

GSDLAB TECHNICAL REPORT

# Configuration Challenges in Linux and eCos: A Survey

Arnaud Hubaux, Yingfei Xiong, Krzysztof Czarnecki

GSDLAB-TR 2011-09-29

September 2011



Generative Software  
Development Lab



Generative Software Development Laboratory  
University of Waterloo  
200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1

**WWW page:** <http://gsd.uwaterloo.ca/>

The GSDLAB technical reports are published as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

# Configuration Challenges in Linux and eCos: A Survey

Arnaud Hubaux, Yingfei Xiong, Krzysztof Czarnecki

September 29, 2011

## Abstract

Operating systems expose sophisticated configurability to handle variations in hardware platforms like desktops, servers, and mobile devices. The configuration of an operating system like Linux contains thousands of options guarded by hundreds of complex constraints. To guide users throughout the configuration activity, configurators implement various mechanisms to produce correct configurations. However, configuration still remains a difficult and challenging process. To better understand the challenges faced by users during configuration, we conducted surveys among Linux and eCos users to answer the two following questions:

- What challenges do users most frequently face?
- How significant is the conflict resolution problem?

In this paper, we report on the results of these two surveys.

## 1 Linux

This section focuses on the configuration tools used for the Linux kernel. We investigate the Linux kernel because it is probably the configurable system with the largest community of users and developers, and it spans a wide variety of configuration domains.

To harvest data, we designed a questionnaire (see <https://docs.google.com/spreadsheets/viewform?formkey=dHJrQ1B6OXVCRTRmbGJ2Z1FQTmdpQVE6MA>) that was posted on several forums and mailing lists of mainstream Linux distributions and the Linux Kernel:

- Gentoo: <http://www.gentoo.org/>
- Debian: <http://forums.debian.net/>
- Ubuntu: <http://ubuntuforums.org/>
- Linux Questions: <http://www.linuxquestions.org/>
- openSUSE: <http://en.opensuse.org/>, <http://lists.opensuse.org/>  
[opensuse-factory](http://lists.opensuse.org/opensuse-kernel), <http://lists.opensuse.org/opensuse-kernel>

- Fedora: <http://fedoraforum.org/>
- Mandriva: <http://www.mandriva.com/>
- LinuxMint: <http://www.linuxmint.com/>
- pclinuxos: <http://www.pclinuxos.com/forum/>
- centOS: <https://www.centos.org/>
- archlinux: [arch-general@archlinux.org](mailto:arch-general@archlinux.org)
- Kernel: <http://vger.kernel.org/>

This document contains a summary of 97 answers to the questionnaire, and discussions on a Gentoo forum and the opensuse-factory mailing list triggered by our questionnaire. This summary starts with a quantitative analysis that presents the profiles of the respondents and their configuration practice. Then, a qualitative analysis reports examples in which the activation of an option posed problems, and outlines 6 classes of configuration challenges expressed by users.

## 1.1 Quantitative analysis

The quantitative analysis is based on 81 answers to the questionnaire. 16 answers were discarded either because the responses to some questions were inappropriate or too vague to conduct quantitative analyses. The questions addressed by each point are recalled before the discussion on the results.

### 1.1.1 Respondent profiles

#### Level expertise.

*What is your level of expertise in Linux?*

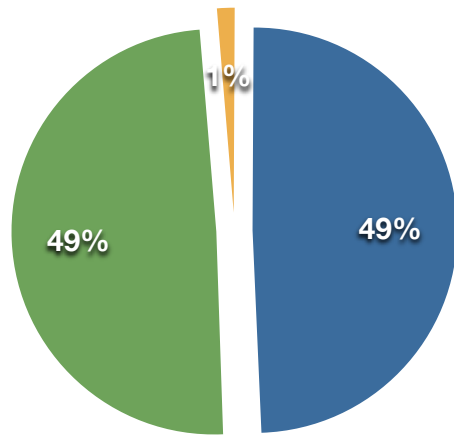
- *Expert*
- *Moderate*
- *Newbie*

According to Figure 1, half of the respondents consider themselves experts, which confirms the evolution of Linux user profiles. Modern distributions now render the configuration of the Linux kernel unnecessary for most average users. Those who still configure the kernel have specific needs that require a significant amount of experience and knowledge.

#### Years of expertise.

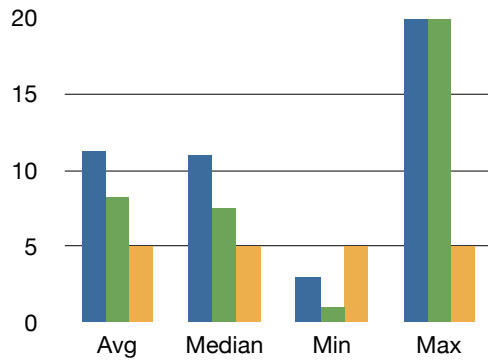
*For how many years have you been using Linux?*

Figure 2 indicates to what extent the level of expertise is connected to the number of years of experience with Linux.



● Expert      ● Moderate      ● Newbie

Figure 1: Distribution of expertise.



