UPGRADE
The European Journal for the Informatics Professional
http://www.upgrade-cepis.org
Vol. VI, issue No. 3, June 2005

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

UPGRADE is the anchor point for **UP**ENET (**UP**GRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
• **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
• **Novática**, journal from the Spanish CEPIS society ATI
• **OCG Journal**, journal from the Austrian CEPIS society OCG
• **Pliroforiki**, journal from the Cyprus CEPIS society CCS
• **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by **Novática** <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**, and in Italian (summary, abstracts and some articles online) by the Italian CEPIS society ALSI (*Associazione nazionale Laureati in Scienze dell'informazione e Informatica*, <http://www.alsi.it>) and the Italian IT portal **Tecnoteca** <http://www.tecnoteca.it/>

UPGRADE was created in October 2000 by CEPIS and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

ISSN 1684-5285

Monograph of next issue (August 2005):
"Normalisation & Standardisation in IT Security"
(The full schedule of **UP**GRADE is available at our website)

Monograph: *Libre* Software as A Field of Study
(published jointly with Novática*, in cooperation with the European project CALIBRE)
Guest Editors: *Jesús M. González-Barahona and Stefan Koch*

MOSAIC

UPENET (**UP**GRADE European NETwork)

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>, and in Italian (online edition only, containing summary, abstracts, and some articles) by the Italian CEPIS society ALSI (*Associazione nazionale Laureati in Scienze dell'informazione e Informatica*) and the Italian IT portal Tecnoteca at <http://www.tecnoteca.it>.

Measuring *Libre* Software Using Debian 3.1 (Sarge) as A Case Study: Preliminary Results

Juan-José Amor-Iglesias, Jesús M. González-Barahona, Gregorio Robles-Martínez, and Israel Herráiz-Tabernero

The Debian operating system is one of the most popular GNU/Linux distributions, not only among end users but also as a basis for other systems. Besides being popular, it is also one of the largest software compilations and thus a good starting point from which to analyse the current state of libre *(free, open source) software. This work is a preliminary study of the new Debian GNU/Linux release (3.1, codenamed Sarge) which was officially announced recently. In it we show the size of Debian in terms of lines of code (close to 230 million source lines of code), the use of the various programming languages in which the software has been written, and the size of the packages included within the distribution. We also apply a 'classical' and well-known cost estimation method which gives an idea of how much it would cost to create something on the scale of Debian from scratch (over 8 billion USD).*

Keywords: COCOMO, Debian, *Libre* Software, *Libre* Software Engineering, Lines of Code, Linux.

1 Introduction

On June 6, 2005, the Debian Project announced the official release of the Debian GNU/Linux version 3.1, codenamed "Sarge", after almost three years of development [6]. The Debian distribution is produced by the Debian project, a group of nearly 1,400 volunteers (a.k.a. maintainers) whose main task is to adapt and package all the software included in the distribution [11]. Debian maintainers package software which they obtain from the original (upstream) authors, ensuring that it works smoothly with the rest of the programs in the Debian system. To ensure this, there is a set of rules that a package should comply with, known as the Debian Policy Manual [5].

Debian 3.1 includes all the major *libre* software packages available at the time of its release. In its main distribution alone, composed entirely of *libre* software (according to Debian Free Software Guidelines), there are more than 8,600 source packages. The whole release comprises almost 15,300 binary packages, which users can install easily from various media or via the Internet.

In this paper we analyse the system, showing its size and comparing it to other contemporary GNU/Linux systems[1]. We decided to write this paper as an update of *Counting Potatoes* (see [8]), and *Measuring Woody* (see [1]) which were prompted by previous Debian releases. The paper is structured as follows. The first section briefly presents the methods we used for collecting the data used in this paper. Later, we present the results of our Debian 3.1 count (including total counts, counts by language, counts for the largest packages, etc.). The following section provides some comments on these figures and how they should be interpreted and some comparisons with Red Hat Linux distributions and other free and proprietary operating systems. We close with some conclusions and references.

2 Collecting The Data

In this work we have considered only the **main** distribution, which is the most important and by far the largest part of any Debian release. It is composed exclusively of free software (according to Debian Free Software Guidelines, DFSG [7]). Other sections, such as **non-free** or

[1] GNU/Linux systems are also known as 'distributions'.

