LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

LoongArch Biweekly

Session #3 (June 11, 2026)

Discord Server

@LOONGSON_USERS

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Before We Start

- This meeting is organised by Loongson Hobbyists' Community, a 3rd-party organisation unaffiliated with Loongson Technology or any of the companies, entities, and institutions mentioned in this document.
- Please limit discussion to open information only - we will not answer questions for products and designs currently under development or testing.
- Please refrain from making political statements or commentary for the duration of the meeting.
- Opinions expressed in this meeting are strictly of the individual speakers.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

How This Meeting Works

- Most of the upstream updates comes from the Chinese biweekly, which took place last Sunday.
  - With additional updates from the past three days.
- If you would like to present, please contact for editing permissions.
  - Telegram: @JeffBai, Discord: @mingcongbai
  - Or simply shout out in the group chats!
  - You may edit until the sessions begin.
- Presentations will be in English, but Russian translation/clarification is available per request. We will have a short Q&A after each segment.
  - If you would like to help in other languages, please let us know.
- Meetings will be live streamed on YouTube…
  - … and will be recorded and made available on YouTube, VK, and Bilibili.

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Upstream Development

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

GCC

- **Xi Ruoyao (xry111)**:
  - Backported a fix for the stack smashing protector, where the stack canary value may be overridden due to unexpectedly long lifetime, to GCC 14.4 and 13.5.
    - This fix caused a regression, where the compiler would mistakenly generate 64-bit memory access instructions for LA32 - fix pending.
  - Reported a bug that GCC's main branch failed to build on LoongArch due to an ICE.
    - **Andrea Pinski** from Qualcomm reported that this issue also affects AArch64, albeit during a different stage.
    - Andrea has since fixed issue with verification from Xi Ruoyao.
    - **Xi Ruoyao** then added a compile-time check to prevent another misuse of the `EXECUTE_IF_SET_IN_HARD_REG_SET` macro.
- **Jeff Law**:
  - Fixed a test failure introduced after the ext-dce change, which caused `scan-asm` to fail on the test case `mul-const-reduction.c`.

GCC

- **Jiajie Chen**:
  - Reported that both GCC and Clang don't generate optimized code based on `vfmax.s` for the `astcenc_r` test in the SPEC CPU 2026 benchmarking suite.
  - **Xi Ruoyao** later pointed out that NaN handling of `vfmax.s` does not conform to the C standard, but the sequence of `vfcmp.slt.s` + `vbitsel.v` could be used implement same operation.
    - This optimization brought a throughput uplift from 68M op/s to 198M op/s with Clang, and from 37M op/s to 198M op/s with GCC.
    - Preliminary analysis shows that GCC was unable to auto-vectorize this operation due to the fact that multiple Phi nodes were combined into one monolithic node during the threadfull optimization pass.
- **Xinmudotmoe**:
  - Proposed that the `preserve_none` calling convention should be implemented for LoongArch.
    - It turns out that Rui Wang (heiher) has already implemented this calling convention back in January and has since been merged (will be available with a future LLVM version).

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

LLVM

- **Runze Lin (lrzlin)**:
  - Optimized the vector truncation operations based on the `[x]vpickev` instruction, along with a new SelectionDAG combine logic which uses `[x]vftintrz.w.d` instruction to truncate double-precision floating point values to signed 32-bit integers, instead of unsigned integers.
  - Optimized the code generation for LASX, changing `ANY_EXTEND` to `ZERO_EXTEND`, preventing those vectors from being unnecessarily scalarized.
  - Optimized conversion between unsigned integers and floating point values with `vftintrz.lu.d`, which converts float/double to 64-bit integers and vice versa with `vffint.d.lu`.
- **Rui Wang (heiher)**:
  - Added new SelectionDAG combination logic for widening addition/subtraction operations with the `[x]vhaddw`/`[x]vhsubw` instructions, along with a test case.

LLVM

- **Lei Wang (wangleiat)**:
  - Fixed a possible ICE where the logic handling `FP_EXTEND` operation for `v2f32` vectors may reach `ReplaceNodeResults()`, which couldn't handle such operations and will raise an internal compiler error (Issue #198339).
    - The fix extends this function with an empty `case ISD::FP_EXTEND` which immediately returns and falls back to default type legalization logic.
    - This fix also introduced a test case verifying the generated code for the widening operation from `<2 x half>` to `<2 x float>`.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Rust

- **Rui Wang (heiher)**:
  - Migrated the vector load (`vld`/`xvld`) and store (`vst`/`xvst`) implementation from inline assembly blocks to the `intrinsics::simd` interface provided by the Rust compiler.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Linux Kernel