■ Expert      ■ Moderate      ■ Newbie

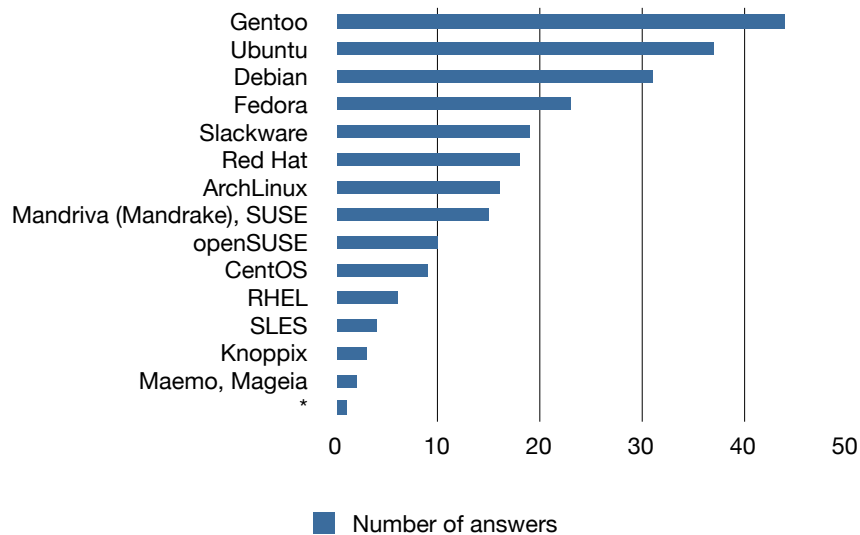
Figure 2: Average, minimum and maximum number of years of expertise per category.

**Linux distributions.**

*What Linux distributions do you have experience with?*

Most respondents had experience with more than one distribution (three on average), hence the large amount of answers reported here. With a total of 39 different distributions, Figure 3 shows a great diversity of back-

grounds in the sample of respondents.



\* Agnula, Android, Aptosid, Asmutils, Buildroot, Chakra, Crunchbang, Damnsmall, DSL, Exherbo, Feather, Linux from scratch, MeeGo, Mint, OpenWRT, PCLinuxOS, Puppy, Sabayon, Salix, Scientific Linux, Slax, Sorcerer, Zenwalk

Figure 3: Ranking of distribution used by the respondents.

## Types of project.

*What types of project have you been configuring the Linux kernel for?*

The neat classification of the types of projects users reconfigure the Linux kernel for turned out to be difficult. First, most users who answered *personal use* frequently gave other options (e.g. server or enterprise). This actually makes perfect sense as many Linux users use it both personally and professionally. Then, the overlap between enterprise and the other categories (e.g. server and embedded systems) is hard to quantify. They have been kept separate in Figure 4 to stay as close as possible to the received answers.

### 1.1.2 Configuration practice

#### Configuration baselines

*What default configuration do you generally use to start the configuration of the kernel?*

- *Default configuration provided by the distribution*

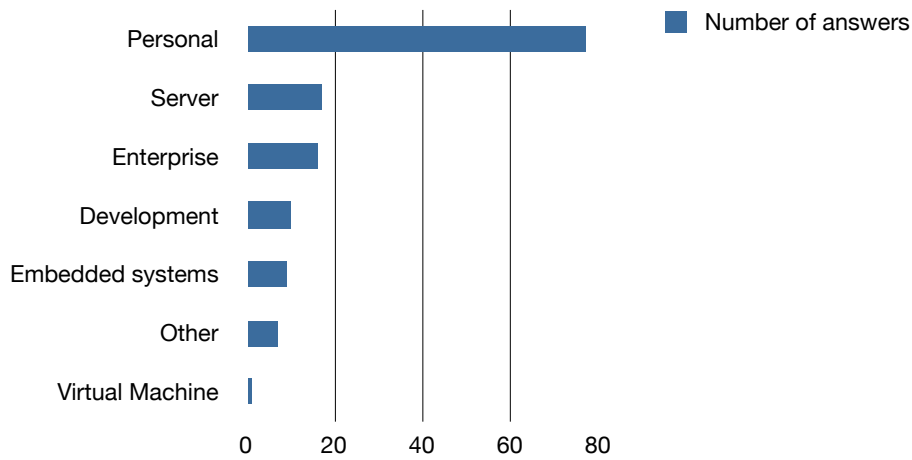


Figure 4: Distribution of usage.

- *Copy of an existing configuration*
- *Configuration available online*
- *Other*

The baseline denotes the default selection of options from which the configuration starts. As Figure 5 shows, more than 90% of the users either copy an existing configuration or use the default configuration provided by their distribution.

#### **Concurrent configuration.**

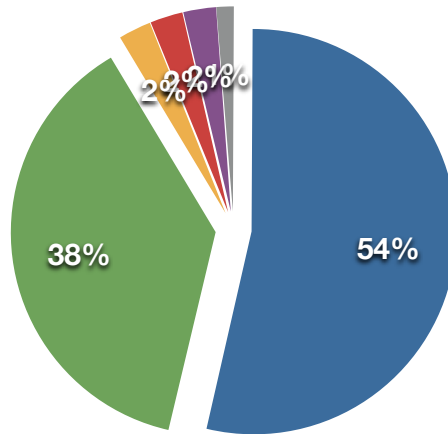
*How many users are usually involved in the configuration activity?*

*If more than one user is involved, are there conflicts between their decisions?*

- *Yes*
- *No*

*If you answered yes to the previous question, how do you handle these conflicts?*

In most cases, the configuration of the Linux kernel is performed by one user (see Figure 6). Approximately 12% of the respondents report that there is more than one user involved. Also, conflict resolution in collaborative configuration is not a problem. If conflicts arise, users talk to each other or the project manager decides which option to select.



