Some technical information about #rosa-devel

№3

ROSA Planet

Just Open!

Contens

Introduction

Slowly but surely, we've got our third ROSA Planet. Since the second bulletin has been released, we have presented the release of ROSA Enterprise Linux Server, and in December we are planning to finish our ROSA Desktop 2012. We really hope, that it won't hurt the New Year issue of ROSA Planet and we call to you for participation (more information in the first article). We hope for your help and support.

It's not a secret that we really like to receive positive reviews and feedback from you . It is even more useful to get critical comments. But what makes us really proud is a wave of similar projects ina Russian Linux community which was initiated by our example. If it isn't an epidemic, than it is certainly a virus :)

ROSA: Questions and Answers

In the issue that we are planning to release before the New Year, we would like to offer you something new. If you have questions to developers or management of our company, feel free to send them to ask-rosa@rosalab.ru. Questions will be transferred to corresponding persons and the most interesting ones will appear in the New Year ROSA Planet issue. We would be very much obliged to you if you do not write questions in the news comments on different public resources, forums, etc., but send them to the specified address. It will really simplify collection and analysis of the questions.

Desktop

Developer rut — power consumption and hardware detection

While developing ROSA Desktop 2012, we have provided a huge mass of conversions and the transition to new versions of software: kmod, grub2 and new kernels. We have also decided to use the last version of X-server, numbered of 13, just to keep up to the other distributions (Ubuntu was the first to use it and it will be applied in all new products of this winter). This has led to a questions: how the detection of hardware will work? Why now do we detect devices a bit worse and what shall we do with videocards support and novelties that comes with cores > 3.4.x?

As a result, we performed revision of distribution software and we performed a giant work to apply all novelties to the source code. At this moment all components are fully developed in ROSA, you can find git with them on ABF (https://abf.rosalinux.ru/soft).

Now, in order, what we did for the last time: X-server was updated to the version 13, all modules and drivers were updated too. The transition was painless: we have only one loss, it is free drivers for AMD/ATI videocards older than HD5xxx.

 Ldetect-lst: a part that is responsible for correct detection of hardware in ROSA utilities. Here we have two main changes:

- Update of device database (we hasn't did this since spring, 2011) and also new modules of recieving information — from the list of kernel modules, because of transition to the KMOD;

- New algorithm of work with AMD/ATI videocards, which was rewritten and now allows to correctly choose drivers (obsolete driver radeonhd has been deleted and we have started to

develop a mechanism for legacy drivers which, unfortunately, don't work with the latest X-server).

Drakx-utilities: the key problem that was solved is recovery of detection of power consumption modes. They were rewritten to work with cpupower. Now desktop kernels automaticly detect enabled ondemand governor (this means cutting frequency and power consumption of CPU, when it doesn't need them) and system correctly detects necessary kernel modules and loads them at start.

XFdrake: after migration to the Grub2, we have lost ability to add new option (nokmsboot) in kernel bot line. We have also rewritten this part, restored correct installation and work of videodrivers (especially proprietary ones). Together with upgated Card+ list, videocard detection should be much better and more correct now.

Of cource, we have some problems: not every time the switch from a free driver to a proprietary one is smooth, new X-server presents its own surprises, Pentium 4 and old Atom users forced to use detours for right power consumption. But the most important thing is that all this problems can be solved.

In general, we have performed a significant work to update the core system. This problem hasn't been addressed for at least last two years

Integration between DrakConf u KDE Control Center

One of the directions of ROSA Desktop development is getting rid of obsolete drakxtools. In particular, it would be nice to avoid the situation when we have two places to configure the system - DrakConf ("Configure Your Computer" icon in SimpleWelcome) and KDE CC (or approrpiate tools in Gnome and LXDE; "Configure Your Desktop" icon). some of drakxtools program have direct (and more contemporary) analogues in KDE, but for some of the tools alternatives with the same functionality are missing.

