

FOSDEM 2014

Maksim Melnikau



ULB

We are recording everything!



- The video team could give you:
- Camera operators
 - Video analysis
 - Audio capture

Presented by FOSDEM video team
Thank you for the video recording of the event



How Find and Fix Million Grammar and Style Errors in Wikipedia

- ▶ Wikipedia uses LanguageTool(LT) to find grammar errors
- ▶ LT - the next step after spell checking, LGPL, 10 regular committers, Java+XML
- ▶ finds errors, but explanation sometimes wrong
- ▶ 68000 assembler (suggest: assemblers), if a is algebraic over K(suggest: an)
- ▶ LT: plain text => sentences => words => find part-of-speech and base form
=> analyze sentences against error patterns
- ▶ LT patterns are easy to contribute in XML format, no Java skills required
- ▶ LT supports many different languages including Russian and Belorussian
- ▶ grammar — rules that describe how valid words, sentences, and texts looks like
- ▶ *Sorry for my bed English* grammatically is fine...
- ▶ <http://community.languagetool.org/feedMatches/list?lang=en>
- ▶ no need to stick to spell checking today - more powerful checks are available

kdbus, Lennart Poettering

- ▶ D-Bus is powerful IPC: method call transactions, signals, properties, broadcasting, discovery, introspection, policy, activation, security, monitoring, expose APIs, File Description passing, language agnostic
- ▶ D-Bus has limitations: suitable only for control, not payload, inefficient; not available in early boot, initrd; baroque codebase
- ▶ if you try to solve problem with XML, you have two problems
- ▶ but still, D-Bus is fantastic, solves real problems
- ▶ kdbus suitable for large data (GiB!), zero-copy, optionally reusable, implicit timestamping; always available; no XML...
- ▶ 2 previous tries to get D-Bus in kernel grandiosely failed

miracast on Linux

- ▶ miracast: HDMI over IP over Wifi
- ▶ ieee 802.11; wifi-p2p => wifi direct; wifi-display => miracast
- ▶ miracast: P2P transport setup, ip link auto discovery, A/V streams
- ▶ mirascast: many Linux wifi drivers not working (b43, brcmac, rtl818x, ath5k)
- ▶ some supposed to work (ath9k, brcmfmac, iwl-mvm)...
- ▶ known to work: iwl+mwm + intel wifi 7260 + wpa_supplicant: git-78f79 ...
- ▶ HDMI over IP is RTSP + RTP + h264 + audio + mpeg2-TS
- ▶ Additional Features: PTP, HDCP, UIBC, split-sink

Sailfish and Jolla

- ▶ half people at sailfish talk have Jolla device already
- ▶ Jolla: recovery mode, fastboot, unlocked bootloader, flash own kernels, full root
- ▶ sailfishos: systemd, gcc, btrfs, gstreamer, Wayland, qt5
- ▶ Jolla not contribute to: L&F UI, 3rd party closed source drivers, some NDA stuff
- ▶ <https://together.jolla.com/questions/>
- ▶ contribute to sailfishos: contribute to nemo, mer, and a lot of upstream projects!
- ▶ libhybris - leverage existing Android hardware adaptation
- ▶ libhybris - port Android/bionic linker to glibc environment
- ▶ load glibc and bionic to address space of process — works for almost all cases
- ▶ `android__dlopen("libEGL.so")` - we could wrappers that accessed the android ones
- ▶ libhybris today used by Jolla/SailfishOS, Intel/Tizen, Canonical/Ubuntu

Fedora.Next

- ▶ Fedora.Next split to Workstation, Server, Cloud
- ▶ Fedora Workstation — GUE for Students, Developers, etc ...
- ▶ Fedora Server — headless *pet* server, server roles, IaaS Host, stable platform for critical infrastructure
- ▶ Fedora Cloud — cloud image *cattle* server, scale-out, packaged images for clouds
- ▶ Fedora has so many infrastructure problems: bugs, reviews, build system, etc ...

FOSDEM network, NAT64 and DNS64

- ▶ FOSDEM had ipv6-only wifi network by default
- ▶ but too many people escaped to fosdem-dualstack
- ▶ IPv4 has run out, IPv5 never made it to public use, so IPv6
- ▶ there was a war in begging of IPv6: 64bit vs unlimited!
- ▶ clients, content, carriers, applications, hardware - nobody want to do first step
- ▶ World IPv6 day - lets turn it on and see what breaks
- ▶ Google, facebook, yahoo, youtube, netflix, akamai and many more run ipv6 today
- ▶ different countries enables ipv6 one by one - France, Germany, Belgium etc
- ▶ if you run NAT anyway - why not unale IPv6 and use NAT64 and DNS64 ?!
- ▶ we can hide a complete legacy internet in a /96!
- ▶ Nexus(Android) could not get ipv6 only address
- ▶ FOSDEM'14 — first general-purpose conference which has ipv6 network by default

