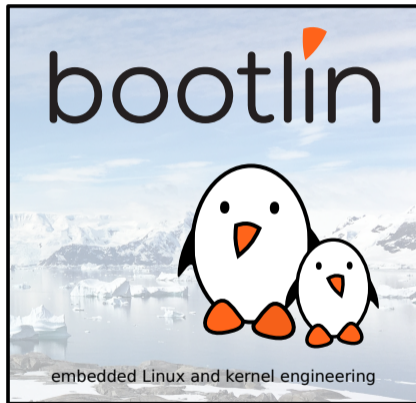




Buildroot: what's new?

Thomas Petazzoni
thomas.petazzoni@bootlin.com

© Copyright 2004-2020, Bootlin.
Creative Commons BY-SA 3.0 license.
Corrections, suggestions, contributions and translations are welcome!





- ▶ CTO/Embedded Linux engineer at Bootlin
 - ▶ Embedded Linux **expertise**
 - ▶ **Development**, consulting and training
 - ▶ Bootloader, Linux kernel, Yocto Project, Buildroot
 - ▶ Complete Linux BSP development
 - ▶ Hardware support in bootloader/Linux
 - ▶ Strong open-source focus: upstreaming and contributions
 - ▶ Freely available training materials
- ▶ Co-maintainer of **Buildroot**
- ▶ Living in **Toulouse**, France

bootlin





Buildroot at a glance

- ▶ Is an **embedded Linux build system**, builds from source:
 - ▶ cross-compilation toolchain
 - ▶ root filesystem with many libraries/applications, cross-built
 - ▶ kernel and bootloader images
- ▶ **Fast**, simple root filesystem in minutes
- ▶ **Easy** to use and understand: kconfig and make
- ▶ **Small** root filesystem, default 2 MB
- ▶ More than **2500 packages** available
- ▶ Generates filesystem images, not a distribution
- ▶ Vendor neutral
- ▶ Active community, stable releases every 3 months
- ▶ Started in 2001, oldest still maintained build system
- ▶ <http://buildroot.org>





Agenda

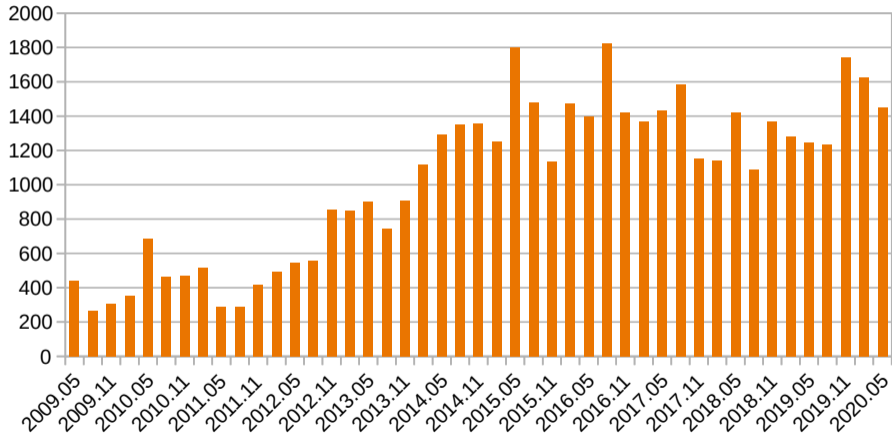
- ▶ See what's new in Buildroot within the last two years
- ▶ Covering Buildroot 2018.05 to Buildroot 2020.05
 - ▶ Community activity
 - ▶ Release schedule
 - ▶ Architecture support
 - ▶ Toolchain support
 - ▶ Package infrastructure improvements
 - ▶ Download infrastructure improvements
 - ▶ Interesting package updates and additions
 - ▶ Reproducible builds
 - ▶ Top-level parallel build
 - ▶ Tooling improvements