- Copy of an existing configuration
- Configuration available online
- defconfig
- Default configuration provided by the distribution
- allnoconfig
- localmodconfig

Figure 5: Distribution of expertise.

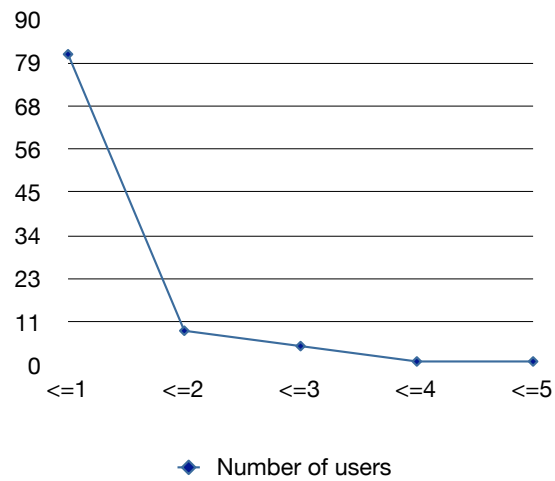


Figure 6: Number of users per configuration.

**Changes and activation of inactive options.**

*How many changes (e.g. enabling an option or setting an attribute value) to the default configuration do you usually make? Please provide a range.*



Some options can sometimes be inactive (hidden or greyed), i.e. they can neither be selected nor deselected. Please estimate the number of inactive options you need to activate per project. Please give a range.

Figure 7 shows how many users (Y-axis) respectively make more than  $x$  changes or activate inactive options (X-axis). Half of the respondents make between 20 and 50 changes while they activate between 0 and 5 inactive options.

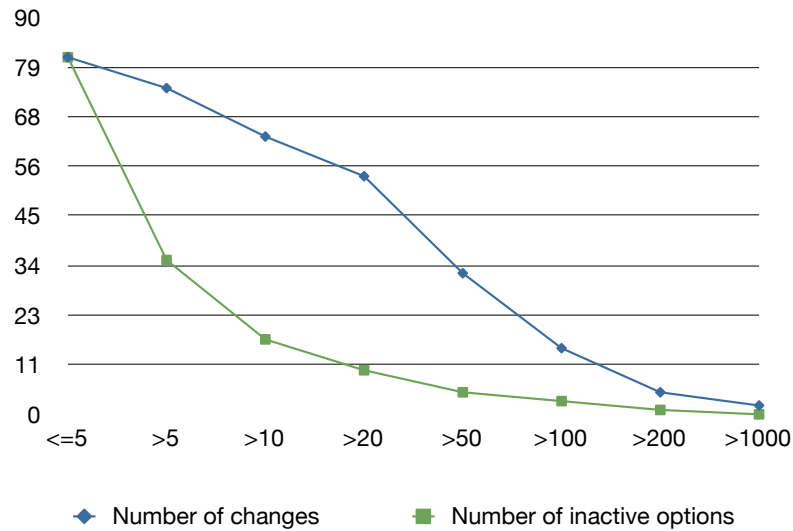


Figure 7: Cumulative number of changes and activations of inactive options.

### Option activation approaches.

*How do you usually proceed to activate an option that is inactive?*

- *I know from experience what other options to enable/disable to activate it*
- *I follow the constraints documented in the help menu*
- *I ask for a solution on a mailing list/forum*
- *Other*

When an inactive option has to be activated, users most often manually follow the sequence of constraints defined in the help menu (see Figure 8). A quarter of the users solely rely on their experience to know which options should be enabled or disabled to activate the inactive option. Here N/A accounts for the few users that trust the configuration provided by the distribution and do not change it.

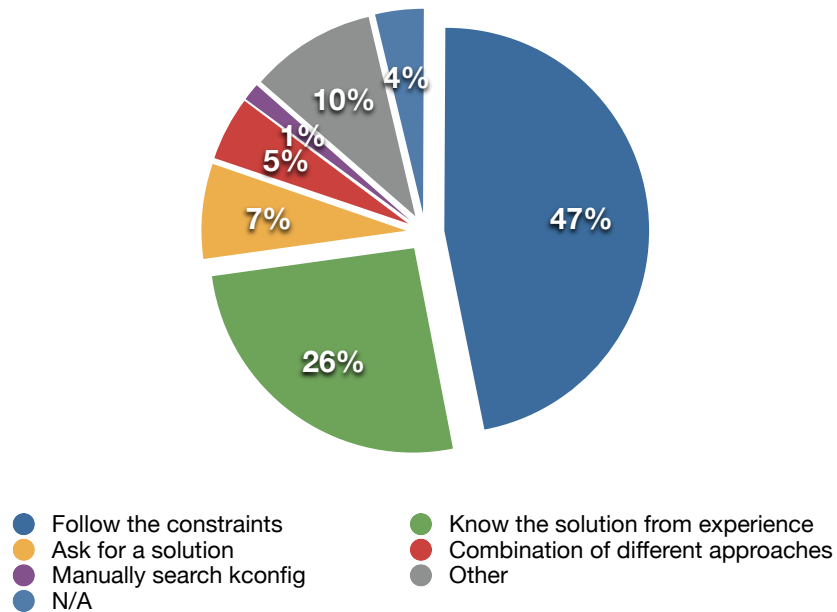


Figure 8: Distribution option activation approaches.

