Proposal for Publishing

Introduction to Operating Systems Abstractions Using Plan 9 from Bell Labs

Francisco J Ballesteros

Description

Teaching Operating Systems is complex, because it is a complex and wide area. After many years of teaching, I have found that the only way that works is to

1 Teach how to **use** an operating system, while introducing the abstractions provided by a particular operating system.

2 Teach the theory (and practice) behind the interface for the abstractions. That is, what is covered in most operating systems courses.

3 Teach how an actual system is implemented.

The problem is that most books regarding this subject fall into one of two classes. They are either very practical books about *how to use* a system, or they are very theoretical books for teaching theory. But Operating Systems is a practical subject! Some books try to do it all at once (Like Tanembaum's book describing Minix). In my experience, this just does not work. At least, it does not for some introductory courses I teach. To make an analogy, it is best to learn first how to drive a car, and then it might be of interest how the car actually works.

The proposed book fills a huge gap, that between how-tos and existing operating systems textbooks. I was very reticent to write this book, but evidence has just forced me to do so.

The book is intended for a **first semester on Operating Systems**. This is usually followed by a second semester where a more classical approach including all relevant theory can be applied.

This work is being done to be used in our first semester in Operating Systems at Rey Juan Carlos University. It describes, for someone who has (basic) programming skills, how to effectively use a system. At the same time, it introduces most of the system abstractions to the student, for the first time.

An important point is that the system described is an excellent one. It is very clean, while being more powerful than other systems. It was implemented by people very respected in the field. The book describes system interfaces using Plan 9 from Bell Labs, the system made by the same group that made UNIX (and C) in the 70s. Using other systems to teach operating systems, like for example Linux or Windows, makes the students focus on particular system details (that are likely to change really soon) instead of learning the concepts and acquiring the skill of effectively using a system. In any case, the system used in the book is close enough to UNIX to let students use Linux effectively really soon after completing the course.

People that got access to the draft have reported that it is outstanding. Indeed, I know that it is being recommended by some group working at IBM's research center in Austin, and by a professor at CMU that had access to the draft as well. Few peopled had access to the text, but those who had found it very useful.

Outstanding features

- This work describes a system built by some of the most reputed systems programmers. With clean interfaces. Excellent for teaching purposes, and still very useful to learn how to use a system for the first time.
- By reading the book, the reader can learn how to effectively use an entire system even if he/she just

followed some programming courses.

- The book introduces theory in a very practical way, introducing problems (solved through the book) that force the student to require more concepts and tools to solve them. It focuses on the need to motivate students so they really want to learn more to solve their problems.
- The book covers topics really necessary, but usually forgotten in operating systems texts. For example, graphical user interfaces.
- As far as I know, this is the only book with a description of a *modern* system. Most other texts focus on technology that (despite the hype) comes actually from the 70s. At least, the book prepares the students for things to come, which is what college is for in the end.
- Many ideas described in the book haven been applied to Linux and other systems as advanced features. Therefore, although introductory, the book provides very useful insights not usually covered on other texts.
- The system used in the book is open software, and can be download and used for free.
- The system used really can be understood by a single person. That does not happens anymore with UNIX (e.g., Linux) or Windows, which are the systems of choice for most other books.. This is very frustrating for students (and professors) that still believe in magic after following a systems course.
- There is another book available from the author that describes, line by line, how the system implementation works. So, there is potential for continuing using the same system in following courses. This book has been a great success and has been used by most people in the Plan 9 community. All the students at Rey Juan Carlos University (and other centers) are using it. It is available at `http://lsub.org/who/nemo/9.pdf.gz`.
- Last but not least, the book is easy to read. This is very hard to achieve for systems books, but it seems to be the case according to people with access to the draft.

Competition

Theoretical OS books. Most of books on Operating Systems made to be used as textbooks can be found here. There is no need to cite them. Their aim is to introduce the reader to theoretical OS concepts. They are really useful in that a newcomer to the field is better exposed to OS issues by reading first one of them. However, readers still do not know how all this applies to applications they build, and to systems they use. Despite the examples. It is hard to teach how a car works to anyone who does not drive.

Internals books. Books describing internals of Operating Systems, describe the main design guidelines of operating systems in-use. I consider these books intermeditate between the classical textbooks and books describing an implementation. Such books are for more advanced courses (or more knowledgeable readers) and are not a real competition for the proposed book.