Buildroot: an active project

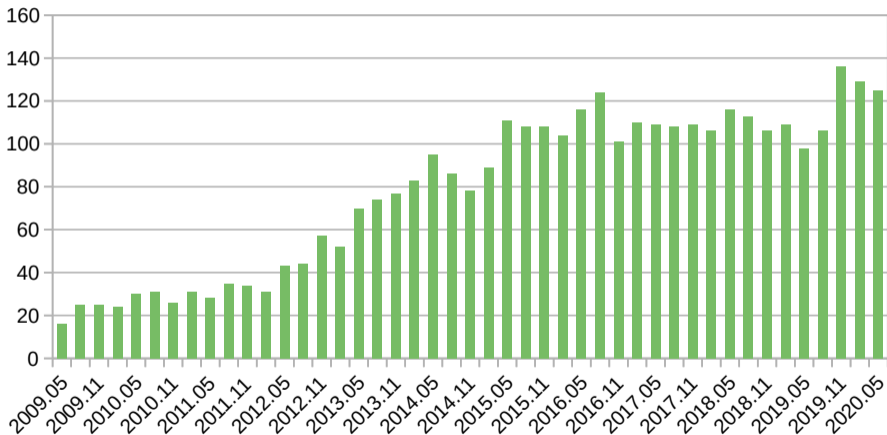
Number of commits per release





Buildroot: an active project

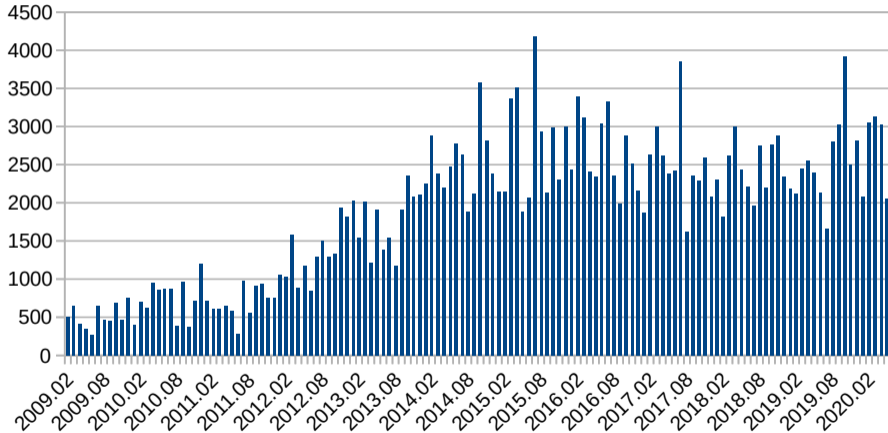
Number of contributors per release





Buildroot: an active project

E-mails on the mailing list





Release schedule: long term support added

- ▶ What we already had
 - ▶ **Four releases** per year: YYYY.02, YYYY.05, YYYY.08, YYYY.11
 - ▶ **3-month cycles**: 2 months development 1 month stabilization
- ▶ What we also have now
 - ▶ **LTS**: Long Term Support release
 - ▶ Each YYYY.02 release is **supported during one year**
 - ▶ **Security** updates, bug fixes
 - ▶ YYYY.02.x maintenance branch, and regular (\approx monthly) point releases
 - ▶ 2017.02: to 2017.02.11, 860 commits
 - ▶ 2018.02: to 2018.02.12, 1075 commits
 - ▶ 2019.02: to 2019.02.11: 1301 commits
 - ▶ 2020.02: to 2020.02.3: 344 commits (so far)





Architecture support

- ▶ Support for new CPU architectures
 - ▶ **RISC-V**, 32-bit and 64-bit
 - ▶ **NDS32**
- ▶ Support for **new variants** of existing architectures: ARM Cortex cores, x86 cores, MIPS cores, ARC cores, C-SKY cores
- ▶ Removal of *Blackfin* architecture
- ▶ Overall, **support for**: ARC, ARM, AArch64, C-SKY, m68k, Microblaze, MIPS, NDS32, NIOS2, OpenRISC, PowerPC, RISC-V, SuperH, SPARC, x86, Xtensa