- Platform Support
  - **Eric Briggers**: Remove the Loongson RNG driver (Revision 1)
    - Eric argues that `rng_alg` this driver utilizes has several vulnerabilities, that it served no purpose for other kernel functions, and that the effort to fix them are unwarranted.
    - Eric also suggests that a `hwrng` driver can be introduced instead.
  - **Huacai Chen**: QEMU `fw_cfg` support for LoongArch, allowing guest VMs to read configuration from the host machine (Revision 1)
  - **George Guo**:
    - Kexec Hand-Over (KHO) support for LoongArch (Revision 2)
    - `klp-build` support for LoongArch (Revision 1)
  - **Haoran Jiang**: `STRICT_MODULE_RWX` support for LoongArch to implement stricter memory access policies (Revision 1)
  - **Hongliang Wang**: `clock` attribute for Loongson 2 I2C devices (`i2c-ls2x`) (Revision 5)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Linux Kernel

- Platform Support (cont.)
  - **Binbin Zhou**:
    - Loongson I2S support (Revision 5)
    - ASoC support for 2K2000, Forever-Pi (2K0300), and DL2K0300/B (Revision 2)
- KVM Subsystem
  - **Tao Cui**: Paravirtualized KVM TLB Flush support for LoongArch (Revision 2)
  - **Qiang Ma**: Fixed emulation error for the `csrxchg` instruction in `kvm_emu_xchg()` (Revision 1)
  - **Bibo Mao**: Configure the maximum FPU feature set supported by the vCPU (Revision 3)
  - **Bibo Mao**: Fixed a bug in the user-space FPU register interface, where the FPU only receives/returns 64-bit wide data instead of full 256-bit wide data (Revision 1)
  - **Yanfei Xu**: Added out-of-bound check for `irqchip` array index (Revision 2)
- BPF Subsystem
  - **Chenguang Zhao**: Allow the `kptr xchg` operation to be inlined (Revision 1)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Linux Kernel

- Other Functional Fixes
  - **George Guo**: Fixed live update builds when `CONFIG_KFENCE` is enabled (Revision 1)
  - **Xuewen Wang**: Fixed a bug in the `set_direct_map_valid_noflush()` function where wrong value got passed to `__set_memory` (Revision 1)
- Code Cleanup & Refactor
  - **Markus Elfring**: Simplify `show_cpuinfo()` (Revision 1)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Box64

- v0.4.3-3 tagged
- Removed a workaround which forcibly uses the emulated GnuTLS functions.
- Fixed several AVX-related issues and enabled AVX/AVX2 by default.
- Fixed WoW64 handling via **`IRET`**.
- Fixed absolute path resolution in **`exec()`** for Box32.
- Fixed **`RDTSC`** emulation, which caused *Death Stranding* to run very slowly.
- Fixed error handling for **`io_uring`** file descriptors in **`mmap()`** emulation.
- Fixed the error handling of **`signal()`** function emulation.
- Added function wrappers for PipeWire.
- Fixed a bug in code caching, where it activates even with DynaRec disabled.
    - With this fixed, code caching has been enabled by default.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Box64

- Fixed unexpected removal of executable bits in code pages from native libraries.
- Added emulation for prctl operation **`PR_SET_SYSCALL_USER_DISPATCH`**.
  - This **`prctl`** operation has been used since Wine 11.5 to handle certain raw Windows syscalls.
- Added a Python interpreter wrapper for Steam's container (pressure-vessel).
- Allocated a bespoke register **`vzero`** to simplify SSE/AVX instructions.
- Fixed LBT handling of the inlined function **`UpdateFlags()`**.
- Fixed a bug where LBT's status did not update on jumps to the next code block.
- Fixed a bug with **`MOVSXD`** emulation in the Box64 interpreter.
- Added more wrappers for **`libmvec`** and **`libsqlite3`** to support Open WebUI.

Box64

- Introducing Lager - a utility to allow Box64 to run on kernels with 16KiB page size.
  - Essentially a QEMU KVM launcher, which generates and executes the startup command automatically.
  - It also acts as the `init` program (PID 1), which sets up the guest environment required to run emulated programs.
  - https://github.com/ksco/lager

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Eden Emulator

- A fork of Yuzu, the Nintendo Switch emulator.
- Two Pull Requests has been merged for LoongArch support.
  - Wired up LoongArch support for the build system.
  - Added the fundamental framework for the LoongArch dynamic recompiler, along with implementations of various simple IR operations.
- Next PR is underway…
  - Implementing more IR operations for the LoongArch64 dynamic recompiler.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Other Updates