Juan-José Amor-Iglesias has an MSc in Computer Science from the *Universidad Politecnica de Madrid*, Spain, and he is currently pursuing a PhD at *the Universidad Rey Juan Carlos* in Madrid, Spain. Since 1995 he has collaborated in several free software related organizations: he is a a co-founder of LuCAS, the best known free documentation portal in Spanish, and Hispalinux, and collaborates with Barrapunto.com. <jjamor@gsyc.escet.urjc.es>

Jesús M. González-Barahona teaches and researches at the *Universidad Rey Juan Carlos*, Madrid, Spain. He started working in the promotion of *libre* software in the early 1990s. Since then he has been involved in several activities in this area, such as the organization of seminars and courses, and the participation in working groups on *libre* software. He currently collaborates in several *libre* software projects (including Debian), and participates in or collaborates with associations related to *libre* software. He writes in several media about topics related to *libre* software, and consults for companies on matters related to their strategy regarding these issues. His research interests include *libre* software engineering and, in particular, quantitative measures of *libre* software development and distributed tools for collaboration in *libre* software projects. He is editor of the Free Software section of **Novática** since 1997 and has been guest editor of several monographs of **Novática** and **UP**GRADE on the subject. <jgb@gsyc.escet.urjc.es>

Gregorio Robles-Martínez is a PhD candidate at the *Universidad Rey Juan Carlos* in Madrid, Spain. His main research interest lies in *libre* software engineering, focusing on acquiring knowledge of *libre* software and its development through the study of quantitative data. He was formerly involved in the FLOSS project and now participates in the CALIBRE coordinated action and the FLOSSWorld project, all European Commission IST-program sponsored projects. <grex@gsyc.escet.urjc.es>

Israel Herráiz-Tabernero has an MSc in Chemical and Mechanical Engineering, a BSc in Chemical Engineering and he is currently pursuing his PhD in Computer Science at the *Universidad Rey Juan Carlos* in Madrid, Spain. He 'discovered' free software in 2000, and has since developed several free tools for chemical engineering. <herraiz@gsyc.escet.urjc.es>

contrib, are not covered here. The approach used for collecting the data is as follows: first, the sources for the distribution are retrieved from the public archives on the Internet, through archive.debian.org <ftp://archive.debian.org> and its mirrors, on a per-package basis. Debian provides source code packages and binary packages. We have used the former in this study, although the latter are what tend to be downloaded by users as they are pre-compiled. For each source code package there may be one or many binary packages.

Our second step was to analyse the packages and extract the information that we were looking for using **SLOCCount**[2] [12]. The lines of code count is only an estimate due to some peculiarities of the tool (basically based on source code and programming language identification heuristics) and the criteria chosen for the selection of packages [8].

The final step was to identify and remove packages that appear several times in different versions (for instance, this happens with the GCC compiler) so as not to count the same code more than once. This may lead to an underestimation as in some cases the source code base may not be that similar (in the case of PHP, we have left the PHP4 version but removed PHP3), so we have kept some cases where we know that at least significant amounts of common code (for instance for xemacs and emacs or for gcc and gnat) are present. The final step is to draw up a set of reports and statistical analyses using the data gathered in the previous step and considering it from various points of view. These results are presented in the following section.

3 Results of Debian 3.1 Count

After applying the methodology described we calculated that the total source lines of code count for Debian 3.1 is *229,496,000* SLOC (Source Lines Of Code). Results by category are presented in the following subsections (all numbers are approximate, see [4] for details).

Language	Source Lines of Code (SLOC)	%
C	*130,847,000*	57
C++	*38,602,000*	16.8
Shell	*20,763,000*	9
LISP	*6,919,000*	3
Perl	*6,415,000*	2.8
Python	*4,129,000*	1.8
Java	*3,679,000*	1.6
FORTRAN	*2,724,000*	1.2
PHP	*2,144,000*	0.93
Pascal	*1,423,000*	0.62
Ada	*1,401,000*	0.61
TOTALS	***229,496,000***	**100**

Table 1: Count of Source Lines of Code by Programming Language in Debian 3.1.

3.1 Programming Languages

The number of physical SLOC and percentages, broken down by programming language, are shown in **Table 1**.

Below 0.5% there are some other languages such as Objective C (0.37%), ML (0.31%), Yacc (0.29%), Ruby (0.26%), C# (0.23%) or Lex (0.10%). A number of other languages score less than 0.1%.

The pie chart in Figure 1 shows the relative importance of the main languages in the distribution. Most Debian packages are written in C, but C++ is also to be found in many packages, being the main language in some of the most important ones (such as OpenOffice.org or Mozilla). Next up comes Shell, which is mainly used by scripts supporting configuration and other auxiliary tasks in most packages. Surprisingly LISP is one of the top languages, although this can be explained by the fact that it is the main language in several packages (such as emacs) and is used in many others. While this is not reflected in our results, there is a historical trend towards a relative decline of the C programming language combined with a growing importance of more modern languages such as Java, PHP, and Python.