Toolchain support: internal toolchain

Internal toolchain: Buildroot builds your toolchain from source

- ▶ No significant change, just regular updates
- ▶ **gcc** 8.x and 9.x added (gcc 10.x coming soon), gcc 4.9, 5.x, 6.x removed
- ▶ **binutils** updated, 2.33.1
- ▶ **uClibc-ng** updated, 1.0.34
- ▶ **musl** updated, 1.2.0
- ▶ **glibc** updated, 2.30
- ▶ Useful testing done by Romain Naour using the *toolchains-builder* project





External toolchain: Buildroot uses an existing pre-built toolchain

- ▶ ARM toolchains added
- ▶ AArch64 big-endian toolchains from ARM and Linaro added
- ▶ Andes NDS32 toolchain added
- ▶ Updates to numerous existing toolchains
- ▶ Declaring external toolchains from `BR2_EXTERNAL` trees





New package infrastructures: Go, Meson, QMake

- ▶ Package infrastructures **factorize the common logic** to configure, build and install packages that use a standardized build system
- ▶ **Three new** package infrastructures:
 - ▶ `golang-package` for Go-based packages
 - ▶ `meson-package` for Meson-based packages
 - ▶ `qmake-package` for QMake-based packages (Qt world)
- ▶ **Already** had support for: Autotools, CMake, Kconfig, Luarocks, Perl, Python, Erlang, Waf and kernel modules





Go package example: docker-cli

package/docker-cli/docker-cli.mk

```
#####  
#  
# docker-cli  
#  
#####  
  
DOCKER_CLI_VERSION = 18.09.9  
DOCKER_CLI_SITE = $(call github,docker,cli,v$(DOCKER_CLI_VERSION))  
DOCKER_CLI_WORKSPACE = GOPATH  
  
DOCKER_CLI_LICENSE = Apache-2.0  
DOCKER_CLI_LICENSE_FILES = LICENSE  
  
DOCKER_CLI_DEPENDENCIES = host-pkgconf  
  
DOCKER_CLI_TAGS = autogen  
DOCKER_CLI_BUILD_TARGETS = cmd/docker  
  
DOCKER_CLI_LDFLAGS = \  
    -X github.com/docker/cli/cli.GitCommit=$(DOCKER_CLI_VERSION) \  
    -X github.com/docker/cli/cli.Version=$(DOCKER_CLI_VERSION)  
  
DOCKER_CLI_INSTALL_BINS = $(notdir $(DOCKER_CLI_BUILD_TARGETS))  
  
$(eval $(golang-package))
```



Meson package example: libmpdclient

package/libmpdclient/libmpdclient.mk

```
#####  
#  
# libmpdclient  
#  
#####  
  
LIBMPDCLIENT_VERSION_MAJOR = 2  
LIBMPDCLIENT_VERSION = $(LIBMPDCLIENT_VERSION_MAJOR).16  
LIBMPDCLIENT_SOURCE = libmpdclient-$(LIBMPDCLIENT_VERSION).tar.xz  
LIBMPDCLIENT_SITE = http://www.musicpd.org/download/libmpdclient/$(LIBMPDCLIENT_VERSION_MAJOR)  
LIBMPDCLIENT_INSTALL_STAGING = YES  
LIBMPDCLIENT_LICENSE = BSD-3-Clause  
LIBMPDCLIENT_LICENSE_FILES = COPYING  
  
$(eval $(meson-package))
```