In the coming release, we are going to drop DrakConf as an application. Those drak tools that don't have direct analogues in KDE, will be integrated into KDE Control Center. In order to do this, we are preparing a [KCM module](#) for every of that tools. You can already try the first versions of such KCM wrappers for some tools:

kcm-drakauth
kcm-drakfirewall
kcm-drakguard
kcm-drakinvictus
kcm-draksec
kcm-harddrake
kcm-rpmdrake-sources
kcm-rpmdrake
kcm-rpmdrake-update
kcm-update-freq
kcm-userdrake
kcm-XFdrake

It's likely that not all this modules will be installed by default in the release. The list of modules that are going to be added by default can be observed here:

Alternatively, you can install the **drakconf-kde4** package; those kcm wrappers that we are going to provide by default will be installed together with that package.

Hardrake will be available in the Hardware section, most of other tools - in the System Administration section:
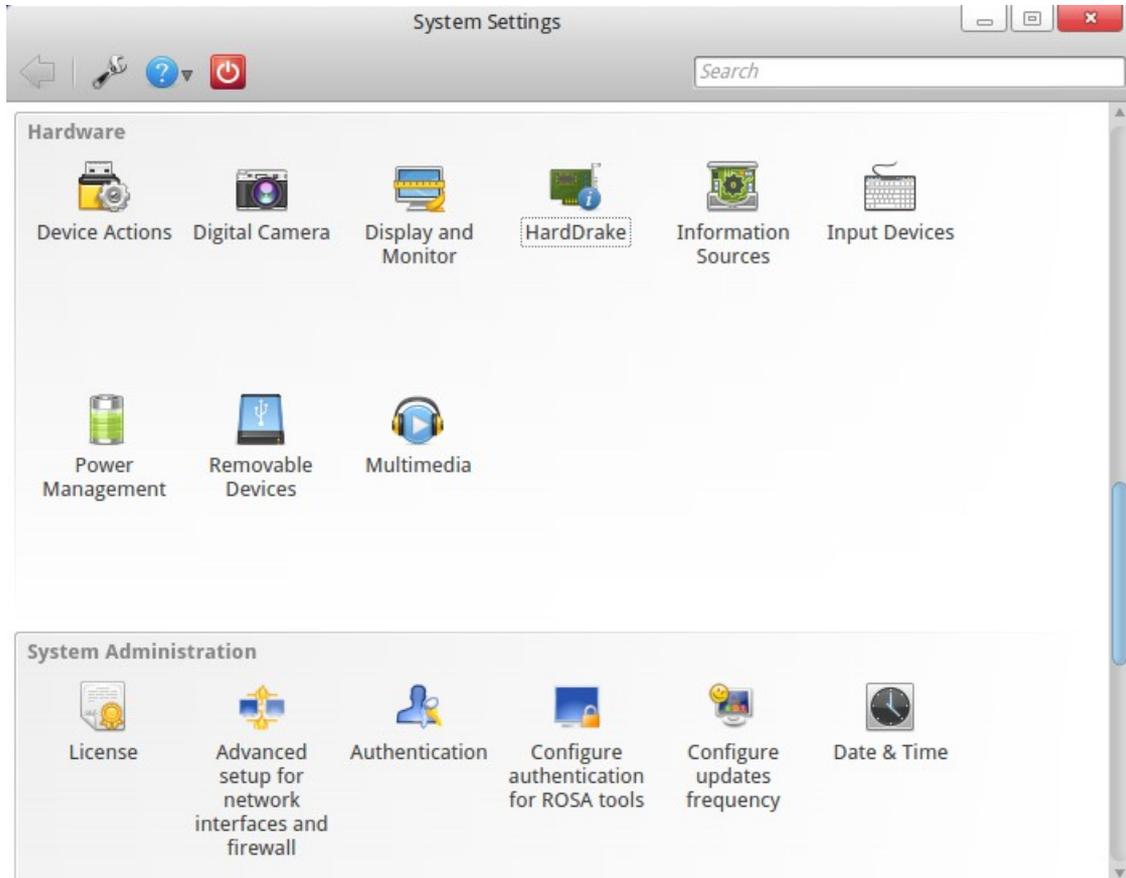
Development tools

FBA: reports about closure of repositories with respect to binary symbols and «obsolete» packages

For a long time regulary updated reports are available at http://fba.rosalinux.ru on dependency closure of ROSA repositories. However, presence of all the dependencies doesn't guarantee success of apps work, because packages installed by dependencies can miss some elements nessesary for normal work. The most often example is absence of shared libraries and binary symbols required for application . This happens if package with nessesary library isn't installed by dependencies or contains too fresh version of library, which hasn't got some functions.