3.2 Largest Packages

The following list shows the most important Debian 3.1. packages over 2 MSLOC broken down by size. For each package we give the package name, version, total number of SLOC, composition of programming languages, and a description of the purpose of the software.

- OpenOffice.org (1.1.3): 5,181,000 SLOC. C++ accounts for 3,547,000 SLOC. C accounts for 1,040,000 SLOC. There is also code written in 15 more languages, either scripting languages (such as shell, tcl, python or awk) or non-scripting languages (pascal, java, objective-C, lisp, etc).
- Linux kernel (2.6.8): 4,043,000 SLOC. C accounts for 3,794,000 SLOC, Makefiles, assembler and scripts in several languages accounts for the rest. This is the latest kernel included in the Debian 3.1 release.
- NVU (N-View) (0.80): 2,480,000 SLOC. Most of the code is C++, with more than 1,606,000 SLOC, plus a large percentage of C (798,000 SLOC). Other languages, mainly scripting languages, are also used. It is a complete web authoring system capable of rivalling well known proprietary solutions such as Microsoft FrontPage.
- Mozilla (1.7.7): 2,437,000 SLOC. Most of its code is C++, with more than 1,567,000 SLOC plus a large percentage of C (789,000 SLOC). Mozilla is a well known open source Internet suite (WWW browser, mail client, etc).
- GCC-3.4 (3.4.3): 2,422,000 SLOC. C accounts for 1,031,000 SLOC, Ada for 485,000 SLOC and C++ for 244,000 SLOC. Other languages are used minimally. GCC is the popular GNU Compiler Collection.
- XFS-XTT (1.4.1): 2,347,000 SLOC. Mainly 2,193,000 SLOC of C. Provides an X-TrueType font server.
- XFree86 (4.3.0): 2,316,000 SLOC. Mainly 2,177,000 SLOC of C. An X Window implementation, including a graphics

[2] We use SLOCCount revision 2.26. It currently recognizes 27 programming languages.

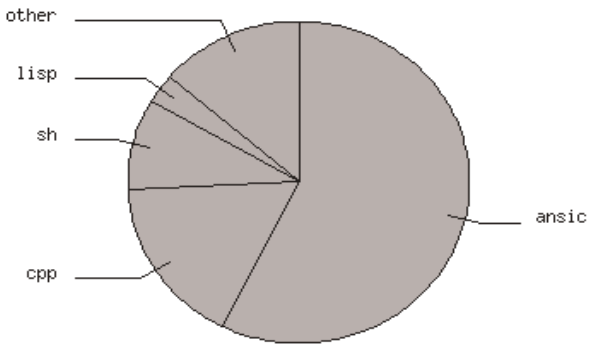

Figure 1: Breakdown of Source Lines of Code for The Predominant Languages in Debian 3.1.

server and basic programs.

- VNC4 (4.0): 2,055,000 SLOC. VNC4 is a remote console access system, mainly programmed in C with 1,920,000 SLOC.
- Insight (6.1): 1,690,000 SLOC, mainly programmed in C (1,445,000 SLOC). Insight is a graphical debugger based on GDB.
- kfreeBSD5-source (5.3): 1,630,000 SLOC. This is the source code of 5.3-FreeBSD kernel, a base for a future GNU distribution based on FreeBSD kernel.

It should be noted that this list would have varied if Debian maintainers had packaged things following different criteria. For instance, if all emacs extensions had been included in the emacs package it would have been much further up the table (probably in the "top ten" list). However, a Debian source package tends to be very much in line with what upstream authors consider to be a package, which is usually based on software modularization principles.

Figure 2 provides a breakdown of the sizes of all Debian 3.1 packages. Throughout our study of Debian distributions over time, from version 2.0 (released in 1998) to version 3.0 (released in 2002), we have observed that the mean size of packages is around 23,000 lines [10]. For Debian 3.1 the mean size of packages has increased to 26,600 lines. The reason behind this is not yet clear, and further studies need to be conducted, but it may be because the number of packages is growing faster than the number of maintainers, so that the previous equilibrium no longer exists.

