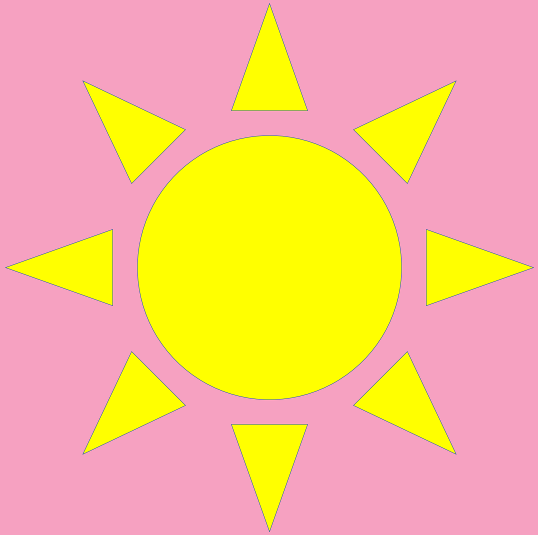


Chimera Linux

A BSD/LLVM distro from scratch

Who?

- Developer at Igalia @ WebKit team
- FOSS contributor since 2007
- FreeBSD user since 2009
- POWER architecture maintainer at Void Linux
- Amateur gamedev

What is Chimera Linux?

- A new Linux distribution
- Not too interesting?
- Created from scratch
- FreeBSD-based userland
- Built fully with LLVM/Clang

What is Chimera Linux?

- Musl libc
- Rolling release
- Highly portable (ppc64le, aarch64, x86_64...)
- Bootstrappable
- Custom source package build system

What is Chimera Linux?

- Apk-tools for binary packages
- Lightweight service management (dinit)
- General purpose, graphical desktop
- Wayland, PipeWire, curated main/ repository
- Avoid legacy stuff when possible

Motivations

- Have some fun and start a community
- Explore alternatives, increase diversity
- Improve software portability
- Make a distro I will be happy with
- Learn from others’ mistakes, fix them

Motivations

- Prove Linux != GNU/Linux
- LLVM is a great toolchain
- Sanitizers, exploit mitigations (CFI)
- Actual usable LTO (ThinLTO)
- Superior cross-compiling support

Principles

- Not minimalist/”suckless”
- Not reactionary
- Not traditionalist
- If it sucks, get rid of it
- Systemd is not the root of all evil

Principles

- But technical debt is
- Be strict by default; enforce best practices
- There should be one obvious way to do it
- Be self-sustaining; good tooling is important
- Portability is not a joke

Principles

- Neither is bootstrapping
- It should probably not be written in shell
- Good things should be easy to do
- Bad things should be a pain in the ass
- Documentation is important

Principles

- Opinionated development is good
- Fun environment is an important thing
- There is no fun without good community
- At least not in the long term
- Meritocracy and technical-only spaces are crap

Early history

- Idea: ~2015 (Linux with BSD userland)
- Development start: May 2021
- Early June: first working prototype, <50 templates
- Reimagination of xbps-src from Void
- Using GCC, GNU userland, xbps

Early history

- Cports: “ports” tree with integrated build system
- Written in Python (package templates too)
- Designed to be fast (no buildsystem overhead)
- Namespaces for strict sandboxing
- Fully unprivileged

Example template

```
pkgname = "libpng"
pkgver = "1.6.37"
pkgrel = 0
build_style = "gnu_configure"
hostmakedepends = ["pkgconf"]
makedepends = ["zlib-devel"]
pkgdesc = "Library for manipulating PNG images"
maintainer = "q66 <q66@chimera-linux.org>"
license = "Libpng"
url = "http://www.libpng.org/pub/png/libpng.html"
source = f"$(SOURCEFORGE_SITE)/{pkgname}/{pkgname}-{pkgver}.tar.xz"
sha256 = "505e70834d35383537b6491e7ae8641f1a4bed1876dbfe361201fc80868d88ca"

def post_install(self):
    self.install_license("LICENSE")

@subpackage("libpng-devel")
def _devel(self):
    return self.default_devel()

@subpackage("libpng-progs")
def _progs(self):
    return self.default_progs()
```

First steps

- Drop coreutils and related
- Project: bsdutils; bare and semi-functional
- Help port remaining tools for coreutils parity
- Additional new ports:
- diff, grep, sed, ed, patch, m4, gzip

Porting BSD tools

- Code surprisingly clean and portable
- A few BSDisms shared between all code
- No REG_STARTEND in musl...
- tail(1): replace kqueue with inotify
- Otherwise pretty uneventful

Dropping xbps

- Original plan: use FreeBSD’s pkg
- Turned out not entirely ready for our use
- Alpine Linux’s apk proved a great fit
- Quick integration; robust by mid July
- Custom package generation code for now

Coreutils out, apk in

- Mid June: GNU userland mostly gone
- Still using GCC (and xbps for now)
- Late mid June: initial code for apk generation
- Early late June: xbps + coreutils out, apk fully in
- Why apk?