### Resolution time.

*On average, how long does it take to find out how to activate an inactive option?*

- *A few minutes*
- *A few dozen minutes*
- *A few hours*
- *A few days*
- *I do not remember*

Inactive options are activated in a few minutes in most of the cases, as show in Figure 9. Yet, every five decision, the activation of an inactive options requires a few dozen minutes.

### Kernel reconfiguration.

*How often do you reconfigure the Linux kernel?*

Extracting precise numbers is rather complicated as they range from a few times a year to several times a day. The causes are, however, easier to categorize: new install; new kernel release; new patch; new hardware; remove unnecessary options (e.g. for faster boot); development.

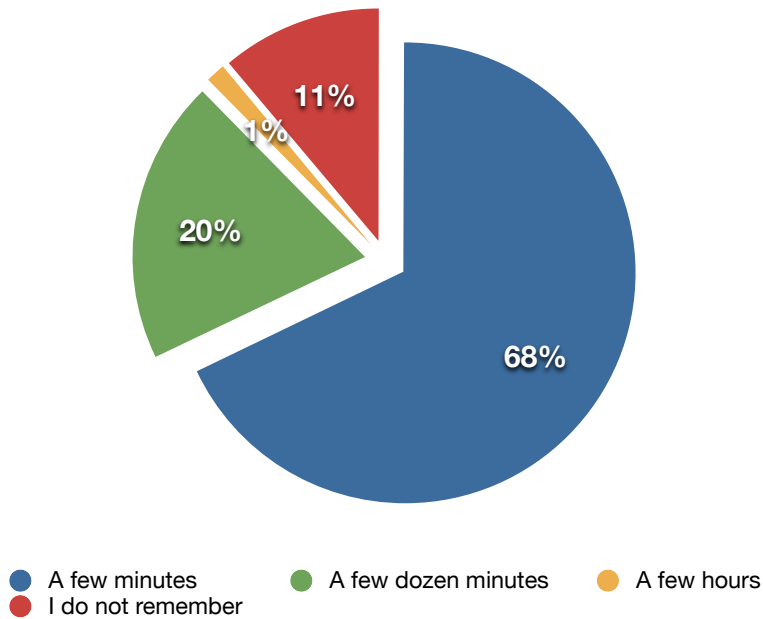


Figure 9: Distribution of the time necessary to activate an inactive option.

## 1.2 Qualitative analysis

The qualitative analysis is based on the complete set of answers (97 in total) and the comments received on the different forums and mailing lists in which our questionnaire was published.

### 1.2.1 Examples of conflicts

In total, we collected 15 examples of problematic option activations. Unfortunately, most of these examples were rather imprecise. We present below those which contain enough details to understand the problem and are most representative. The descriptions are verbatim copies of users' answers modulo minor edits to improve readability.

- The lack of information for the `zcache` option in 2.6.39 caused me to spend some time looking for `cleancache` (as per `zcache`'s help entry). It turned out that some portion of `zcache` (and `cleancache`) had been included in 2.6.39, but `zcache` wasn't actually selectable until 3.0, despite `zcache` appearing via `menuconfig` search.
- I wanted to disable the `framebuffer` option but it was not activated. For the option to appear I had to disable `drm`.
- My kernel version is 2.6.38.8. It is the version that comes with Mageia 1. I wanted to enable the `usbip` module, and had to search through the

whole Device Drivers tree to find it. That should be made easier. Also a `grep -i usblp .config` did not help me.

- To enable the legacy clock I needed to disable RTC. I also had to enable MTD in order to set the device mapper settings
- I was attempting to enable a display panel on an OMAP device with a 2.6.32 kernel. There were several options for display devices, panels, and video outputs. Some hardware was supported by multiple drivers which excluded each other in the kernel configuration. I had to perform an exhaustive search through all of the mutually exclusive device driver options until a configuration was found that provided the desired device support on test hardware.
- I wanted to enable `config_xen` but the dependencies are not visible in `menuconfig`
- My kernel version was 2.6. I was following a HowTo that told me to enable some iptables match or target that I couldn't even find. I looked all throughout the configuration options, unsuccessfully. I decided to come back to it later, reasoning that maybe something else I would enable would then make the option present. When I went back later, there it was. I decided to find out what activated it, and went to a different part of the networking configuration, and disabled likely suspects which I had just enabled. I narrowed it down until I identified the option that made it visible. I enabled the options. I haven't had to alter it since then, because I always either copy the config file or refer to it when upgrading or doing a similar project.
- The toughest activation I had was in version 2.6.38 to 3.0 when trying to enable Xen. It took a while for me to find the required kernel configuration on Xen's homepage. It turns out I had to enable PAE.

### 1.2.2 Problems faced

As for the examples, we only present the most representative comments expressed by the respondents. To facilitate understanding, the problems are grouped in related topics. The original number of comments per category is indicated between () for completeness.