3.3 Effort and Cost Estimations

The COCOMO model (COnstructive COst MOdel) [2] provides a rough estimation of the human and monetary effort needed to generate software of a given size. It takes as an input metric the number of source lines of code. Since this estimation technique is designed for 'classical' software generation processes and for large projects, the results it gives when applied to Debian packages should be viewed with caution. In any case, we will use a basic COCOMO model to give us an effort estimation based in its size. Using the SLOC count for the Debian source packages, the data provided by the basic COCOMO model are as follows:

- Total physical SLOC count: 229,495,824
- Estimated effort: 714,440.52 person-months (59,536.71 person-years). Formula: *2.4* * (KSLOC^1.05)
- Estimated schedule: 105.84 months (8.82 years). Formula: 2.5 * (Effort^0.38)
- Estimated cost to develop: 8,043,000,000 USD
- To reach these figures, each project was estimated as though it had been developed independently, which is true for nearly all cases. For calculating the cost estimation, we have used the mean salary for a full-time systems programmer in 2000 according to Computer World [3] - 56,286 USD per year - and an overhead factor of 2.4 (for an explanation of how this factor is arrived at and other details of the estimation model see [13]).

4 Comparison with Other Systems

To put the figures shown above into context, here are some software sizes for operating systems. The figures that appear in Table 2 have been obtained from several different sources (listed in [10]) and refer to approximate lines of code.

Most of these numbers (in fact, all of them, except for Red Hat Linux, Fedora Core and Debian) are estimates as it even difficult to know what they consider as a line of code (i.e. whether they take into account comments and blank lines or not). However, for the sake of this paper they provide enough insight and hence we consider them suitable for comparison purposes.

It should also be noted that, while Red Hat and Debian include a great many applications and, in many cases, even several applications within the same category, Microsoft and Sun operating systems include only a limited number of them (which also tend to be small in size). If the most common applications used in those environments were to be included, they would be far larger. However, it is also true that all those applications are neither developed nor put together by the same team of maintainers, as is the case of Linux-based distributions.

From these numbers, it can be seen that Linux-based

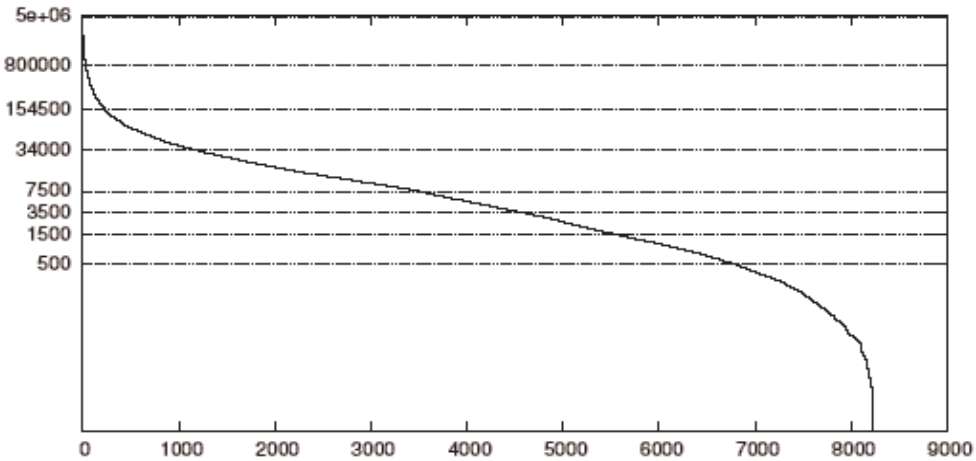

Figure 2: Package Sizes for Debian 3.1. Counts in SLOCs Are Represented on A Logarithmic Scale.

distributions in general, and Debian 3.1 in particular, are some of the largest pieces of software ever put together by a group of maintainers.

5 Conclusions and Related Work

Debian is one of the largest software systems in the world, probably the largest. Its size has grown with every release, 3.1 being twice the size of 3.0. For the last few releases, the main languages used to develop packages included in Debian are C and C++. In fact C, C++ and Shell represent more than 75% of all source code in Debian. The number of packages continues to grow steadily, doubling almost every two years.

The Debian GNU/Linux distribution, put together by a group of volunteers dispersed all over the world, would, at first sight, appear to show a healthy and fast-growing trend. Despite its enormous size it continues to deliver stable releases. However, there are some aspects that put into doubt the future sustainability of this progress. For instance, mean package size is showing an unstable behaviour, probably due to the number of packages growing faster than the number of maintainers. Nor can we forget that we have had to wait almost three years for a new stable release and that the release date has been seriously delayed on several occasions.