Benefits of apk-tools

- Lightweight but surprisingly elegant
- Transactional and constraint-based
- Robust dependency solver
- Support for triggers, virtual packages
- But about that GCC…

Dropping GCC

- Fairly easy: not much packaged yet
- Add standard LLVM build
- Tell it use its own runtime instead of GCC’s
- Recompile everything with it; fix errors (few)
- Finally remove GCC packaging (end of June)

Cross-compiling with GCC

- Target triplets and cross-toolchains
- Build: machine compiler runs on
- Host: machine compiled stuff runs on
- Target: machine compiled stuff works for
- Host == target most of the time

Cross-compiling with GCC

- Separate cross-toolchain for each host
- Binutils (linker, assembler, etc)
- Build “freestanding” compiler
- Build libc (musl) with it
- Build “final” compiler

Cross-compiling with GCC

- That means many compilers, one per target
- Tell builds what to use: e.g. <triplet>-gcc
- Kind of clunky and complicated
- Requires very specific infra
- Can we make it better?

Cross-compiling with LLVM

- One compiler for everything!
- One build of Clang targets every target
- Still need a “cross-toolchain”
- Compiler-rt builtins, musl, libunwind, libc++
- Slighly tricky to build

Cross-compiling with LLVM

- Compiler-rt needs libc headers to build
- Temporary header-only musl
- Build compiler-rt core bits: static only
- Add into sysroot; build musl with them
- Build libunwind (with --unwindlib=none)

Cross-compiling with LLVM

- Build libc++abi (with -nostdlib CXXFLAGS)
- Build libc++ (same)
- Build rest of compiler-rt (sanitizers)
- Tell Clang the -target + --sysroot
- Every arch built at once

Integrating into cbuild

- Separate host and target build dependencies
- Install host deps normally + cross-toolchain
- Install target dependencies with apk
- Treat cross sysroot as target apk’s root
- Dummy base package to provide virtual root

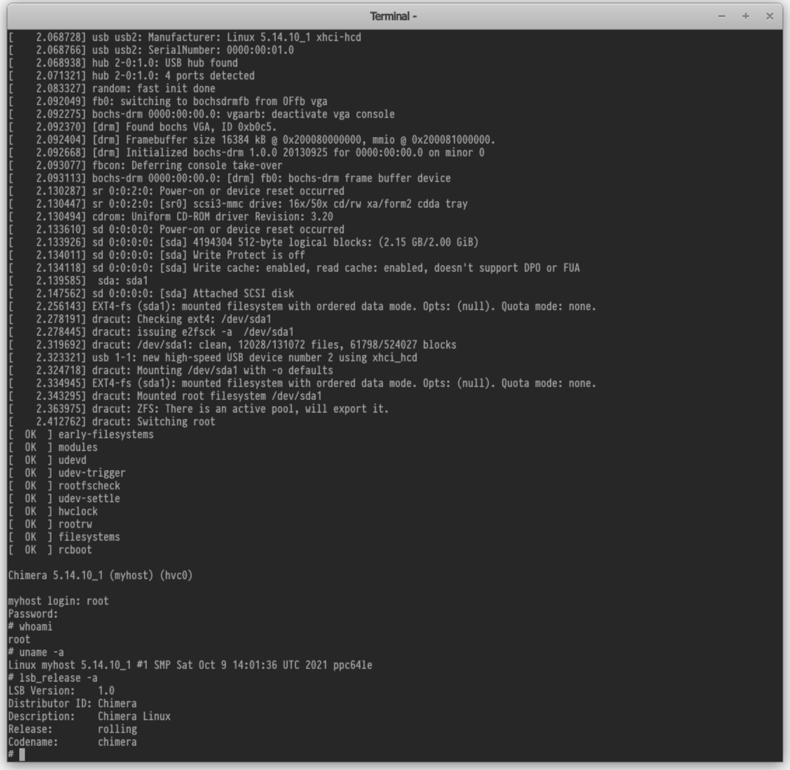
Integrating into cbuild

- In theory that is everything
- In practice various workarounds are needed
- Mostly handled transparently in cbuild
- Ensure correct flags are exported, etc.
- Proper handling of host/target build profiles