- Understanding dependencies (13):
  - Generally, when the dependency list is too long, the help is then not very useful. A lot could be done to improve this!
  - It can be difficult to find how to disable a `kconfig` symbol, especially if it is selected by many other symbols.
  - I would like all driver names mentioned in the help menu. For instance: enabling this driver will give you driver foo.

- I have problem changing features from module to built-in when the dependencies are also set to modules.
  - In most cases, I want to deactivate a feature I do not need or want and have to deactivate other features, which depend on that feature. I have to follow all dependencies list in the help menu and deactivate them manually. For example, some network filesystems depend on exotic network protocols I do not want to have in my kernel.
  - A very common problem is incorrect 'select' statements, where some option implicitly turns on another option but not its dependencies.
  - I've had more problems with deactivating options than activating them.
  - As far as consistency checking and helping determine inter-related dependencies on settings, I have long wished for a better kernel configuration tool that either menuconfig or xkernelconfig, but it seems that the kernel guys learn their way around the configurator by much exposure, and the rest of us have to just figure it out, or ask on these forums.
- Understanding/Finding the documentation (15):
    - Many options' effects are insufficiently documented so I end up blindly choosing the default or recommended value.
    - It is hard to figure out what a feature does. The description sometimes are not clear.
    - The defaults often disagree with the help text advice. For example, it will say, "If not sure, say "Y", but the option will be disabled by default, or vice-versa.
    - Sometimes I really don't know what an option does. For example, in 3.1-rc2 there is "CONFIG\_WATCHDOG\_CORE: Say Y here if you want to use the new watchdog timer driver core." What is the difference between CONFIG\_WATCHDOG and CONFIG\_WATCHDOG\_CORE? I can enable the watchdog without this option, but will it work? If disabled, will it use an old driver core or what? Also, it says "WatchDog Timer Driver Core" but there is no timer mentioned in CONFIG\_WATCHDOG\_CORE. Just to be safe, I disabled this option. After all, my watchdog worked before this option was introduced so it should continue to do so without it being enabled, shouldn't it?
    - Many hardware drivers and kernel functionalities are still not layman accessible, although the default choice gives some info on the general use of that option. Hardware drivers are specially tricky (I messed with sensors recently), since motherboards often feature many chips that we don't even know existed in the first place.

- Few explanations for all the options, in clear words understandable to non-kernel hackers
- Unclear relationships between drivers and new hardware. Even if the kernel supports a new device it is often not obvious which drivers should be used.
- High number of separate items for various hardware pieces I do not own but have to either compile or manually disable.
- Some features are inadequately described in the configure help, making it difficult to decide what the features do without reading and understanding the source code changes that occur.
- I have been using Gentoo for ten years or so and I still don't understand all the kernel options. Quite often there is a recommended option (in menuconfig), which is invaluable when you don't know what to choose, but there are also some items where you just have to take your best guess. This can be particularly difficult for a first-timer. With more experience, you sort of get a feel for what's important and what isn't. No doubt I could find out more about the options with a bit of research, but in essence this is just bad software design which breaks the First Law of Programming: (1) never present a user with a choice without also explaining the implications of each option and provide this information in the same part of the program where the choice is presented . It's absolutely shocking how often that rule is broken, and not just in the kernel of course.
- Sometimes the help-text to a configuration is minimal/lacking.
- Keeping only what is needed (4):
  - I have sometimes problems finding out which options I don't need so I can leave them out.
  - Minimising the number of modules compiled in is important in order to reduce the potential "attack surface" within the kernel. Thus, enabling all options is not an option, and discovering exactly which options are required for given hardware is necessary. Of course, this takes much longer when configuring a newly built machine, compared to an existing, upgraded one.
  - One thing we should always keep in mind is to generate a minimal kernel to do the job. We do not want a bloated one-size-fits-all kernel, akin to the genkernel solution used to boot the distribution media. It makes sense there to get something up and running to start the installation process, but a production kernel needs to avoid unneeded components both for speed, and economy reasons, not to mention security.
- Finding the right option (5):

- It is frequently incredibly annoying to find the menu item for a particular option, even given the name of the config option.
  - It is hard to find where a feature is. When one follows some howto, the kernel feature is moved or renamed and difficult to locate.
  - menuconfig grows larger on every kernel release, which makes finding the right options harder. Unfortunately xconfig is not exactly usable (yet).
  - Usually it is difficult to search for new options. In the nconfig interface, using F8 one can search for a config option, but the search term must be close enough to (a substring of) the actual configuration/module name. Till now, to determine the options I need to enable, I usually follow a sequence of steps. (1) Feed lspci -n into a debian hcl webpage which returns me the list of modules I need. (2) Search for those modules in the kernel. (3) Search in Google or Gentoo documentation for any other options I might want to enable (for example configuring the machine as a router, etc). Since the config options are pretty well documented, it would be much nicer if the kernel provided a way to search through all the documentation. For example, if I have a device by manufacturer X then searching for the manufacturer name X should return me a list of the kernel options that are applicable to that manufacturer. The nconfig interface itself is a sizable improvement compared to menuconfig. One can quickly jump to the options in a page by searching for the string via `"/`.
- Handling kernel updates (4):
    - Occasionally, options which were introduced or changed in new kernel versions are not fully documented, giving insufficient information regarding whether I should enable or disable the new option. As I use a saved configuration and 'oldconfig' as my primary method of configuring kernels, I typically do not have to reselect options which I had previously enabled, vastly simplifying the configuration work.
    - Driver cleanups that move PCI ids between PCIe and PCI drivers can be a real pain if it happens on a kernel update and I was not expecting it.
  - Erroneous kconfig (3):
    - My most common problem is inconsistencies in the kconfig files. Typically, I select an option and find out that it does not build when combined with other options. I then study the source of the problem and change the kconfig files to correctly reflect the underlying situation, by adding additional 'depends' statements or changing the 'select' statements to work correctly.
    - Stupid/nonsensical/dangerous defaults.

