses which have not been tested yet: By taking a more detailed look at the patches and programmers' e-mail taxonomies, we could study whether there are differences in development based on the types of organizations. For example, one could make an educated guess that companies contribute more on driver development whereas universities are interested in more abstract problems.

An interesting viewpoint is, whether there are differences which correlate to geography. For example, based on prejudice, one might guess that from Germany companies participate in the Linux Kernel community,

whereas French participate as individuals, and United States' contribution is from companies and universities.

Prior to a publications of a product, Linux-related companies naturally develop the Linux Kernel. For example, a company manufacturing servers, might be active in developing Linux just before a new computer hardware is introduced to market. This activity is visible from our data. Therefore, economical indicators might be possible to develop based on the fact that activity leads to release, which in turn leads to rise of the value of the company. Therefore, activity in developing Linux leads to rise of the value. Studying the hypotheses is left as future work.

## References

- [1] Netcraft, "Web server survey." [http://news.netcraft.com/archives/web\\_server\\_survey.html/](http://news.netcraft.com/archives/web_server_survey.html/), 2006.
- [2] R. McMillan, "Mozilla gains on IE," *PC World*, 2004.
- [3] "Linux online." <http://linux.org>, 2006.
- [4] L. Torvalds and J. C. Hamano, "GIT - fast version control system." <http://git.or.cz>, 2006.
- [5] "Kernel hackers' guide to git." <http://linux.yyz.us/git-howto.html>, 2005.
- [6] L. Torvalds, "Linux: Documenting how patches reach the kernel." <http://kerneltrap.org/node/3180>, 2004.
- [7] T. G. Project, "Gnome: The free software desktop project." <http://www.gnome.org/>, 2006.
- [8] T. Aaltonen, J. Järvenpää, and T. Mikkonen, "Oss architecture and implications," tech. rep., eBRC, 2006.
- [9] "Wikipedia, the free encyclopedia." [http://en.wikipedia.org/wiki/Main\\_Page/](http://en.wikipedia.org/wiki/Main_Page/), 2006.
- [10] K. Nakakoji, Y. Yamamoto, Y. Nishinaka, K. Kishida, and Y. Ye, "Evolution patterns of open-source software systems and communities," in *Proceedings of the International Workshop on Principles of Software Evolution(IWPSE2002)*, pp. 76–85, ACM Press, 2002.
- [11] "Kernelnewbies wiki." <http://kernelnewbies.org/>, 2006.
- [12] D. Weiss, "Quantitative analysis of open source projects on sourceforge," in *Proceeding of the First International Conference on Open Source Systems*, 2005.
- [13] I. T. Gregorio Robles, Hendrik Scheider and N. Weber, "Who is doing it? - a research on libre software developers," 2001.