Improving cbuild

- Run unit tests for everything out of box
- Sandboxing: disallow network access
- Sandboxing: read-only container root
- Sanitize container environment
- Prevent breakout to outside system

First boot



```
[    2.068728] usb usb2: Manufacturer: Linux 5.14.10_1 xhci-hcd
[    2.068766] usb usb2: SerialNumber: 0000:00:01.0
[    2.068938] hub 2-0:1.0: USB hub found
[    2.071321] hub 2-0:1.0: 4 ports detected
[    2.083327] random: fast init done
[    2.092049] fb0: switching to bochsdrmfb from OFfb vga
[    2.092275] bochs-drm 0000:00:00.0: vgaarb: deactivate vga console
[    2.092370] [drm] Found bochs VGA, ID 0xb0c5.
[    2.092404] [drm] Framebuffer size 16384 kB @ 0x200080000000, mmio @ 0x200081000000.
[    2.092668] [drm] Initialized bochs-drm 1.0.0 20130925 for 0000:00:00.0 on minor 0
[    2.093077] fbcon: Deferring console take-over
[    2.093113] bochs-drm 0000:00:00.0: [drm] fb0: bochs-drm frame buffer device
[    2.130287] sr 0:0:2:0: Power-on or device reset occurred
[    2.130447] sr 0:0:2:0: [sr0] scsi3-mmc drive: 16x/50x cd/rw xa/form2 cdda tray
[    2.130494] cdrom: Uniform CD-ROM driver Revision: 3.20
[    2.133610] sd 0:0:0:0: Power-on or device reset occurred
[    2.133926] sd 0:0:0:0: [sda] 4194304 512-byte logical blocks: (2.15 GB/2.00 GiB)
[    2.134011] sd 0:0:0:0: [sda] Write Protect is off
[    2.134118] sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[    2.139585]  sda: sda1
[    2.147562] sd 0:0:0:0: [sda] Attached SCSI disk
[    2.256143] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts: (null). Quota mode: none.
[    2.278191] dracut: Checking ext4: /dev/sda1
[    2.278445] dracut: issuing e2fsck -a  /dev/sda1
[    2.319692] dracut: /dev/sda1: clean, 12028/131072 files, 61798/524027 blocks
[    2.323321] usb 1-1: new high-speed USB device number 2 using xhci_hcd
[    2.324718] dracut: Mounting /dev/sda1 with -o defaults
[    2.334945] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts: (null). Quota mode: none.
[    2.343295] dracut: Mounted root filesystem /dev/sda1
[    2.363975] dracut: ZFS: There is an active pool, will export it.
[    2.412762] dracut: Switching root
[  OK  ] early-filesystems
[  OK  ] modules
[  OK  ] udevd
[  OK  ] udev-trigger
[  OK  ] rootfscheck
[  OK  ] udev-settle
[  OK  ] hwclock
[  OK  ] rootrw
[  OK  ] filesystems
[  OK  ] rcboot

Chimera 5.14.10_1 (myhost) (hvc0)

myhost login: root
Password:
# whoami
root
# uname -a
Linux myhost 5.14.10_1 #1 SMP Sat Oct 9 14:01:36 UTC 2021 ppc64le
# lsb_release -a
LSB Version:    1.0
Distributor ID: Chimera
Description:    Chimera Linux
Release:        rolling
Codename:       chimera
# 
```