## 2 eCos

This section focuses on the configuration tools used for eCos. We investigate eCos because it is used by a growing community developing embedded software with high expectations in terms of quality and efficiency.

To harvest data, we designed a questionnaire (see <https://docs.google.com/spreadsheet/viewform?formkey=dDhrRmNVZ3NQmMyNTUORjBQOHg4eEE6MQ>) that was posted on the eCos mailing list and also submitted to the developers of several projects, among which ReconOS and DancOS.

This document contains a summary of 9 answers to the questionnaire. This summary starts with a quantitative analysis that presents the profiles of the respondents and their configuration practice. Then, a qualitative analysis reports techniques used to handle conflicts not resolved by the configurations tools, and outlines several configuration challenges expressed by users.

### 2.1 Quantitative analysis

#### 2.1.1 Respondent profiles

##### Level expertise.

*What is your level of expertise with eCos?*

- *Expert*
- *Moderate*
- *Newbie*

According to Figure 10, more than half of the respondents consider their expertise to be moderate, whilst a third are claimed experts.

##### Years of expertise.

*For how many years have you been using eCos?*

Figure 11 indicates to what extent the level of expertise is connected to the number of years of experience with eCos. The only newbie respondent had less than one year of experience with eCos, hence its absence from this chart. Similarly, one respondent with a moderate level of expertise had less than one year of experience with eCos.

##### Types of project.

*What type of project have you been using eCos for?*

The limited number of answers and diversity of projects gives hardly comparable groups, as shown in Figure 12. Furthermore, FPGA (Field-Programmable Gate Array) are generic programmable boards, which provides little information on the type of project.



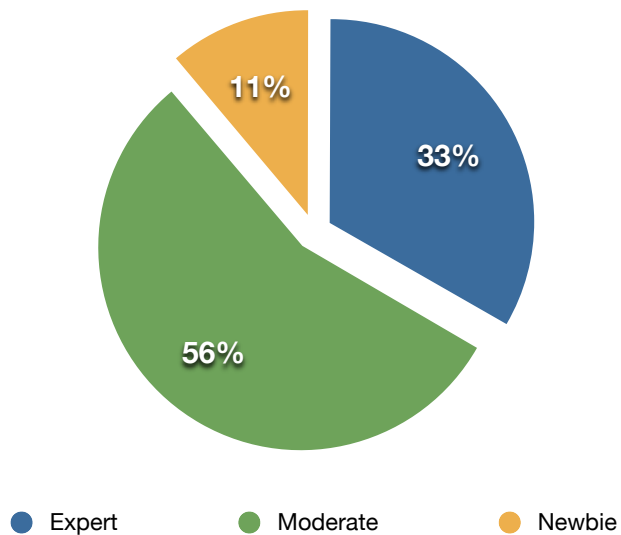


Figure 10: Distribution of expertise.

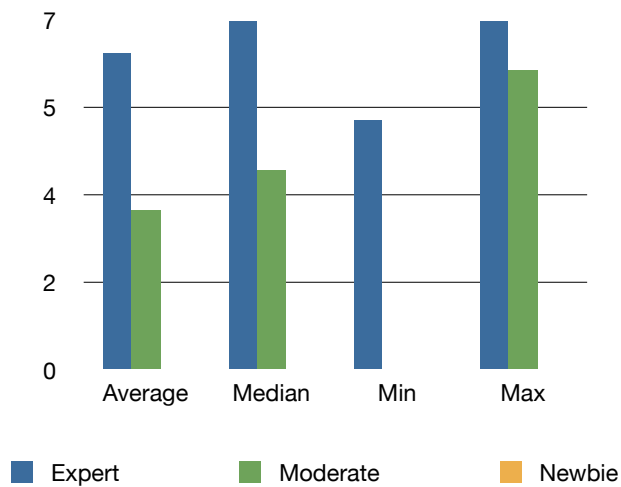


Figure 11: Average, minimum and maximum number of years of expertise per category.

### 2.1.2 Configuration practice

#### Concurrent configuration.

*How many users are usually involved in the configuration activity?*

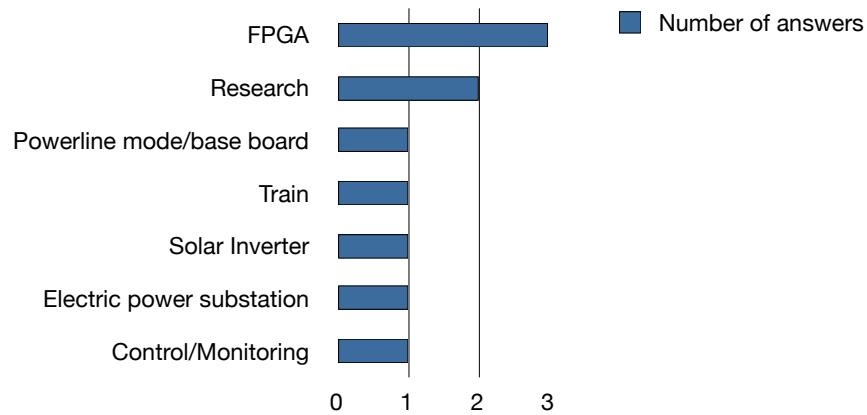


Figure 12: Distribution of usage.