- Bug reports and feature requests:
  - **tsubin** reported that PHPBench runs slowly on 3B6000 even when compared to Raspberry Pi 5, as previously reported by Phoronix.
  - **Xiaowei Xu (xuxiaowei-com-cn)** requested to add LoongArch support for the core plugins in the CNI (Container Network Interfaces) project.
  - **Xiaowei Xu** also requested to add LoongArch support to Kubernetes' release engineering tools.
- General contributions:
  - **Alessandro Gatti (agatti)** added LoongArch support for MicroPython, a minimal Python interpreter for microcontrollers.
  - **a6d9a6m** improved LoongArch support for the Comix kernel, adding support for automatic OSCOMP (China's National College Students' OS Development competition) test runs, and fixed a crash when booting with large amount of RAM.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Other Updates

- General Contributions (Continued):
  - **ErnstPeng** resubmitted the patches for LoongArch SIMD optimizations in x265, an H.265/HEVC video encoder - **the patches have now been merged after nearly 2 years in the pipeline.**
  - **Henrik Rydgård (hrydgard)** fixed a rendering issue with the game *Need for Speed: Carbon: Own the City*, as reported in Issue #21780.
    - The patch also fixed other LoongArch specific issues.
  - **Hanlu Li (LaurenIsACoder)** fixed a **`PT_LOAD`** segment overlapping bug in LAT (Loongson's x86 Architecture Translator) - the overlap occurs when trying to run 4KiB-paged x86_64 ELF binaries on a 16KiB-paged kernel.
    - She also cleaned up the LATX tree, removing non-LoongArch target code.
  - **Dongyan Qian (MarsDoge)** added LoongArch support for the UEFI GOP backend of the LVGL, an embedded graphics library, with a conditional compilation branch.
    - **Are we finally going to see a fancy BIOS setup interface?!**

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Other Updates

- General contributions (Continued):
  - **Anjia Wang (ouankou)** fixed a build failure and a linker selection bug for rex, a compiler suite to build source-to-source transformation and analysis tools via Fortran.
    - The build script for rex's Docker image failed to choose `ld.lld` as the linker for LoongArch, instead it chose `ld.bfd`, which resulted in a 702.4s link time when linking `librose.so`. Link time was greatly reduced to 10.7s thanks to this change.
  - **solarpasserby** finished the LoongArch port for RespOS, an OS kernel based on rCore-v3, adding LoongArch-specific linker script and a fix for entry address offset caused by `.eh_frame_hdr`.
  - **tannevaled** added support for `GOOS=tamago GOARCH=loong64` to tamago-go, a bare-metal runtime for secure and embedded applications written in Go, fully passing the testsuite.
    - The support was added on a phased approach, as corresponded with the maintainers in the Issue #70 - it therefore did not include changes related to PIE executables and external binaries.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Other Updates

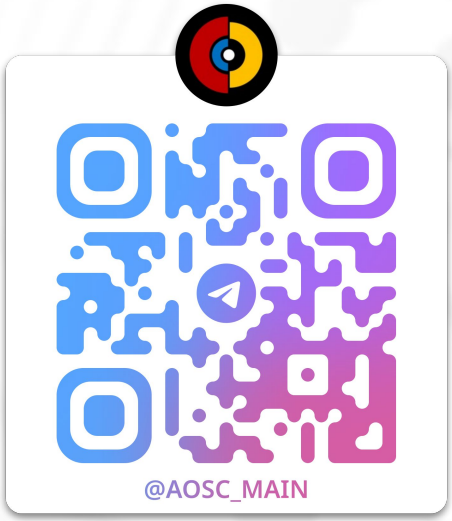
- General contributions (Continued):
  - **Xiaowei Xu (xuxiaowei-com-cn)** requested to add LoongArch support for etcd, a distributed key-value storage system, along with a detailed list of pending changes.
    - A maintainer, **Josh Berkus (jberkus)**, explained that all tests for etcd depend on Kubernetes' test infrastructure (Prow), and Prow does not yet support LoongArch.
    - **Xiaowei** later added LoongArch support to coredns, a core Kubernetes component for name resolution.
  - **Xiaowei** also added LoongArch-specific CI build and release workflow to the core plugins of CNI, Container Network Interface.

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Distro/OS Updates

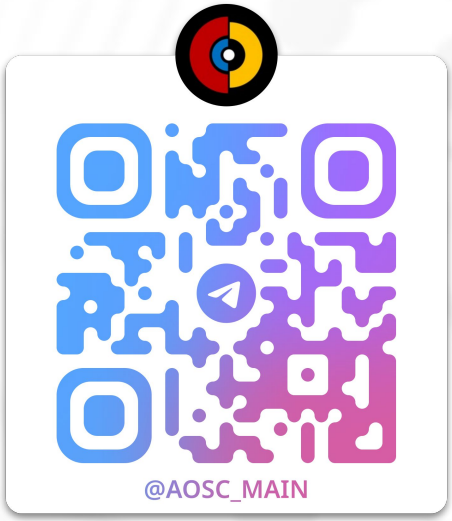
AOSC OS

- Core 13.2.1
  - Backported the fix in GCC PR121875, fixing the build failure with Node.js 24.16.0 and newer simdjson versions (including projects using this library).
- Linux 7.0.11 is now available as a stable update.
  - The update includes preliminary fixes for the mainline SMC voltage and frequency scaling driver.
  - Full voltage and frequency scaling on 6000 series CPUs and 2K3000/3B6000M will be available when paired with the public beta firmware (version 202605), coming this week or the next.