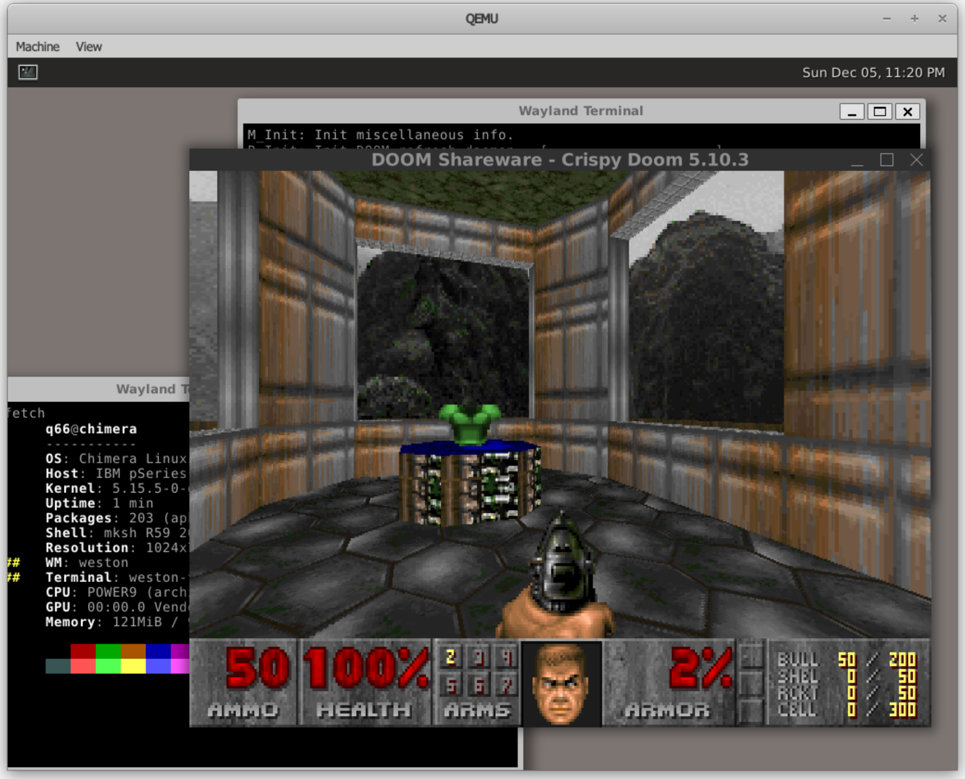
First boot

- Early October
- Linux kernel 5.14/15; some build workarounds
- Finally added init system and set it up
- Added initramfs generator – initramfs-tools
- Ported from Debian; surprisingly easy

Towards graphical system

- Increasing packaging pace
- Weston compositor by early November
- In cbuild: added update-check
- Checks upstreams for new versions
- Automatic; crucial for keeping things updated

Yes, it runs DOOM



```
M_Init: Init miscellaneous info.
```

```
fetch
     q66@chimera
     -----------
     OS: Chimera Linux
     Host: IBM pSeries
     Kernel: 5.15.5-0-
     Uptime: 1 min
     Packages: 203 (ap
     Shell: mksh R59 2
     Resolution: 1024x
##   WM: weston
##   Terminal: weston-
     CPU: POWER9 (arch
     GPU: 00:00.0 Vend
     Memory: 121MiB /
```

