Linux and gaming: the road to performance

André Almeida

Kernel Recipes 2023

Linux has no roadmap

- Users are motivated to send changes to fix bugs
- Companies add new features to build products
- It's driven by use cases
- End users can take part of it as well

igalia

Linux has no roadmap

- Cloud providers are pushing for better storage performance
- Android and ChromeOS for system responsiveness, better graphic stack
- Embedded products for better energy, memory and resources usage
- Developers push for tracing and debug tools

Linux has no roadmap

- Community and collaborative development
- Value and benefits for everyone
- Linux is a result of years of efforts

igalia

And what does gaming brings to the table?

igalia

Linux gaming

- People have been playing on Linux for years
- Mostly native titles for a long time
- Some ports here and there
- Lack of stable ABI make it hard to deploy
  - Different libc, OpenGL implementations, etc

igalia

Linux gaming

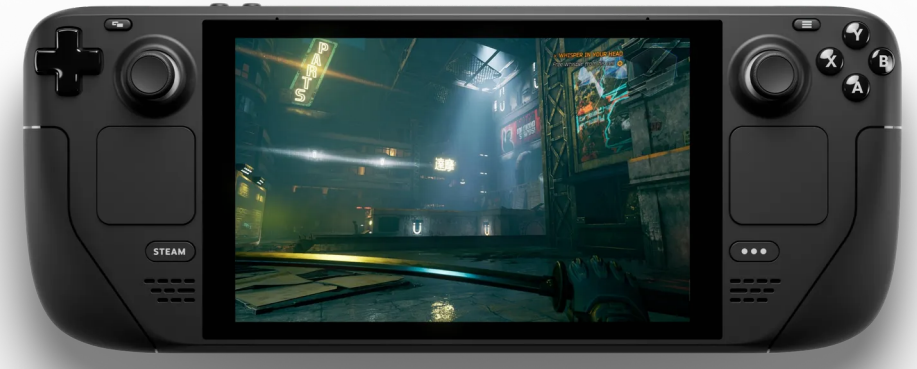
- Most games are developed for Windows
- Wine and PlayOnLinux
- A lot of manual work and tweaking, forums discussions
- Active community
- Unity for Linux, 2012

igalia

Linux gaming

- **Valve**
- Steam client for Linux, 2012
- Proton, 2018
- Steam Deck, 2022

igalia



igalia

Valve

- Sponsor the development of all the stack
- Linux kernel, Mesa, Wine, Gstreamer, etc
- DXVK, VKD3D
- Proton: all those projects together

igalia

Proton

- Bundle of projects + extra patches
- Tool to run Windows games on Linux
- Aim to delivery same or best performance
- **https://www.protondb.com/**

igalia

Wine

- Translate Win32 API calls to Linux calls
- Non Win32 calls runs natively, no overhead (x86)
- Black box reverse engineering
- Sometimes really straightforward, sometimes needs a lot of userspace workaround

igalia

```
NTSTATUS WINAPI NtQuerySystemTime(LARGE_INTEGER *ti
    clock_gettime(CLOCK_MONOTONIC, &ts);

    time->QuadPart = ticks_from_time_t(ts.tv_sec)
        + (ts.tv_nsec + 50) / 100;
}
```


igalia

Currently, the Linux kernel has seen a lot of gaming workloads

igalia

What was missing on Linux?

- Work done by different companies, including Igalia
- Non-exhaustive list

igalia

Case-insensitive filesystems

- NTFS is case insesitive, ext4 (and others) aren't
- Games try to find for `ASSETS/Image.png`, but there's just `assets/image.PNG`
- Userspace workaround: try all case combinations
- Kernel solution: normalize filenames for lookups
- Inside the fs: `Image.png`
- On lookup: `normalize("Image.png") == normalize("image.PNG")`

igalia

Case-insensitive filesystems

- Initial support for ext4, reused for F2FS (Android)
- WIP for bcachefs, tmpfs
- Work by Gabriel Krisman

igalia

btrfs' and identical fs IDs

- btrfs identifies a filesystem by its fsid
- It's not possible to mount two filesystems with the same ID
- A/B partitioning systems may have two identical FSs
- New `TEMP_FSID` feature to assign a random fsid
- Work by Guilherme Piccoli

igalia

futex

- Win32's `WaitForMultipleObjects()` called a lot by games
- Mapped to `eventfd()`, but proved to not scale well with a lot of waiters
- Futex was the natural solution...
- ... but it only supported waiting on one futex per call

igalia

futex2

- What initially was a new futex op, became a new futex interface
- Let's solve all old futex problems in a new API:
  - NUMA awareness
  - Multiple sizes futexes
  - Wait on multiple futexes
- Benefits for games, other apps and servers workloads
- Work by André Almeida, Peter Zijlstra, Thomas Gleixner et al.