To automatically track such cases, we set up analisys of all ROSA ELF-files on FBA to compare provided and required sets of bynary symbols and libraries. At this moment such analisys is provided for the main repository only. Reports are available here: http://fba.rosalinux.ru/test/elf_closure/

While working with large repositories, you can face the following problem: two packages are available in repository one of which is marked as «obsolete» in favor of another. Frequently, this is a necessary measure, because other packages can depend on the obsolete one , and while they aren't updated, you can't delete such package from repository. However, if you have a few thousands of packages, it is hard to observe the current situation with dependences and to delete packages that became useless. Now FBA helps in solving this task, providing reports about obsolete packages:

http://fba.rosalinux.ru/test/repomanage_obsoletes/

ROSA Server

Results of testing binary compatibility of RHEL and ROSA Server

It's not a secret that our operationing system ROSA Server is based on the most popular and stable enterprise distribution — RedHat Enterprise Linux (RHEL). We have added our own applications, just to improve the functionality for our customers. After all, the distribution and system part was modifed to provide normal working of these apps. As many of you likely know, such changes can break the binary compatibility between distributions and complicate the transfer of software between distributions (if the binary compatibility was broken, recompilation or maybe even modification of the program can become nessesary).

While modifying the package base from RHEL, we aimed to preserve full binary compatibility with the parent distribution. This should guarantee that all applications that work in RHEL will successfully run in ROSA Server. To control the binary compatibility while developing, we have applied special tools — such as ABI Compliance Checker, which is being developed in ROSA at the moment. The tool was created to detect modifications in libraries tat can break the compatibility.

Unfortunately, compatibility can be broken not only by modifications in libraries. Problem can be also introduced be changes in Perl or Python modules, configuration files or any «interface» files of the system that can be used by apps, etc. We don't have a toolkit for completely automated analysis of all possible modifications, but specially for such cases we have an utility for visualisation of changes in packages — PkgDiff, using which you can watch the changes quickly.

For visualisation of changes in all packages of distribution, we have created a special tool - DistDiff that launches PkgDiff on modifed packages which include interface files, and saves statistical information about changes. Report of the tool was analysed by experts in binary compatibility from ROSA. As a result, most modifications were found to be compatible, and incompatible changes were fixed before the release. Changes in packages, specified in the final version of the report don't lead to compatibility breakage. So ROSA Server is 100% compatible with RHEL at binary level.

ROSA ABF

ABF: Basic API, inline comments

Implementation of two new features has been presented by ABF team in October: ABF API and inline comments.

Now you can manage all basic operations in ABF, excluding mainteiner db and product build (ISO creation), using ABF API. Documentation can be found at http://abf-doc.rosalinux.ru/. ABF API provides about 60 API calls. Note that API is in a beta state. Breaking changes may occur.

The main product where ABF API is currently being used is ABF console client.

The next feature is inline comments: now ABF allows you to either comment on each commit as a whole, or click on any line and start a conversation on that particular line. Now you can discuss more specifically spec files, patches or new code. Try it!

ABF console client

As it was written before, the new way of interaction with ABF - console client - have recently appeared. It took ABF developers much effort to develop API in October. It made it possible to add lots of features to console client, but it was decided to improve the basic functionality first. So, what is ABF-Console-Client needed for? Let's take a look at package life-cycle. User wants to create (or get existing) project, modify it, build and publish. Let's start in order.