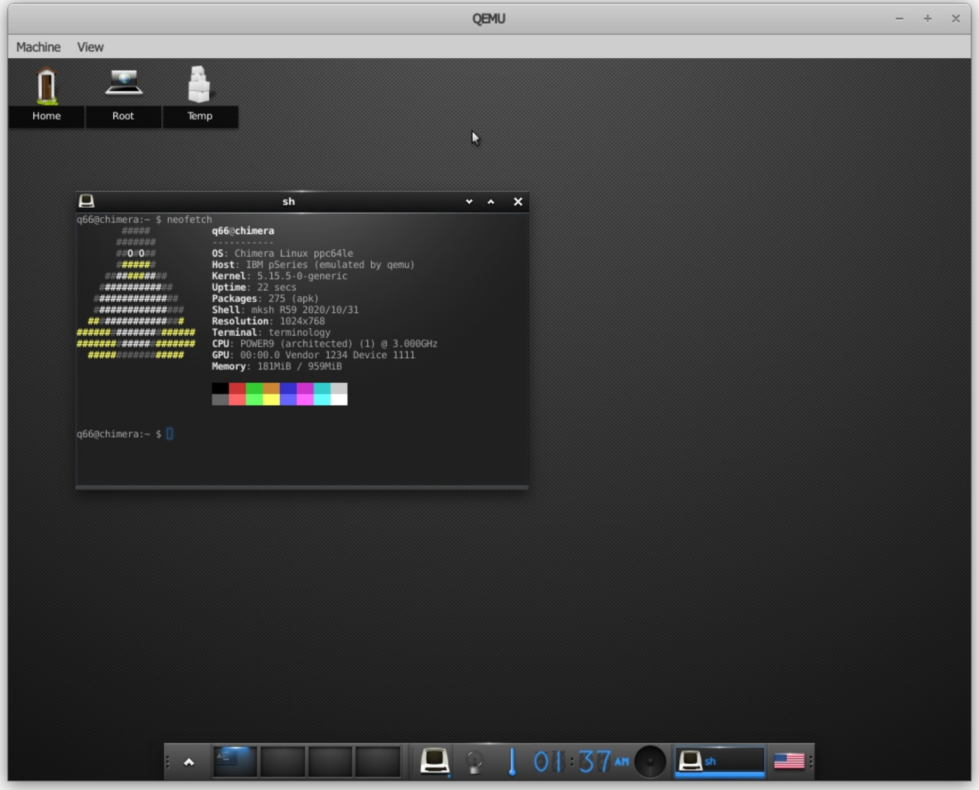
DOOM

- Early December; milestone achieved
- Also turned on systemwide LTO
- Added GRUB bootloader
- Added PipeWire sound server
- User services

User services

- Automatic user instances of dinit via PAM
- Packages can install user services
- D-Bus session bus: not ad-hoc anymore
- Also PipeWire, WirePlumber
- Future expansion

Xorg



```
q66@chimera:~ $ neofetch
        #####           q66@chimera
       #######          -----------
       ##O#O##          OS: Chimera Linux ppc64le
       #######          Host: IBM pSeries (emulated by qemu)
     ###########        Kernel: 5.15.5-0-generic
    #############       Uptime: 22 secs
   ###############      Packages: 275 (apk)
   ################     Shell: mksh R59 2020/10/31
  #################     Resolution: 1024x768
#####################   Terminal: terminology
#####################   CPU: POWER9 (architected) (1) @ 3.000GHz
  #################     GPU: 00:00.0 Vendor 1234 Device 1111
                        Memory: 181MiB / 959MiB



q66@chimera:~ $
```

Xorg

- Added mid December
- More packaging: Gtk+3
- Desktops: Enlightenment, pekwm
- Basic input, video drivers
- Rust toolchain