Download infrastructure improvements

- ▶ Main improvement: **Git caching**, for Git-fetched packages
 - ▶ Before: complete clone of the Git repository, checkout the requested version, create a tarball with the source code, throw away the Git repository
 - ▶ Drawback: another full clone next time the package version is changed
 - ▶ Now: keep a clone of the git repository in `$DL_DIR/<package>/git/`, much faster download when a Git-fetched package is updated
- ▶ Tarballs are now stored in per-package sub-directories in `$DL_DIR`
- ▶ Major rewrite of the internals of the download infrastructure, `package/pkg-download.mk`, `support/download/`

```
$DL_DIR
+ uboot
+ git
  + api
  + arch
  + board
  + .git
  + Kbuild
  + Kconfig
  + Makefile
  + MAINTAINERS
  [...]
+ u-boot-2018.11.tar.bz2
+ u-boot-2019.04.tar.bz2
+ uboot-228801a215909365a....tar.gz
+ uboot-37425027b617af670....tar.gz
+ zeromq
  + zeromq-4.2.5.tar.gz
  + zeromq-4.3.1.tar.gz
  + zeromq-4.3.2.tar.gz
```



Package updates and additions

- ▶ Between 2018.05 and 2020.05:
 - ▶ 435 packages have been added
 - ▶ A few packages have been removed: individual X.org proto packages, Qt4, GStreamer 0.10
- ▶ Addition of: Rust (compiler, cargo), LLVM/Clang (not as a compiler), Mender, OpenJDK, OpenRC init system, OP-TEE OS, GObject-Introspection, AppArmor, zillions of Perl/Python modules
- ▶ Update of all major software stacks: Qt 5.15, X.org 1.20, GStreamer 1.16, Wayland 1.18, Weston 8.0, Kodi 18.7.



```
$ git log --format=oneline 2018.05..2020.05 package/ \  
  | grep -i bump | wc -l  
4016
```




Hardening options

- ▶ Addition of support for building the entire code base with a number of security hardening mechanisms
- ▶ Improvement of Stack Protection options:
BR2_SSP_REGULAR, BR2_SSP_STRONG, BR2_SSP_ALL
- ▶ Addition of *RELRO* protection options:
BR2_RELRO_NONE, BR2_RELRO_PARTIAL,
BR2_RELRO_FULL
- ▶ Addition of buffer-overflow detection (*FORTIFY SOURCE*) options: BR2_FORTIFY_SOURCE_NONE,
BR2_FORTIFY_SOURCE_1, BR2_FORTIFY_SOURCE_2
- ▶ Mostly contributed by Collins Aerospace





New target: make show-info

- ▶ New top-level target:
`make show-info`
- ▶ Outputs a JSON blurb that provides lost of metadata about the packages enabled in the current configuration
- ▶ JSON makes it easily usable in scripts and tools
- ▶ Allows to analyze the contents of a system, validate the choice of packages, get their download URL, and more.
- ▶ Other analysis tool already present:
 - ▶ `legal-info`
 - ▶ `graph-build`, `graph-size`

```
{  
  "busybox": {  
    "type": "target",  
    "virtual": false,  
    "version": "1.31.0",  
    "licenses": "GPL-2.0",  
    "dl_dir": "busybox",  
    "install_target": true,  
    "install_staging": false,  
    "install_images": false,  
    "downloads": [  
      {  
        "source": "busybox-1.31.0.tar.bz2",  
        "uris": [  
          "http+http://www.busybox.net/downloads",  
          "http|urlencode+http://sources.buildroot.net/busybox",  
          "http|urlencode+http://sources.buildroot.net"  
        ]  
      }  
    ],  
    "dependencies": [  
      "host-skeleton",  
      "host-tar",  
      "skeleton",  
      "toolchain"  
    ],  
    "reverse_dependencies": []  
  },  
}
```