How-Tos. Books like those from O'Really are excellent to get a quick start for particular tools. However, they are not well suited for teaching, and they do not expose the reader to the systems abstractions and classic concepts about how to use a system. Not to talk about the systems they use. Most of them are more than 30 years old now, and not very appropriate for a University course.

Pedagogical elements

Probably the main point of the book is that all the book is a big example of how a system is used. The reader has available several versions of the system described and can use it to see how things work. Furthermore, the system is close enough to Linux to allow the student to explore his/her Linux system.

The book has been built according to the experience of the author. He needed a book to introduce operating systems concepts (by showing what they do, and how they are used and applied), but sadly, he could not find any such book.

The book is centered on solving practical problems, motivating students to learn concepts and tactics for effectively using a system.

There is more documentation and some other tools available from the author, that we are using in our

lectures.

In few words, **this is just the book for getting introduced to operating systems.**

Audience/market

The intended audience is made of students of introductory operating systems courses, as well as of people who want to learn how a system is used. In this respect, the audience is also made of people wanting to use good interfaces done by well-reputed programmers like the ones who built UNIX, C, and Plan 9.

There is a growing and widespread interest in Linux and Unix like systems and operating systems in general. More and more people are interested in how to use effectively an operating system. Unix systems have grown to be too complex to be understood by just one person.

Therefore, audience would include anyone willing to learn how a full fledge operating system works. This includes hobbyists, professional programmers and and CS an CE students. All this is specially true for people interested in Unix since the system it is based on one which was written by the people who designed and implemented it after realizing their mistakes.

After reading the book, the reader will learn:

1 What a system is

2 How is it used.

3 How to automate tasks, to let the machine do the job.

4 Why most system interfaces have particular functionalities.

5 How complex things can be done in a (more) simple way.

The requisite for understanding the book is a to know how to program. It would help to have knowledge about basic computer architecture concepts.

Promotion

The main user group useful to promote the book is the community of Plan 9 users, which can be reached through `9fans@cse.psu.edu`.

In any case, all Linux user communities will be interested on this book, because after all, it describes the system built after considering that UNIX (hence Linux) had serious problems regaring networks and other technologies. Most (if not all!) knowledge gained by reading the book will be very useful for using a UNIX system.

The book should be shelved in the ‘‘Operating Systems’’ bookstore category.

Support

The early draft of the book is going to be used during the next semester course at Rey Juan Carlos University of Madrid. Supplementary material may be made available on its web site. See `http://lsub.org/cursos`.

A second book, and the implementation of the Plan 9 system is the main supplementary material. All the source code for Plan 9 applications and commands is easy to read and understand after reading the book. There is much to learn by doing so.

It is important that the code is available from the internet, at `http://plan9.bell-labs.com`. It can be downloaded for free by people following the book, and this is important.

Status

An initial draft has been completed. It certainly needs reviewing and more work but its current state is good enough to be used in a real OS course.

The book is about 390 pages long on A4 paper, and has been formatted using tools from the system described.

The book itself does not include copyrighted material, as far as I know.

Reviewers

Authors of the system can be invaluable reviewers. In fact, Rob Pike and other authors of the system might be willing to revise the manuscript. All of them had access to the draft.

Table of Contents and Sample chapters

The current version of the draft can be downloaded only for evaluation purposes from `http://lsub.org/who/nemo/9.intro.pdf`. Please, do not distribute more than needed to review and evaluate.

Curriculum Vitae Excerpt

Prof. Francisco J. Ballesteros (http://lsub.org/who/nemo) got his MS in CS on 1993 and his PhD on CS on 1998, at Technical University of Madrid. While an undergraduate, he got several grants from European research projects where he developed run-time software for programming languages. Later, he worked for several years on telecommunications companies, doing systems software (He is the (co)author of LiS, a STREAMS framework for Linux.) Since 1995 he has been a professor at several Spanish Universities where he has been teaching and developing Operating Systems. He developed the Off++ kernel, for the 2K Operating System jointly with the SRG at University of Illinois at Urbana Champaign. 2K evolved later into the Gaia OS. He has been working in R&D on both the Plan 9 from Bell Labs and the Plan B Operating Systems. He is the head of the Systems Lab (http://lsub.org) where Plan B is being developed. At present, he is teaching five lectures on Operating Systems, and doing research in the field. A list of publications can be found at the web page.

ESCET, Universidad Rey Juan carlos. C/ Tulipan s/n E-28933 Mostoles (Madrid, Spain).

Phone: +34-91-664-7469 (work) +34-91-684-4123 (home)

Mobile: +34-66-770-1004

Mail: nemo@lsub.org