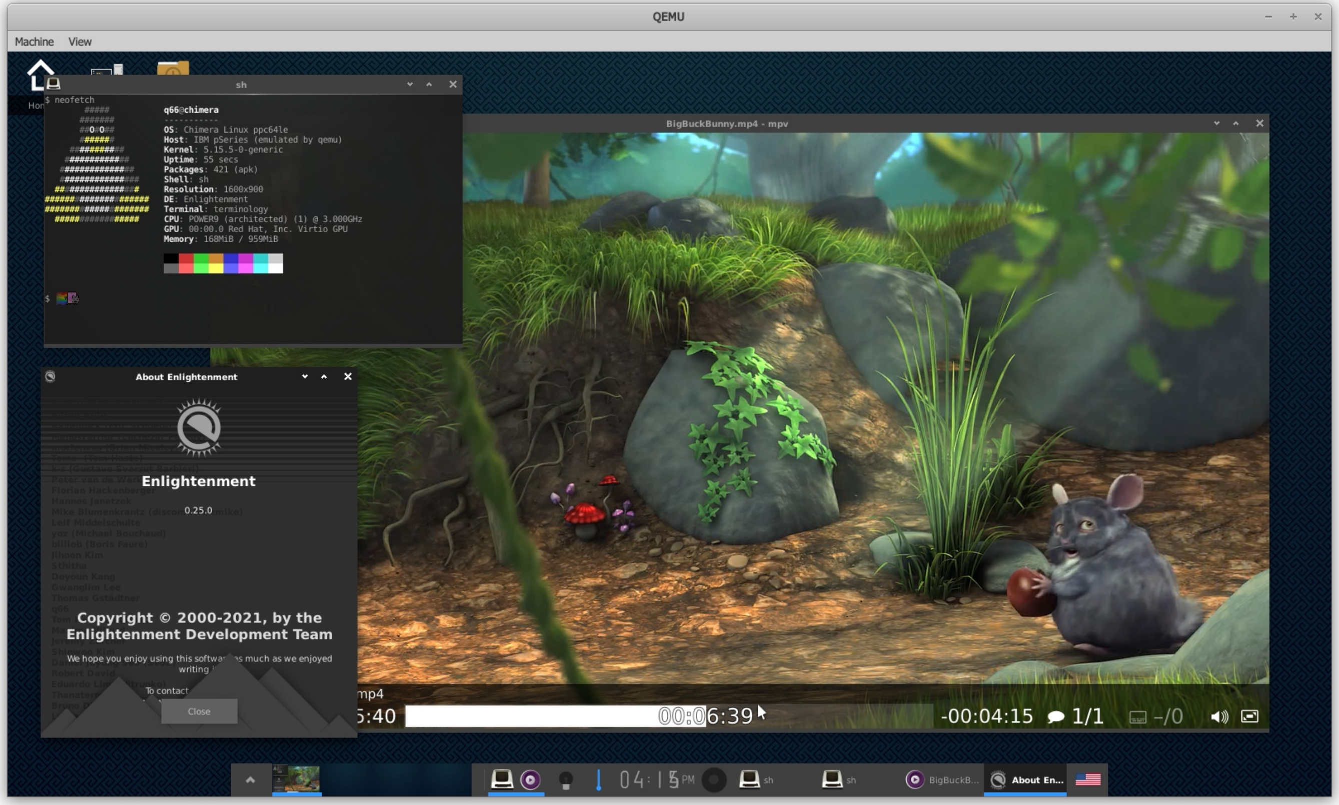
Rust

- Needed for librsvg
- Major bootstrap pains
- Official x86_64-musl binaries sort of worked
- Symlink libgcc_s.so to libunwind.so
- Enough to recompile for our environment

Rust

- Major patching needed
- Added custom vendor triplets
- Generated custom bootstrap binaries
- Cross-compiled for ppc64le, aarch64
- Bootstrapped natively using those

Multimedia



```
$ neofetch
q66@chimera
-----------
OS: Chimera Linux ppc64le
Host: IBM pSeries (emulated by qemu)
Kernel: 5.15.5-0-generic
Uptime: 55 secs
Packages: 421 (apk)
Shell: sh
Resolution: 1600x900
DE: Enlightenment
Terminal: terminology
CPU: POWER9 (architected) (1) @ 3.000GHz
GPU: 00:00.0 Red Hat, Inc. Virtio GPU
Memory: 168MiB / 959MiB

$
```