Regarding other software systems, there are few detailed studies of the size of modern, complete operating systems. The work by David A. Wheeler, counting the size of Red Hat 6.2 and Red Hat 7.1 is perhaps the most comparable. Some other references provide total counts of some Sun and Microsoft operating systems, but while they do provide estimates for the system as a whole, they are not detailed enough. Debian is by far the largest of them, although this comparison has to be taken with a degree of caution.

To conclude, it is important to stress that this paper aims to provide estimations based only on a preliminary study (since the release is not yet officially published). However, we believe they are accurate enough to allow us to draw some conclusions and compare them with other systems.

Operating System	Source Lines of Code (SLOC)
Microsoft Windows 3.1 (April 1992)	*3,000,000*
Sun Solaris (October 1998)	*7,500,000*
Microsoft Windows 95 (August 1995)	*15,000,000*
Red Hat Linux 6.2 (March 2000)	*17,000,000*
Microsoft Windows 2000 (February 2000)	*29,000,000*
Red Hat Linux 7.1 (April 2001)	*30,000,000*
Microsoft Windows XP (2002)	*40,000,000*
Red Hat Linux 8.0 (September 2002)	*50,000,000*
Fedora Core 4 (previous version; May 2005)	*76,000,000*
Debian 3.0 (July 2002)	*105,000,000*
Debian 3.1 (June 2005)	*229,500,000*

Table 2: Size Comparison of Several Operating Systems.

Acknowledgements

This work has been funded in part by the European Commission, under the CALIBRE CA, IST program, contract number 004337, in part by the *Universidad Rey Juan Carlos* under project PPR-2004-42, and in part by the Spanish CICyT under project TIN2004-07296.

References

[1] Juan José Amor, Gregorio Robles y Jesús M. González-Barahona. *Measuring Woody: The size of Debian 3.0* (pending publication). Will be available at <http://people.debian.org/~jgb/debian-counting/>.

[2] Barry W. Boehm. *Software Engineering Economics*, Prentice Hall, 1981.

[3] Computer World, *Salary Survey 2000*. <http://www.computerworld.com/cwi/careers/surveysandreports>.

[4] Jesús M. González Barahona, Gregorio Robles, and Juan José Amor, *Debian Counting*. <http://libresoft.urjc.es/debian-counting/>.

[5] Debian Project, *Debian Policy Manual*. <http://www. debian. org/doc/debian-policy/>.

[6] Debian Project, *Debian GNU/Linux 3.1 released (June 6th 2005)*. <http://www.debian.org/News/2005/20050606>.

[7] Debian Project, *Debian Free Software Guidelines (part of the Debian Social Contract)*. <http://www.debian.org/social_contract>.

[8] Jesús M. González Barahona, Miguel A. Ortuño Pérez, Pedro de las Heras Quirós, José Centeno González, and Vicente Matellán Olivera. *Counting potatoes: The size of Debian 2.2*. UPGRADE, vol. 2, issue 6, December 2001, <http://upgrade-cepis.org/issues/2001/6/up2-6Gonzalez. pdf>; Novática, nº 151 (nov.-dic. 2001), <http://www.ati.es/novatica/2001/154/154-30.pdf> (in Spanish).

[9] Jesús M. González-Barahona, Gregorio Robles, Miguel Ortuño-Pérez, Luis Rodero-Merino, José Centeno-González, Vicente Matellán-Olivera, Eva Castro-Barbero, and Pedro de-las-Heras-Quirós. *Anatomy of two GNU/Linux distributions*. Chapter in book "Free/Open Source Software Development" edited by Stefan Koch and published by Idea Group, Inc., 2004.

[10] Gregorio Robles, Jesús M. González-Barahona, Luis López, and Juan José Amor, *Toy Story: an analysis of the evolution of Debian GNU/Linux*, November 2004 (pending publication). Draft available at <http://libresoft.urjc.es/debian-counting/>.

[11] Gregorio Robles, Jesús M. González-Barahona, and Martin Michlmayr. *Evolution of Volunteer Participation in Libre Software Projects: Evidence from Debian*, julio 2005, Proceedings of the First International Conference on Open Source Systems. Genova, Italy, pp. 100-107. <http://gsyc.escet.urjc.es/~grex/volunteers-robles-jgb-martin. pdf>.

[12] David Wheeler. *SLOCCount*. <http://www.dwheeler.com/sloccount/>.

[13] David A. Wheeler. *More Than a Gigabuck: Estimating GNU/Linux's Size*. <http://www.dwheeler.com/sloc>.