Reproducible builds

- ▶ Goal: given a Buildroot configuration/version, two builds will give two exactly identical results
- ▶ Google Summer of Code in summer 2019, with Atharva Lele working on *Reproducible Builds*
- ▶ Mentored by Arnout Vandecappelle and Yann E. Morin
- ▶ Automated testing on `autobuild.buildroot.org`
 - ▶ Some builds are done twice, with `BR2_REPRODUCIBLE=y`, and then tested for equality
 - ▶ If not equal, comparison done with `diffoscope` to facilitate the analysis
 - ▶ Differences between builds: build time and location
 - ▶ Ultimately between environments
- ▶ Fixes in tar, gzip and cpio handling to avoid timestamp issues
- ▶ More work is needed: improving the diffoscope reporting (in progress), fix the reproducibility issues





Reproducible build result

```
--- /home/naourr/work/instance-1/output-1/images/rootfs.tar
+++ /home/naourr/work/instance-1/output-2/images/rootfs.tar
./usr/lib/asterisk/modules/app_agent_pool.so
/home/naourr/work/instance-1/output-1/host/bin/m68k-linux-readelf --wide --decompress --hex-dump=.rodata {}
@@ -112,15 +112,15 @@
 0x0000a2f2 20746f20 6a6f696e 20746865 20627269  to join the bri
 0x0000a302 6467652e 0a004552 524f5200 4167656e  dge...ERROR.Agen
 0x0000a312 74202725 73273a20 4661696c 65642074  t '%s': Failed t
 0x0000a322 6f20616c 65727420 74686520 6167656e  o alert the agen
 0x0000a332 742e0a00 4e4f545f 434f4e4e 45435445  t...NOT_CONNECTE
 0x0000a342 44002f68 6f6d652f 6e616f75 72722f77  D./home/naourr/w
 0x0000a352 6f726b2f 696e7374 616e6365 2d312f6f  ork/instance-1/o
- 0x0000a362 75747075 742d312f 6275696c 642f6173  utput-1/build/as
+ 0x0000a362 75747075 742d322f 6275696c 642f6173  utput-2/build/as
 0x0000a372 74657269 736b2d31 362e362e 312f696e  terisk-16.6.1/in
 0x0000a382 636c7564 652f6173 74657269 736b2f73  clude/asterisk/s
 0x0000a392 7472696e 67732e68 00416374 696f6e49  trings.h.ActionI
 0x0000a3a2 44004163 74696f6e 49443a20 25730d0a  D.ActionID: %s..
 0x0000a3b2 00737461 72740041 67656e74 73207769  .start.Agents wi
 0x0000a3c2 6c6c2066 6f6c6c6f 77004167 656e743a  ll follow.Agent:
 0x0000a3d2 2025730d 0a004e61 6d653a20 25730d0a  %s...Name: %s..
```



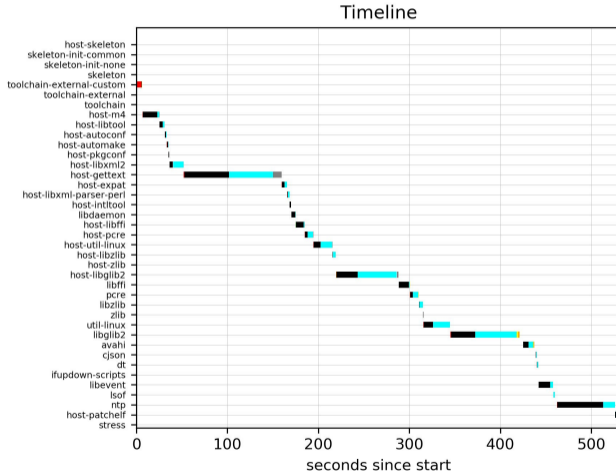
Top-level parallel build

- ▶ **Goal:** build several packages in parallel
- ▶ Experimental support **merged** in 2020.02
- ▶ `BR2_PER_PACKAGE_DIRECTORIES=y` option
 - ▶ Enables *per-package build*
 - ▶ Each package has its own `HOST_DIR` (including the compiler `sysroot`) and `TARGET_DIR`
 - ▶ Guarantees that the dependencies seen by the package are always consistent
- ▶ When enabled, `make -jX` at the top-level works!
- ▶ Still some limitations: Qt5 has issues, the `-rebuild`, `-reconfigure`, `-reinstall` targets are not working yet