Enlightenment

0.25.0

Copyright © 2000-2021, by the Enlightenment Development Team

We hope you enjoy using this softwa as much as we enjoyed writing i

To contact

Close

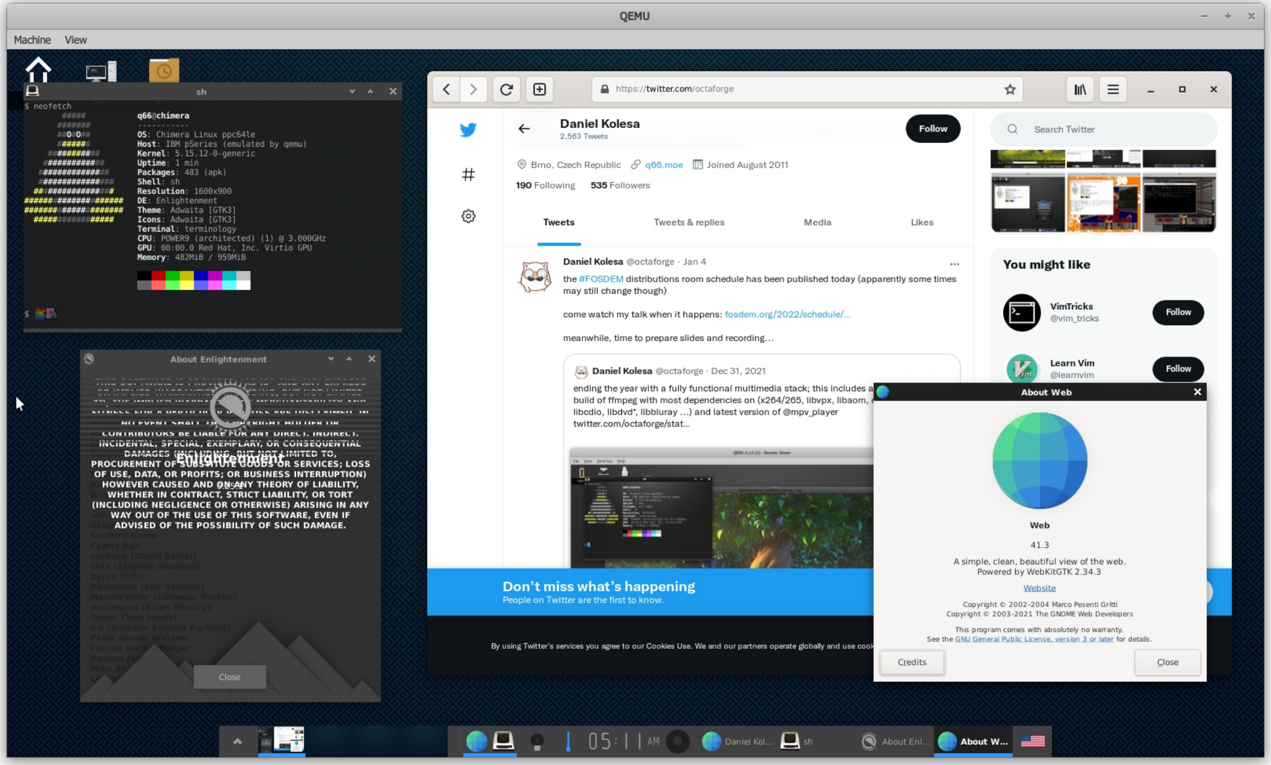
Multimedia

- Made it just before year end
- Complete build of ffmpeg and dependencies
- Mpv media player
- Almost usable desktop system?
- Still need a web browser

Packaging a browser

- OpenSSL bump: 3.0.1
- Initial pieces of GNOME
- WebKitGTK and dependencies
- Epiphany
- Total template count: 500+

Packaging a browser



```
$ neofetch
          #####               q66@chimera
         #######              -----------
         ##O#O##              OS: Chimera Linux ppc64le
         #######              Host: IBM pSeries (emulated by qemu)
       ###########            Kernel: 5.15.12-0-generic
      #############           Uptime: 1 min
     ###############          Packages: 483 (apk)
     ################         Shell: sh
   ##################         Resolution: 1600x900
####################          DE: Enlightenment
#####################         Theme: Adwaita [GTK3]
   #################          Icons: Adwaita [GTK3]
                              Terminal: terminology
                              CPU: POWER9 (architected) (1) @ 3.000GHz
                              GPU: 00:00.0 Red Hat, Inc. Virtio GPU
                              Memory: 482MiB / 959MiB

$
```

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Stafford Horne
Cedric Bail
onefang (David Seikel)
okra (Stephen Houston)
Byron Hillis
Ravenlock (Eric Schuele)
ManoWarrior (Luchezar Petkov)
morlenxus (Brian Miculcy)
Toma- (Tom Haste)
k-s (Gustavo Sverzut Barbieri)
Peter van de Werken

Daniel Kolesa
2,563 Tweets

Brno, Czech Republic · q66.moe · Joined August 2011

190 Following · 535 Followers

Daniel Kolesa @octaforge · Jan 4

the #FOSDEM distributions room schedule has been published today (apparently some times may still change though)

come watch my talk when it happens: fosdem.org/2022/schedule/...

meanwhile, time to prepare slides and recording...

Daniel Kolesa @octaforge · Dec 31, 2021

ending the year with a fully functional multimedia stack; this includes a build of ffmpeg with most dependencies on (x264/265, libvpx, libaom, libcdio, libdvd*, libbluray ...) and latest version of @mpv_player twitter.com/octaforge/stat...

You might like

VimTricks @vim_tricks — Follow

Learn Vim @learnvim — Follow

Don't miss what's happening
People on Twitter are the first to know.

By using Twitter's services you agree to our Cookies Use. We and our partners operate globally and use cook

Web

41.3

A simple, clean, beautiful view of the web.
Powered by WebKitGTK 2.34.3

Website

Copyright © 2002-2004 Marco Pesenti Gritti
Copyright © 2003-2021 The GNOME Web Developers

This program comes with absolutely no warranty.
See the GNU General Public License, version 3 or later for details.

Future

- Migrate to apk-tools 3.x
- Release official binary repositories (and set up CI)
- Package GNOME desktop
- Extend user services framework further
- Migrate my own systems to it

Future

- Improve system documentation
- Enable Clang CFI, UBSan on eligible targets
- Investigate a _FORTIFY_SOURCE implementation
- Integrate non-stub locale support
- Integrate non-stub utmp/wtmp?

Thanks for listening!

- https://chimera-linux.org
- https://github.com/chimera-linux
- #chimera-linux @ OFTC (irc.oftc.net)
- #chimera-linux:matrix.org