igalia

Syscall user dispatch

- Usually games calls syscalls through Win32 wrappers, which is easily translated with Wine
- Some games calls directly using assembly

```
movl $202, %rax
...
syscall
```


igalia

Syscall user dispatch

- Windows and Linux doesn't share even the same syscalls
- It's easy to see how that can go wrong
- A new `prctl` option: `PR_SET_SYSCALL_USER_DISPATCH`

igalia

Syscall user dispatch

- Using `PR_SET_SYSCALL_USER_DISPATCH`, one can define an address range where all syscalls will be redirect to an userspace handler, that can do what's the appropriated translation
- Used by CRIU (Checkpoint/Restore In Userspace) as well
- Work by Gabriel Krisman

igalia

HID throughput

- Virtual reality sets requires a lot of peripherals
- That translates to a lot of HID devices requesting parallel ioctls
- A mutex was used to protect the HID device's table
- Only one HID ioctl operation per time was allowed, causing frame drops
- After replacing it with a RW lock, parallel HID ops got almost 4x faster
- Work by André Almeida

igalia

Split-lock detector handling

- x86 supports atomic ops in more than one cache line once
- This is really expansive as it needs to lock the bus during the whole operation
- Unprivileged apps may DoS a machine by doing this
- For security reasons, Linux then added a 10ms delay to split-lock apps, mitigating potential attacks
- Of course games uses this... and they slowed down too
- Guilherme Piccoli added a new sysctl option to

igalia

Adaptive spinlocks

- Userspace can't spin reliably, so most of locks are mutexes, which needs context switch
- Using rseq, we can implement a reliable spinlock in userspace, that only spins if the lock holder is running, and sleeps otherwise
- This would benefit a lot of workloads, including games
- WIP
- Work by Mathieu Desnoyers, André Almeida

Pagemap scan

- Games uses win32's `GetWriteWatch()` to check if the player had modified the memory of the game, likely cheating
- Linux soft dirty mechanism isn't so precise and fast as the Windows one
  - Merge of pages, mprotect() calls can create false positives
  - pagemap is slower
- New `PAGEMAP_SCAN` ioctl to walk in a selected range of pages, asking if it was written
- Also useful for garbage collectors, emulators and

Panic notifier and kdump

- When a game crashes the kernel, we would like to send a crash report to developers, show an error message and reboot
- This is trick to due, giving the complex work of dealing with a crashed kernel and a lot of platform code is required
- Framebuffer graphics on kdump would be desirable as well
- Work in progress by Guilherme G. Piccoli

igalia

Task scheduler

- Out of tree schedulers
- BPF "custom" schedulers
- Research in progress

igalia

HDR

- AMD GPUs have advanced color transformations features, not exposed by the Linux kernel
- Initial work for an extended DRM color API
- Now, a AMD color API
- Enable display of High Dynamic Range content in Steam Deck
- Work by Melissa Wen, Joshua Ashton and Harry Wentland

igalia

Steam Deck drivers

- You can run the Steam Deck with an upstream kernel
- All device drivers are upstream
- Fixes for existing drivers, most notably amdgpu
- Work by various developers

igalia

GPU resets

- GPUs are complex and can crash
- What should we do after a crash?
- No standard DRM API to report a reset to userspace
- No standard way of collecting and report error details
- Working on a documentation to standardize all of this
- Enhance DRM reset handling
- Work by André Almeida

igalia

Async page flips on Atomic DRM

- Legacy DRM API already supports atomic page flips
- Extending this for the Atomic API
- Work by André Almeida, Simon Ser

igalia

Conclusion

igalia

Thanks!

We are hiring: **https://www.igalia.com/jobs/**

igalia