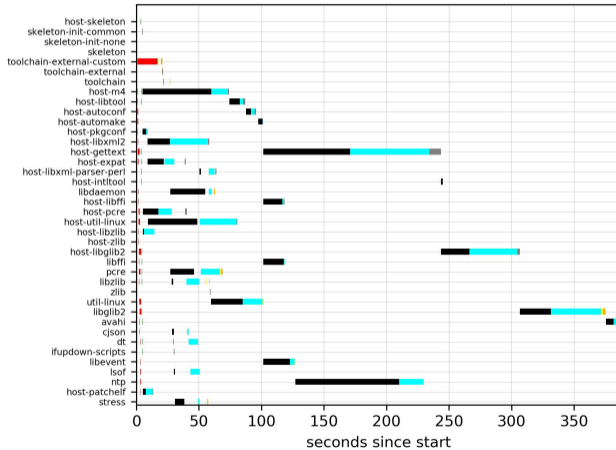
Without top-level parallel build





With top-level parallel build

Timeline





Runtime tests

- ▶ A *runtime test* infrastructure was introduced in 2017.02
- ▶ Each test case:
 - ▶ Builds a well-defined configuration
 - ▶ Boots it under Qemu
 - ▶ Runs some tests to verify that a given feature is working
- ▶ Complementary to *autobuilder* testing, which tests the build of random configurations
- ▶ `./support/testing/run-tests -h`
- ▶ Since 2017.11, many new test cases added
 - ▶ ATF, Python modules, Perl modules, Lua modules, OpenJDK, X.org/Mesa3D, Docker/Docker-Compose, hardening flags





- ▶ Internship of Victor Huesca at Bootlin in summer 2019, mentored by Thomas Petazzoni
- ▶ Topic: improve the Buildroot maintenance tooling
 - ▶ Use of `release-monitoring.org` for tracking upstream releases
 - ▶ Improved Buildroot developer notifications
 - ▶ Improved autobuilder search capabilities





Tooling: use of `release-monitoring.org`

- ▶ `release-monitoring.org` is a service from the Fedora community that tracks upstream releases of open-source projects
- ▶ Currently tracks 27000+ projects
- ▶ Buildroot has 2500+ packages, difficult to make sure they are all kept up-to-date
- ▶ Buildroot `pkg-stats` script produces a table of statistics about Buildroot packages, especially current version vs. upstream version
- ▶ Improvements:
 - ▶ Lots of *mappings* between Buildroot packages and `release-monitoring.org` packages added
 - ▶ Fixes to Buildroot packages for the package version to match better
 - ▶ JSON output in addition to HTML output (useful for tooling, next slide)
 - ▶ Speed improvement



Project: busybox

busybox [Flag](#) [Edit](#)

Latest version
1.31.0

Homepage:
<http://www.busybox.net>

Backend:
custom

Version scheme:
RPM

Version check url:
<http://www.busybox.net/downloads/>

Mappings

Distribution	Package name	
Fedora	busybox	Edit
Alpine	busybox	Edit
NixOS	busybox	Edit
Arch Linux	busybox	Edit
Arch Linux	mkinitcpio-busybox	Edit
Cygwin	busybox	Edit
Buildroot	busybox	Edit

Status

Status	Updated	Description
OK	2019-10-18 08:55:56 (UTC)	No new version found

Versions

Version	Retrieved on (UTC)
1.31.0	2019-06-10 11:03
1.30.1	2019-02-14 15:02
1.30.0	2019-01-07 12:11
1.29.3	Date information unavailable
1.29.2	Date information unavailable
1.29.1	Date information unavailable
1.29.0	Date information unavailable
1.28.4	Date information unavailable
1.28.3	Date information unavailable
1.28.2	Date information unavailable
1.28.1	Date information unavailable
1.28.0	Date information unavailable
1.27.2	Date information unavailable
1.27.1	Date information unavailable
1.27.0	Date information unavailable
1.26.2	Date information unavailable
1.26.1	Date information unavailable
1.26.0	Date information unavailable