AOSC OS

- Other notable updates
  - FFmpeg has been updated to 8.1.1, with support for Vulkan-accelerated video decoding.
  - The ROCm suite has been updated to “TheRock 7.13”.
  - The ROS2 suite has been updated to the latest versions.
  - LoongArch build of PowerShell 7.6.2 is now available.
- **137** security updates in the past 2 weeks, **fixing 16 critical- and 51 high-severity vulnerabilities.**
  - This includes fixes for the HTTP/2 Bomb DoS vulnerabilities found in Apache HTTPD and nginx.
  - Server users - update your system ASAP!



LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

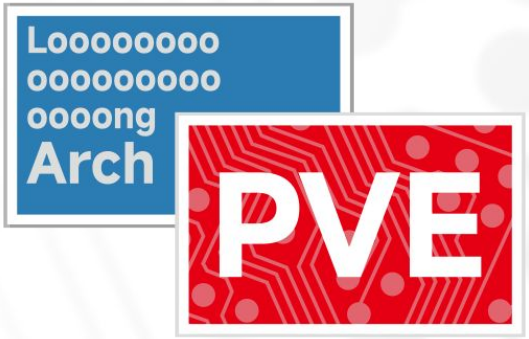
OpenWrt

- The PR submitted by Xinmu 3 weeks ago, which includes three kernel patches for OpenWrt's 6.12 kernel, has been merged and will be included in their next release.
  - This series of patches added support for the integrated Ethernet DWMAC found in 2K3000/3B6000M SoC.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Proxmox VE

- Kaiyang Wu is working on creating an installation media and found that the official Proxmox VE Installer program expects certain file structures and contents found in the official ISO image.
  - However, the upstream did not publish the tools and scripts to build the official ISO images, so we consulted the upstream.
- Proxmox VE customized kernel configuration for LoongArch and the corresponding **`proxmox-default-kernel`** package is now available.
- We continue to rebase our work on new updates from the upstream.



BaseAlt

- Ivan Melnikov working on build infrastructure and kernel build updates
- Alt Server passed QA verification. Overall state is good, several loongarch-specific bugs passed to team for investigation
- Alt Workstation passed Dev verification and ready for QA
- Continued work on Tramplin devices

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Community News & Events

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Roaming Loongson

- Two new boards are available for the project, including:
  - XB612B0_v1.2 motherboards w/ 8-core 3B6000
  - XB612B0_v1.2 motherboards w/ 16-core 3B6000
- Remote access for the boards above may be requested via:
  - https://github.com/loongson-community/1024

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Gaming on LoongArch with Box64

Speaker: ptitSeb

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Gaming on LoongArch with Box64

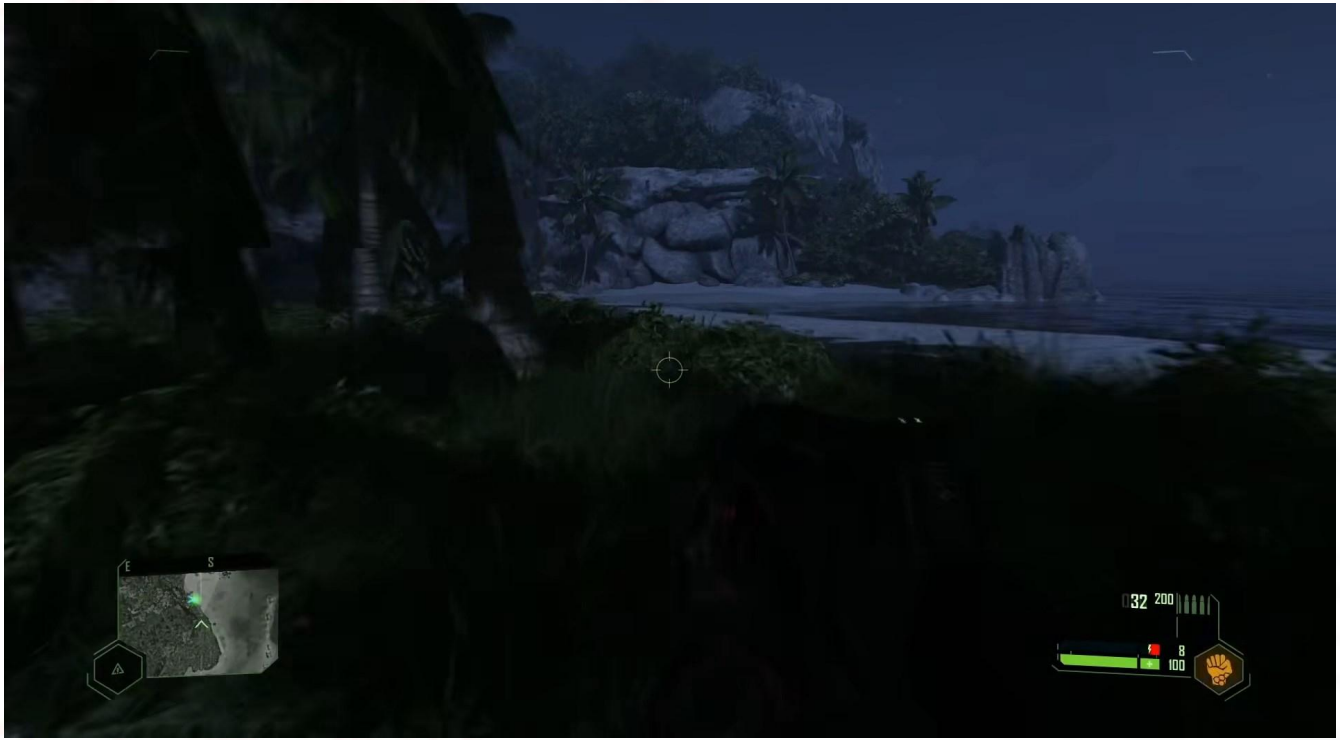
- What can you expect to run
- What do you need
- What is Box64
- How does it work