1. **Clone git repository.** A project name is assumed to be known. User have to open ABF web page, search for a project, go to a project page, copy git URL, clone repository using git. However, it can be done easier. Just execute "abf get PROJECT", where PROJECT is a project name with owner (owner/proj_name). Owner can be omitted, default group will be used.
2. **Project modification.** When you've got files in your project modified, you can execute "abf put -m MSG". It will execute "git add --all", "git commit -m MSG", "git push". Additionally, all the source archives have to be uploaded to File-Store and .abf.yml file have to modified. It's a routine work, especially when you have to build a hundred projects a day. Console client will do it for you automatically.
3. **Create a build-task.** To do it, you have to open an ABF web page, click some check-boxes and click 'Start build'. It looks like a few work to do. But if you have to build hundreds packages a day, you can do it much faster using console client, especially if you write your own script executing console client.
4. **Building process.** To check the current building status, you don't have to check web page anymore. Just execute "abf buildstatus ID" to get a short summary. ID can be omitted and the status for the last built project. It will be described later in detail.
5. **Publication.** "abf publish ID" will do it easily.

As you can see, console client makes maintainer's work easier a bit. That's why we work hard to improve the tool and to satisfy maintainers' needs.

The list of features added in the last version:

- ABF client caches the locations of git-repositories in your system when it accesses them. If you have already got lots of repositories, you can cache their locations at one go. Just run

"abf locate update-recursive -d PATH", where PATH is a directory with repositories. For what perpose should one know the location of every repository? For example, user can now execute "abfcd PROJECT". You can just learn the location of any project by executing "abf locate -p PROJECT". Console client will be able to move files not only among the project branches, but also among projects.

- First version of console client had only got the basic version of bash autocomplete script. Now it autocompletes almost everything: option names, git branches, build and save-to repositories for "abf build" (the latter requires a project name to be specified, so you have to type the --project/-p before --save-to-repository/-s). As a result, it's much easier to work with console client, because you don't have to type the long repository name or remember the list of possible variants. Sometimes autocompletion works long (about 1 second), but the results are getting cached and the process is speeding up.

- The new command, "abf clean" appeared. It scans the spec and .abf.yml files and current directory and shows problems found. If some source or patch file can not be found (or resolved via URL or .abf.yml) - it prints an error message. It also prints warnings, for example, if a file is specified in .abf.yml and in spec file. The test is automatically executed while creating a build task from the current directory and prevents a build task from being sent. This check can be turned off by option --skip-spec-check.

- When user creates a new build task, its ID will be stored and associated with a project. "abf buildstatus" will print the summary for the last build task. If project was specified (or you are in a project directory) - it will print information about the last build task for this project.

- About work with API. The number of unnecessary API calls have been significantly decreased. The fact is that when you, for example, requests information about repository, API returns some particular information about platform too. The older version could only use the ID of this platform and downloaded the full information about platform by one more API call. Now this information is stored in platform object that is marked as 'stub'. When you try to access a field that is not loaded but have to present in this class - a new API call will be made and information will be retrieved. As a result, the first console client call doesn't results in dozens of API calls anymore.

- One more interesting feature - work with (http://en.wikipedia.org/wiki/HTTP_ETag) to cache the results of API calls with autovalidation. It results in speeding the process up without a risk of working with the outdated data. ABF is not well configured now to fully support this technology, but it's going to process requests for cached data fast. It will also stop increasing a counter of API calls (every IP is limited to 500 API calls per hour).

As you can see, the ABF console client development continues. All your feedback (either positive or negative), suggestions and regards are welcome.

LinuxCon Europe 2012

The conference was really informative; the talks were divided to five parallel streams. Often talks for similar target audience were given simultaneously, so we had to make a difficult decision – which one to visit. This was partly compensated by free availability of slides (now you can find a lot of presentations at Linux Foundation) and by opportunity to find speakers at coffee-breaks or at evening parties from organizers, where a lot of interesting discussions on hi-tech themes has taken place. By the way, the conference atmosphere was far from formal — people came to talk about

topics related to their interests and to learn something new (they didn't come for t-shirs, pens and other souvenires, despite of the fact that the conference was rich for this stuff).