CVE checking

- ▶ Goal: ensure Buildroot has fixes for the CVEs reported on all its packages
- ▶ The *NIST* provides the **National Vulnerability Database** (NVD) containing all known CVEs
- ▶ Buildroot `pkg-stats` has been improved to report CVE associated to each package, depending on the package version.
- ▶ New variable `<pkg>_IGNORE_CVES` that allows to ignore CVEs if they are fixed locally by a patch.

```
# debian/patches/length-check.patch  
LIBMAD_IGNORE_CVES += CVE-2017-8374
```

- ▶ Allows package maintainers to know when they have CVEs to look at and update the package accordingly





package/ccache/ccache.mk	0	autotools host	Yes	Yes	Yes	3.7.9	3.7.9 found by distro	0	Link	
package/ccid/ccid.mk	0	autotools target	Yes	Yes	Yes	1.4.31	1.4.32 found by distro	0	Link	
package/ccrypt/ccrypt.mk	0	autotools target	Yes	Yes	Yes	1.11	1.11 found by distro	0	Link	
package/cctz/cctz.mk	0	cmake target	Yes	Yes	Yes	2.3	2.3 found by distro	0	invalid 429	
package/cdrkit/cdrkit.mk	2	cmake target + host	Yes	Yes	Yes	1.1.11	1.1.11 found by distro	0	Link	
package/cegui/cegui.mk	2	cmake target	Yes	Yes	Yes	0-8-7	0-8-7 found by distro	0	Link	
package/cereal/cereal.mk	0	cmake target	Yes	Yes	Yes	1.3.0	1.3.0 found by distro	0	invalid 429	CVE-2020-11104 CVE-2020-11105
package/cgic/cgic.mk	3	generic target	Yes	Yes	Yes	2.07	2.07 found by distro	0	Link	
package/cgilua/cgilua.mk	0	luarocks target	Yes	Yes	Yes	5.2.1-1	5.2.1-1 found by distro	0	Link	



Tooling: improved developer notifications

- ▶ Buildroot has a `DEVELOPERS` file, associating developers with packages, defconfigs, architectures, tests they maintain
- ▶ Already used to send notifications of build failures reported by the Buildroot autobuilders
- ▶ Notification e-mail has been improved with:
 - ▶ Notification about packages that are not up-to-date, according to release-monitoring.org
 - ▶ Notification about packages that have unfixed CVEs according to the NVD database
 - ▶ Failures in the build of defconfigs in Gitlab CI
 - ▶ Failures in the execution of runtime tests in Gitlab CI





Tooling: improved developer notifications

===== Packages having a newer version

name	found by	link to release-monitoring.org	version	upstream	orph?
acpica	DISTRO	https://release-monitoring.org/project/00018	20190703	20190816	
acsccid	DISTRO	https://release-monitoring.org/project/15661	1.1.4	1.1.7	
adwaita-icon-theme	DISTRO	https://release-monitoring.org/project/13117	3.22.0	3.34.0	
aespipe	DISTRO	https://release-monitoring.org/project/21320	2.4e	2.4f	ORPH
alljoyn	DISTRO	https://release-monitoring.org/project/21665	16.04.00a	16.10.02	
alljoyn-tcl	DISTRO	https://release-monitoring.org/project/21666	16.04.00a	16.10.02	
android-tools	GUESS	https://release-monitoring.org/project/13989	4.2.2+git...	10.0.0_r5	
armadillo	DISTRO	https://release-monitoring.org/project/07006	7.900.1	9.800.1	
assimp	DISTRO	https://release-monitoring.org/project/06988	4.1.0	5.0.0	
asterisk	DISTRO	https://release-monitoring.org/project/09838	16.5.1	16.6.0	
at-spi2-atk	DISTRO	https://release-monitoring.org/project/07840	2.26.2	2.34.1	
at-spi2-core	DISTRO	https://release-monitoring.org/project/07841	2.28.0	2.34.0	
atk	DISTRO	https://release-monitoring.org/project/00130	2.33.3	2.35.1	ORPH
atkmm	DISTRO	https://release-monitoring.org/project/07962	2.24.2	2.29.1	