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

“ Does it run Crysis?

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Does it run Crysis?



Crysis Remaster (Steam) running on LoongArch

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Does it run Crysis?

- Many games are running on LoongArch with Box64.
- AAA Games, Indie games, …
  - Many launchers too, like Steam, Battle.net or UbiSoft Connect, just to name a few
- A partial list of game running with Box64 can be found on: https://box86.org/app/

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

“What
do you need?

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Hardware

CPU

Loongson 3A6000 minimum
Loongson 3B6000 recommended
Loongson 3C6000/S, Irtysh C616 or more: Better

GPU

AMD RX 7600 or better recommended
Other GPU might work, but prefer
AMD Chipsets for now

Memory

8 GiB minimum
16 GiB recommended

Storage

512 GiB minimum

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

OS

4KiB page-size Kernel highly recommended

Components

Sound: ALSA / PulseAudio
Video: X11 / Wayland (+Xwayland)
OpenGL 3+ highly recommended
Vulkan 1.3+ highly recommended

Launcher / Tools

Wine with WoW64
Steam (Linux version)
Heroic Launcher, etc...

Box64

Latest Box64 built with Box32
Binfmt integration

AOSC OS Recommended

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

What is Box64?

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Box86 is created as a solution to run commercial PC games on small underpowered portable machine: the Open Pandora.
- Equipped with a 32 bits Arm CPU (1 core from 600 MHz to 1GHz) with 256 MiB or 512 MiB of RAM and SD Cards as storage, the design was to limit emulation and storage footprint, and be able to develop and build on the Pandora itself:
- Emulate as little as possible
- Do not use rootfs
- Use C with as little dependencies as possible

box86

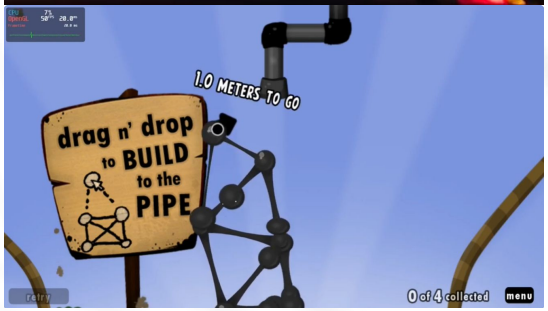
- Box64 was then developed later, based on Box86 sources, as a 64 bits program to run 64 bits (only) x86_64 programs.

- Initially as a separate product, Box32 was created to run 32 bits apps on 64 bits OS.
- To limit the amount of duplicated code to maintain, Box32 was later merged to Box64 sources, and available as a build-time option.
- Box32 as separate binary doesn't exist anymore and is now just a part of Box64.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