There were both pure technical presentations and "philosophical" ones. The popular technical themes included:

- cloud computing (the range of presented companies and projects was really wide — Citrix, Apache Cloudstack, OpenNebula, IBM, HP, and even Microsoft);
- related to them virtualization questions (in the main part of the conference it was told about KVN, Xen and about solutions from Parallels, and after the main part the seria of seminars about KVN and oVirt took place);
- a separate section was focused on Tizen (which also was actively advertised by Intel officials on their stand);
- a whole-day workshop devoted to GlusterFS took place, as well.

Certainly, leaders by number of visitors were reports of invited speakers — not only because of the fact that there were no other presentations in parallel, but also because their topics were actual and speakers were interesting.

Linus Torvalds didn't have any presentation and any particular topic for the talk («I just don't know what you are interested in» — «Nvidia!» — «I have already given an exhaustive answer :)»). Instead of this, we've got entertaining and informative discussion between the speaker and people in the auditorium. Of cource, not only funny questions were asked — for example, the problem of cooperation with developers of Android was touched upon; sometimes they don't hurry to open their developments, not mentioning of passing them to the upstream. The creator of Linux kernel hasn't expressed great concern over this issue, propounded to recall the situation of ten years ago when developers of leading distributions (RedHat, SUSE) prefered to maintain wide range of their own patches to the kernel and not to work on the common code base. Step by step the situation was improved: although a lot of patches patches are still developed inside distributions, they cooperate actively with other commands, particularly with developers of kernel. Torvalds has expressed an opinion that in case of Android the process will be worked out suitable for both sides.

Some session reports also collected respectable numbers of listeners — for example, the story from James Bottomley about the situation with UEFI (James promised that in the nearest future The Linux Foundation will reach agreement with Microsoft and a loader signed by MS key will be available that can be used to load other systems with Secure Boot). Thiago Macieira with story about Qt Project also collected a lot of visitors — many people are monitoring the gradual opening process of developing Qt.

My own experience is that e.g. open bug-tracker for Qt is a really useful thing, disposing to the involvement of third-party members. While working with LBS tests, sometimes we find potential errors in Qt, but two or three years ago the processing of error reports was quite obscure. From time to time we had the impression that some reports just dissapear right behind the bug reporting interface: without additional communication with Qt developers, we couldn't watch the status of the reoport or learn if somebody is working on fixing the problem. Now Qt turned its face to the community. But on Thiago's question «How many people here are subscribed to the mailing list of Qt developers» only Lars Knoll (the chief architect of the project) raised his hand.

Besides the talks, a lot of information was provided at stands of companies and projects. There was a great amount of stands related to cloud computing and virtualization, some stands demostrated distributions and related solutions. Besides ROSA, these category included Fedora/RedHat, SUSE, WindRiver and a number of others. Oracle officials were telling not only about their distribution based on RedHat, but also about other open projects (for example, MySQL), assuring everyone that the corporation is aimed to mutually beneficial cooperation with the community.

With efforts of Intel and WindRiver, people paid attention on embedded systems working under Linux. Intel advertised Tizen and demostrated prototype of IVI based on it. Many visitors looked on Tizen with caution, recalling in memory destiny of its predecessors — Moblin and MeeGo — that, in fact, sinked into oblivion.

Also Intel officials told about advantages of using the x86 architecture in creation of Android-based devices. One of the key moments was not advantages of the architecture itself, but the ease of development. After all, the creation of apps for Android usually happens on desktop (which overwhelmingly has x86 architecture) and testing is performed using the QEMU emulator. Through the use of hardware-assisted virtualization, the performance of QEMU in emulation of x86 platform is much better than in case of ARM, which significantly accelerates testing.

It's interesting that the most actively advertisement for Android was provided at the Intel stand. Google's stand was also devoted to Android, but was more modest, and representatives of the Internet giant didn't provide any agressive marketing actions.

Overall, LinuxCon Europe 2012 was informative and enjoyable; finally, it took two pages just to tell you about the most memorable aspects :)

Authors

Alexander Kazanzev

Andrey Ponomarenko

Anton Kirilenko

Anton Chernyshov

Denis Silakov

Konstantin Kochereskin

Petr Fomichev

All your questions, comments, recalls and wishes about interesting themes,

please, send to the rosa-point@rosalab.ru.