*If more than one user is involved, are there conflicts between their decisions?*

- *Yes*
- *No*

*If you answered yes to the previous question, how do you handle these conflicts?*

In most cases, the configuration of eCos is performed by one user. Two respondents report that sometimes two users configure the same model, but they rarely encounter conflicts. When they do, they resolve it through discussions.

### **Number of changes.**

*How many changes (e.g. enabling an option or setting an attribute value) to the default template do you usually make in a complete project? Please provide an range.*

Figure 13 shows how many users (Y-axis) make more than  $x$  changes (X-axis). Half of the respondents make between 10 and 20 changes.

### **Frequency of conflicts.**

*Please estimate the number of encountered conflicts per number of changes.*

The respondents reported encountering conflicts between one per decision and one per twenty decisions. The discrepancies between the answers and their limited number do not allow to plot a meaningful chart.

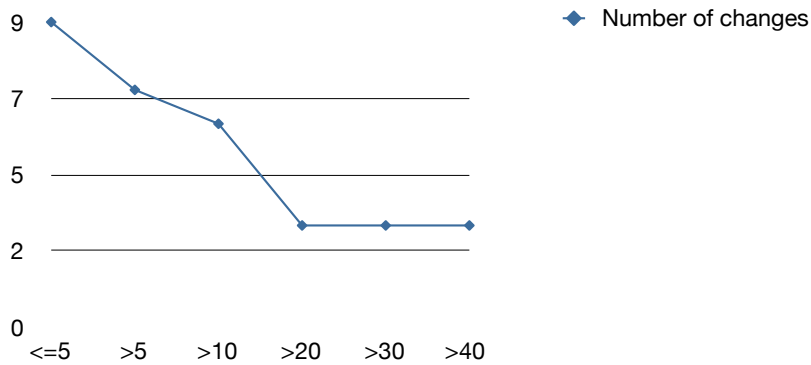


Figure 13: Cumulative number of changes and activations of inactive options.

#### Degree of satisfaction of the conflict resolution mechanism.

*How often is the solution proposed by the tool to resolve a conflict the one you need?*

- Never*
- Occasionally (3 or less out of 10 conflicts)*
- Sometimes (between 4 to 7 out of 10 conflicts)*
- Often (7 or more out of 10 conflicts)*
- Always*
- Other*

Less than a quarter of the respondents are always satisfied with the fixes proposed by eCos (see Figure 14). Half of them are often satisfied, while one is sometimes satisfied and one only occasionally.

#### Alternative conflict resolution techniques.

*If the solution proposed is not the one you need, which of the following alternatives do you most frequently use to solve the conflict?*

- Manually make changes not proposed in the solution*
- Select another hardware platform or template*
- Ask for a solution on a mailing list/forum*
- Other*

Most users manually make changes to resolve the problem, as show in Figure 15. Only one respondent selects another hardware platform or template.

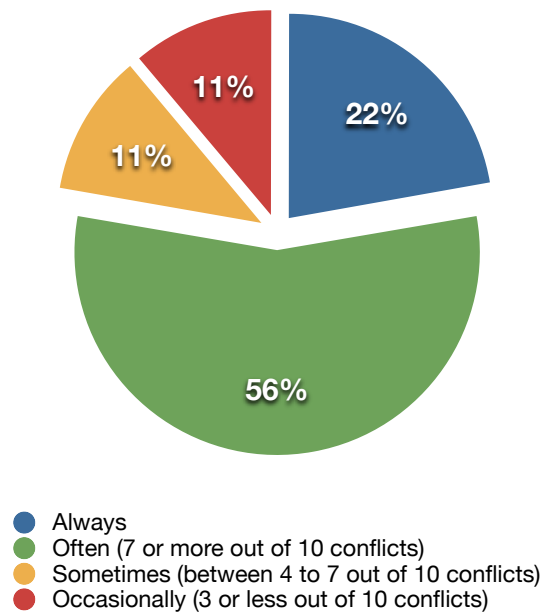


Figure 14: Degree of satisfaction of the conflict resolution mechanism.

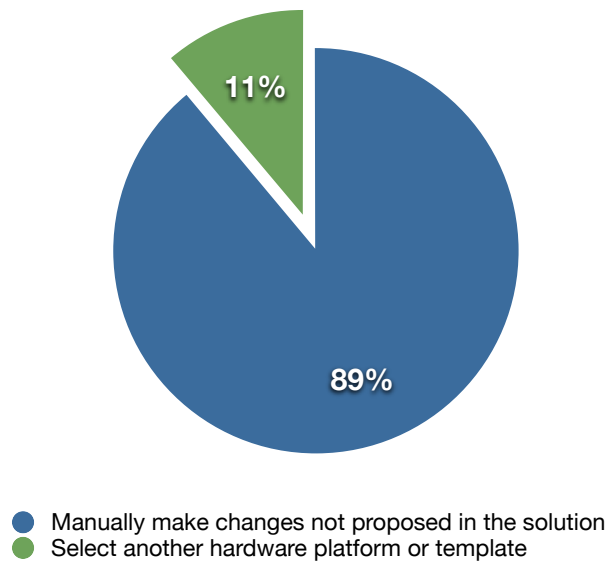


Figure 15: Distribution of alternative conflict resolution techniques.