A quick timeline

- 2018/04: Creation of Box86 git repo (private).
- 2019/02: First version of Box86, interpreter only, capable of running Airline Tycoon Deluxe
- 2019/03: Box86 v0.0.4 and git repo now public on github.
- 2020/02: Box86 v0.1.0 published, now with ARM Dynarec.
- 2020/09: Box86 v0.1.2 with Wine support.
- 2020/11: Box86 v0.1.6 with Vulkan support.
- 2020/12: Creation of Box64 git repo (private).
- 2021/03: Box64 v0.0.2 created, still private, already running games.
- 2021/03: Box64 v0.0.6 Now with an ARM64 Dynarec.
- 2021/07: Box64 v0.1.2 and sources now public on github.
- 2022/04: Box64 v0.1.8 with Vulkan support
  2023/03: Box64 v0.2.2 and Box86 v0.3.0 with Linux Steam support
- 2023/08: Box64 v0.2.4 with Wine Wow64 support, and RISC-V Dynarec
- 2024/05: Box64 v0.2.8 with LoongArch Dynarec and some non-4K page-size support
- 2024/07: Box64 v0.3.0 with AVX+AVX2 support
- 2024/12: Box64 v0.3.2 with Box32 support for 32 bits x86 app in 64 bits OS
- 2024/12: Box86 v0.3.8 current last stable version
- 2025/03: Box64 v0.3.4 better Box32 support as Steam can run without Box86 (fully 64-bit)
- 2025/06: Box64 v0.3.6 introduced WowBox64.dll for Hangover; Volatile Metadata support
- 2025/10: Box64 v0.3.8 introduced DynaCache
- 2026/01: Box64 v0.4.0 with better Box32 support; DRM protection improvements; DynaRec improvements
- 2026/04: Box64 v0.4.2 with Fossilize & Vulkan overlay; SteamRT3 support; DynaRec improvements
- **… and next release in the making**



LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

How
does Box64 work?

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

How does Box64 work?

Box64 is composed of 3 interconnected Blocks:

- The ELF Loader
- The Librarian
- The Emulator

The ELF Loader is responsible for loading x86 & x86-64 ELF files (linux binaries).

The Librarian is responsible for handling libraries (dependencies), either emulated library or native ones.

The Emulator is composed of an Interpreter, in pure C, and DynaRec for certain architecture, that can convert x86 code to native.

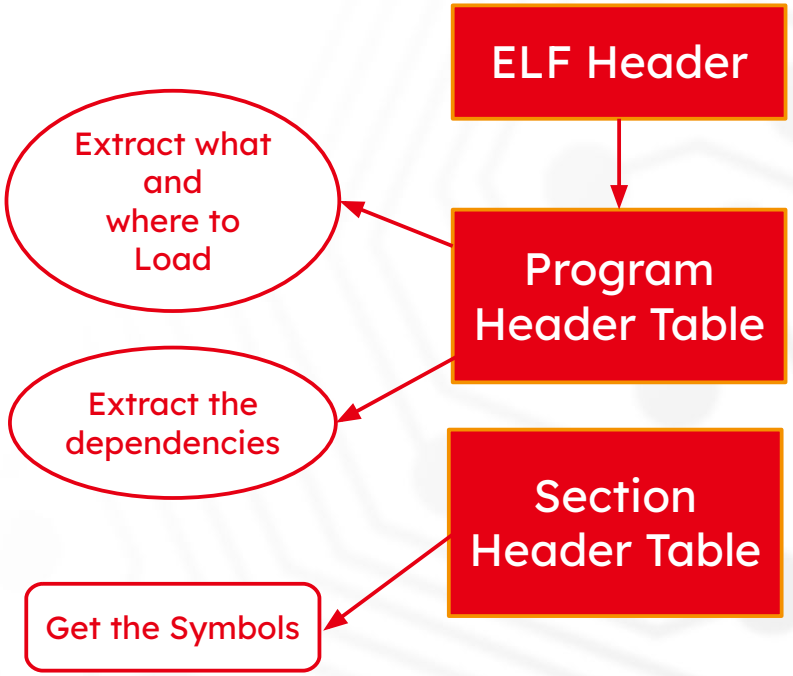
LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

The ELF Loader

When a program is launched with Box64, the **ELF Loader** will first load the main binary in memory.

It will then use the Librarian to load all needed libraries. ELF Loader is also responsible for the symbols resolution using the symbols provided by the Librarian.