===== Packages having CVEs

name	CVE	link
argus	CVE-2011-3332	https://security-tracker.debian.org/tracker/CVE-2011-3332
bash	CVE-2019-18276	https://security-tracker.debian.org/tracker/CVE-2019-18276
binutils	CVE-2019-12972	https://security-tracker.debian.org/tracker/CVE-2019-12972
binutils	CVE-2019-14250	https://security-tracker.debian.org/tracker/CVE-2019-14250
binutils	CVE-2019-14444	https://security-tracker.debian.org/tracker/CVE-2019-14444



Tooling: improved developer notifications

Detail of defconfig failures

defconfig	link to the job	orph?
beaglebone_qt5	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105145	
engicam_imx6qdl_icore_qt5	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105157	
imx6-sabreauto	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105190	
minnowboard_max-graphical	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105208	
qemu_riscv32_virt	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105337	
raspberrypi3_qt5we	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105367	

Detail of runtime-test failures

runtime-test	link to the job	orph?
TestGlxinfo	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105533	ORPH
...ystemSystemdRoIfupdown	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105519	ORPH
...ystemSystemdRoNetworkd	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105521	ORPH
...ystemSystemdRwIfupdown	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105523	ORPH
...ystemSystemdRwNetworkd	https://gitlab.com/buildroot.org/buildroot/-/jobs/318105525	ORPH



Tooling: improved autobuilder search capabilities

- ▶ `autobuild.buildroot.org` collects results from random builds executed by our autobuilders
- ▶ Testing random configurations 24/7, allows to detect numerous dependency problems, version compatibility issues, toolchain problems, and more.
- ▶ Running this testing effort for many years
- ▶ Improvement:
 - ▶ Can now search through build results by config symbol
 - ▶ Ex: *what are the successful builds that had `BR2_PACKAGE_BUSYBOX=y` on ARM, with `uClibc` ?*
 - ▶ Very useful tool when analyzing build issues, and trying to understand in which situations it happens / since when





Other smaller improvements

- ▶ Addition of a `make <pkg>-diff-config` target for *kconfig* based packages: Linux, U-Boot, Busybox, etc.
 - ▶ Shows the difference between the current package configuration and the one that is stored in the Buildroot configuration
- ▶ Support for generating rootfs images in F2FS, BTRFS and EROFS formats
- ▶ Support for *gettext-tiny* as an alternative to full blown *GNU Gettext*
 - ▶ Smaller footprint, smaller build time, for cases where native language support is not needed



Conclusion

- ▶ Very active project
- ▶ LTS release, 1 year maintenance for security/bug fixes
- ▶ New CPU architectures
- ▶ Package infrastructures for new build systems
- ▶ Git caching
- ▶ Packages kept up-to-date, many new packages
- ▶ Top-level parallel build in good progress
- ▶ Reproducible builds effort in progress
- ▶ Maintenance tooling improvements

Online Buildroot training with Thomas Petazzoni, 16 hours to dive into Buildroot, July 28-31, 2020. Register at <https://bootlin.com> !

Questions? Suggestions? Comments?

Thomas Petazzoni
thomas.petazzoni@bootlin.com

Slides under CC-BY-SA 3.0

<https://bootlin.com/pub/conferences/2020/elc/petazzoni-buildroot-whats-new>