**Most difficult problems faced.**

*Which of these two types of conflict is the most difficult to solve?*

- *Enabling/disabling (activating/deactivating) an option*
- *Setting the correct value of an attribute (e.g. string or integer)*

*Do you sometimes have difficulties to change options that are not active?*

- *Yes*
- *No*

78% of the respondents report that enabling/disabling an option is the most difficult task to achieve when the option is deactivated by some constraints, while setting the value of an attribute is the most difficult task for the remaining respondents. 56% of the respondents consider that enabling/disabling an option is actually a problem in practice.

### **Resolution time.**

*On average, how long does it take to solve a conflict?*

- *A few minutes*
- *A few hours*
- *A few days*
- *I do not remember*

*Do you usually resolve conflicts as soon as they appear or do you postpone the resolution?*

- *As soon as they appear*
- *Postpone*

Inactive options are activated in a few minutes in most of the cases, as show in Figure 16. Only one respondent can need more than one hour. Interestingly, this respondent is not the newbie and has more than six years of experience with eCos. All the respondents resolve a conflict as soon as it appears.

## **2.2 Qualitative analysis**

### **2.2.1 Techniques used to solve an unresolved conflict**

In total, we collected 5 suggestions to solve unresolved conflicts. We present below those which contain enough details to understand the technique and are most representative. The descriptions are verbatim copies of user answers modulo minor edits to improve readability.

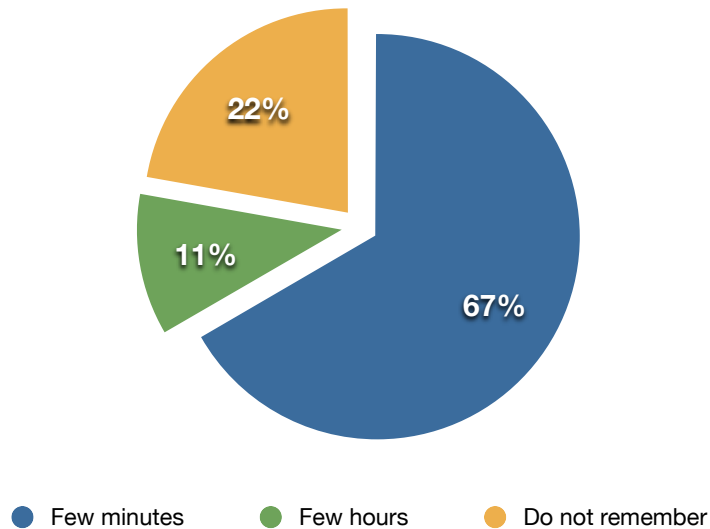


Figure 16: Distribution of the time necessary to activate an inactive option.

- I read the source code, look at the dependencies and work out why it is inactive.
- I look in the CDL files for the dependencies and change it if need be.
- I use a different template configuration file that has the package/option I require already included. Or I ask for help from my colleagues.

### 2.2.2 Problems faced

As for the examples, we only present the most representative comments expressed by the respondents.

- I have issues with multiplicities, such as when you add more than one device of the same type and the configuration only supports a single instance. SPI Flash drivers used to be or maybe still is like this.
- I have problems knowing in which package/template some options reside. When selecting a package knowing which other requirements it has.
- I used wrong configuration file for my design
- Important paths were not included, which caused the failure of the configuration process.

### 3 Conclusion

To understand what challenges are faced by the users of modern configurators, we carried out two empirical studies of Linux and eCos. Two questionnaires were submitted to forums, mailing lists and experts with whom we collaborate. In total, we collected answers from 92 Linux users with up to 20 years of experience, and 9 eCos users with up to 7 years of experience. Many interesting results were found and detailed in this report. Particularly, with respect to conflict handling, we have found the two challenges:

- **Activating inactive features.** 18% of the Linux users report that, when they need to change an inactive option, they need at least a “few dozen minutes” in average to figure out how to activate it. 56% of the eCos users also consider the activation of an inactive option to be a problem.
- **Fix incompleteness.** Existing configurators generate only one fix for an error. However, there are often multiple solutions to resolving an error, and the user may prefer other solutions. 7 out of 9 eCos users have encountered situations where the *generated fix is not useful*.

Several users also report having problems understanding and finding the documentation, finding the right options, handling updates of the Linux kernel, and correcting erroneous CDL or Kconfig files. Finally, the size and complexity of the Linux kernel often prevent users from keeping only the features they need. The minimality of the kernel is not only desirable to speed up boot time and save resources, it is also reduces possible security threats. The recent *kernel seeds*<sup>1</sup> initiative is a first step towards more economical kernels.

---

<sup>1</sup><http://www.kernel-seeds.org/>