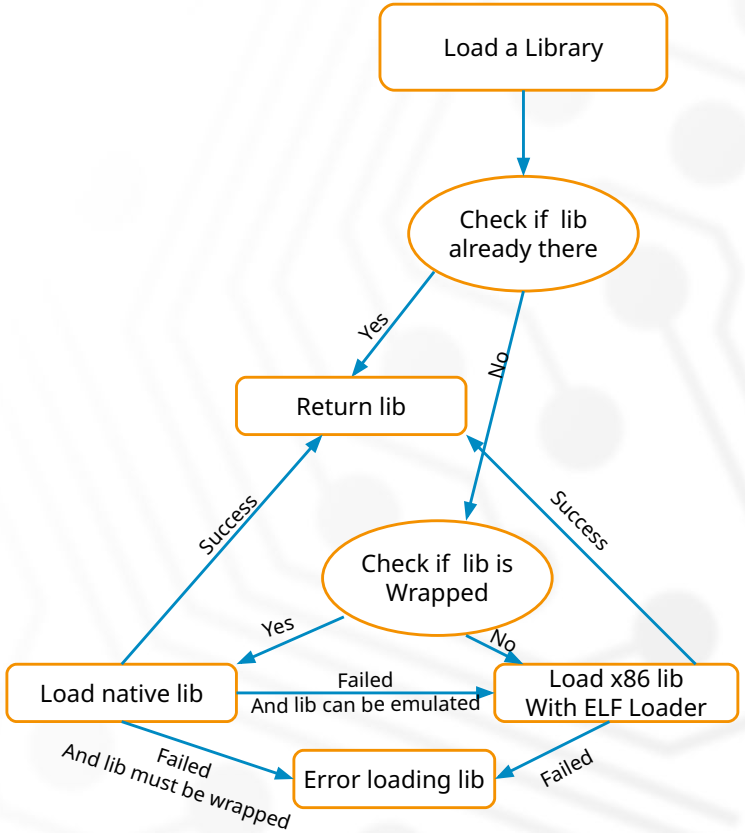
All the needed functions to load ELF files have been coded from scratch in Box64. Parsing, memory allocation, symbols resolution…



LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

The Librarian

- The Librarian will load each library one by one. The first check is to see if a libraries can be “wrapped”.
- That means the native version of the library will be used instead of an x86 one.
- Wrapped libraries is one of the unique concept of Box64. It allows the use of the native library, meaning a rootfs with x86 libs is not needed to run programs. It also avoid emulating the code from those libraries as it’s native, giving some free speed boost.
- Note that there are some libraries that needs to be wrapped, and will never be emulated, like libc for example.
- Ultimately, the Librarian is there to provides Symbols for ELF Loader’s symbol resolution.



The Emulator

- Everything that needs to be emulated (like initialization sections for emulated library or the main code from the loaded ELF) will use the **DynaRec** first if it's available.
- The DynaRec convert a block of x86 code to native code, using a simple "per x86 opcode" approach, but using global block optimisation to limit useless computations (that can lead to fusing some comparison opcodes).
- The goal of the DynaRec is to be as fast (and simple) as possible, while trying to tend to a 1:1 conversion.
- Once a block is converted, it will be re-used we needed.
- Converted blocks can be saved to disk to be re-used at next load, speeding up loading time (DynaCache).
- If an opcode is not handled by the DynaRec, execution will fallback to the **Interpreter**.

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Thank you

for your attention

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Q&A

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Next Meeting

June 25th, 2026 @ 13:00 (UTC)

Discord Server

@LOONGSON_USERS

LOONGSON HOBBYISTS COMMUNITY

Collaborate / Architect / Forward

Links

Slide 5 (Binutils):
zhaozhou - test case removal: https://sourceware.org/git/?p=binutils-gdb.git;a=commit;h=48997323b0f929d3255a4845f8a3800c6c00a53a
xry111 - target configuration: https://sourceware.org/git/?p=binutils-gdb.git;a=commit;h=f8b82c843a960020c493111385b8c5728c20eac9
mengqinggang - linker alignment fix: https://sourceware.org/pipermail/binutils/2026-June/149527.html

Slide 6 (GCC):
xry111 - SSP fix GCC 14.4 backport: https://gcc.gnu.org/cgi-bin/gcc-gitref.cgi?r=r14-12637
xry111 - main branch FTBFS report: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=125609
Jeff Law - scan-asm fix: https://gcc.gnu.org/git/?p=gcc.git;a=commit;h=232f2ca5dd7f15a7ea4320b483803b028d9c5d5f

Slide 7 (GCC):
jiegec - SPEC CPU 2026 misoptimization: https://github.com/loongson-community/discussions/issues/120
xm_moe - preserve_none proposal: https://github.com/loongson-community/discussions/issues/123

Slide 8 (LLVM):
lrzlin - vector TRUNCATE addition: https://github.com/llvm/llvm-project/pull/201548
lrzlin - new Selection DAG combine logic: https://github.com/llvm/llvm-project/pull/201569
lrzlin - ZERO_EXTEND optimization: https://github.com/llvm/llvm-project/pull/201099
lrzlin - conversion between FP and int: https://github.com/llvm/llvm-project/pull/200901
heiher - SelectionDAG combine logic for widening ADD/SUB: https://github.com/llvm/llvm-project/pull/201488
heiher - Tests for the new combine logic: https://github.com/llvm/llvm-project/pull/201487

Slide 9 (LLVM):
leiwangat - unhandled ISD::FP_EXTEND: https://github.com/llvm/llvm-project/pull/201260
leiwangat - corresponding ICE report: https://github.com/llvm/llvm-project/issues/198339