KDE Connect

- ▶ fuse your devices as much as possible and desirable
- ▶ protocol: json based, medium abstracted, easy extended, easy implemented
- ▶ notifications, actions, battery, MPRIS2, send files and urls
clipboard synchronization, encryption, ...
- ▶ Qt => libconnect => server => plugins => D-Bus => plasma, kcm, apps

GPU Offload on Wayland

- ▶ render-nodes — render without authentication to DRM master
- ▶ 1080p buffer with 60fps is 480MB/s, PCI express is 4GB/s, thunderbolt is 1GB/s
- ▶ tiling — special pixel ordering optimized to exploit local spatial coherence — good for performance
- ▶ GPU offload with X DRI2: DDX per device/provider, configure with xrandr
- ▶ two displays: A and B, two cards: 1 connected to A, 2 connected to B — classic nvidia optimus layout
- ▶ wayland gpu offload: shutdown the dedicated GPU when unneeded works now
- ▶ XWayland: wlgamor, X linked to Wayland compositor - no need for gpu offloading

Wine User Experience

- ▶ once a year somebody writes at wine forum what "everything is work, you rock!"
- ▶ Ubuntu still ships 1.4.x wine version, why?!
- ▶ common problem when you answer to user question:
if user hides, you don't know why: does everything work, user give up or died ...

Performance of Wine and Graphical Drivers

- ▶ command stream multi thread (CSMT) - move most d3d work into separate thread
- ▶ better CPU utilization => 2x performance (in theory)
- ▶ easy synchronization in multi-threaded games, even bigger performance gains 3x in CoD 4:MW, btw Windows does the same thing
- ▶ wine CSMT improvements - some games faster on wine than on windows
- ▶ CSMT brings better performance mostly on fast systems like 460gtx + i7 ...
- ▶ drivers don't like to be called from two threads without looking even with separate contexts
- ▶ CSMT and NVIDIAS's threaded opt - essentially the same thing
- ▶ CSMT wine next steps: upstream, improve data streaming, reduce draw overhead in wine, wine performance outside d3d ...
- ▶ wine could have a big problems running on wayland natively — many windows apps rely on window positions for example

Persistent Storage

- ▶ file system performance: maximize throughput or latency?
target embedded, power consumption or performance?
- ▶ high bandwidth has been the traditional focus — backup, streaming video, etc
- ▶ SSD's made life more complicated — not too painful at first, plagued in
- ▶ PCI-e SSD devices turn up the heat
opened a lot of other bottlenecks in Linux Storage Stack — 1 million IOPS/device
- ▶ a single file system is easy for users and applications, and can perform better
- ▶ wow! fsck which works more than week!
- ▶ persistent memory - a variety of new technologies are coming from multiple vendors
— Linux need to be (mostly) technology neutral
- ▶ SNIA - Storage Network Industry Association, Working Group on NVM.
- ▶ SMR and PM together - interesting workload for out future
- ▶ normally block size limit is 4k, storage hardware often have very large - 64k
- ▶ PS — you will never ever wait for storage anymore, CPU will be bottleneck

Concurrent Programming Made Simple - Transaction Memory

- ▶ shared memory (synchronization) + Transactions = Transaction memory (TM)
- ▶ TM — programming abstraction, declare which code sequences are atomic
- ▶ TM is still rather new - standardization for C/C++ started 5 years go
GCC has support since 4.7, HW implementations — Haswell
- ▶ `__transaction_atomic { if (x<10) y++; }` - code must be transaction-safe
- ▶ extend the C11/C++11 memory model — all transactions totally ordered
- ▶ GCC: compiler — ensure atomicity guarantee (at compile time)
- ▶ GCC: TM runtime library (libitm) => enforces atomicity of transactions at runtime
(contains SW-only implementation)
- ▶ performance: it's a tool, not magic - **useful balance**, implementations are wip
- ▶ single-thread performance: STM slower than sequential, HTM equals
in multi-thread: both STM and HTM scales well
- ▶ TM, use it: `gcc -fgnu-tm`, report bugs and dive into libitm / GCC
- ▶ eventually consistency — is not consistency at all

Thank You. Questions

Maksim Melnikau

<mailto:maxposedon@gmail.com>

<https://plus.google.com/+MaksimMelnikau>

https://twitter.com/max_posedon

<https://fosdem.org/2014/>

<https://www.languagetool.org/>

<http://lwn.net/Articles/580194/>

<http://www.freedesktop.org/wiki/Software/openwfd/>

<http://jolla.com/>

<http://fedoraproject.org/wiki/Fedora.next>

<http://en.wikipedia.org/wiki/NAT64>

https://fosdem.org/2014/schedule/event/kde_connect/

<http://wayland.freedesktop.org/>

<http://www.winehq.org/>