Slide 10 (Rust):
heiher - intrinsics::simd migration: https://github.com/rust-lang/stdarch/pull/2146

Slide 11 (Linux Kernel):
Eric Briggers - LOONGSON_RNG removal: https://lore.kernel.org/loongarch/20260529233208.8703-1-ebiggers@kernel.org/
Huacai Chen - QEMU fw_cfg support: https://lore.kernel.org/loongarch/20260529140559.1775511-1-chenhuacai@loongson.cn/
George Guo - kexec handover support (v2): https://lore.kernel.org/loongarch/20260529143238.169169-1-dongtai.guo@linux.dev/
George Guo - klp-build support: https://lore.kernel.org/loongarch/20260604065317.219777-1-dongtai.guo@linux.dev/
Haoran Jiang - STRICT_MODULE_RWX support: https://lore.kernel.org/loongarch/20260606132126.562034-1-haoran.jiang@linux.dev/
Hongliang Wang - clock attribute for ls2x-i2c (v5): https://lore.kernel.org/loongarch/20260604015848.18643-1-wanghongliang@loongson.cn/T/#m852923a3c647d60b6eb3b6732766d99fae668af9
Binbin Zhou - Loongson I2S ASoC driver (v5): https://lore.kernel.org/loongarch/cover.1780304703.git.zhoubinbin@loongson.cn/
Binbin Zhou - ASoC support for various boards: https://lore.kernel.org/loongarch/cover.1780538113.git.zhoubinbin@loongson.cn/

Slide 12 (Linux Kernel):
Tao Cui - KVM PV TLB Flush support (v2): https://lore.kernel.org/loongarch/20260602021819.2373404-1-cui.tao@linux.dev/
Qiang Ma - CSRXCHG fix: https://lore.kernel.org/loongarch/20260604123433.3182173-1-maqianga@uniontech.com/
Bibo Mao - FPU feature set fix (v3): https://lore.kernel.org/loongarch/20260602074316.1647373-1-maobibo@loongson.cn/
Bibo Mao - User-space FPU access fix: https://lore.kernel.org/loongarch/20260603023430.1748197-1-maobibo@loongson.cn/
Yanfei Xu - irqchip array index out-of-bounds check (v2): https://lore.kernel.org/loongarch/20260531135326.2238555-1-yanfei.xu@bytedance.com/
Chenguang Zhao - kptr xchg inlining: https://lore.kernel.org/loongarch/20260602021515.214560-1-zhaochenguang@kylinos.cn/

Links (Continued)

Slide 13 (Linux Kernel):
George Guo - build failure with kernel live updating: https://lore.kernel.org/loongarch/20260604091913.306603-1-dongtai.guo@linux.dev/
Xuewen Wang - parameter error: https://lore.kernel.org/loongarch/20260604090151.3254805-1-wangxuewen@kylinos.cn/
Markus Elfring - simplify show_cpuinfo(): https://lore.kernel.org/loongarch/c36cab4c-0a91-4802-8693-a6ae3cee65b1@web.de/

Slide 14 - 18: None

Slide 19 (Other Updates):
tsubin - low PHP performance on 3B6000: https://github.com/loongson-community/discussions/issues/122
xuxiaowei-com-cn - request for LoongArch support in CNI: https://github.com/containernetworking/plugins/issues/1263
xuxiaowei-com-cn - request for LoongArch support in Kubernets SIG-Release: https://github.com/kubernetes/sig-release/issues/3040
agatti - LoongArch support for MicroPython: https://github.com/micropython/micropython/pull/19292
a6d9a6m - automatic OSCOMP test runs for LoongArch: https://github.com/comix-kernel/comix/pull/261

Slide 20 (Other Updates):
ErnstPeng - LoongArch SIMD optimization for x265: https://github.com/Multicorewareinc/x265/pull/900
hrydgard - Various fixes for PPSSPP: https://github.com/hrydgard/ppsspp/pull/21781
hrydgard - PPSSPP Issue #21780: https://github.com/hrydgard/ppsspp/issues/21780
LaurenIsACoder - avoid PT_LOAD overlaps: https://github.com/lat-opensource/lat/pull/304
MarsDoge - Add LA64 support for the UEFI backend for LVGL: https://github.com/lvgl/lvgl/pull/10221

Slide 21 (Other Updates):
ouankou - Use lld for docker image builds: https://github.com/ouankou/rex/pull/824
(Link is unavalable for solarpasserby's RespOS port due to constant 502 from their GitLab instance)
tannevaled - add GOOS=tamgo and GOARCH=loong64 support: https://github.com/usbarmory/tamago-go/pull/17
tannevaled - Plan for introducing LoongArch support: https://github.com/usbarmory/tamago/issues/70

Slide 22 (Other Updates):
xuxiaowei-com-cn - request for LoongArch support for etcd: https://github.com/etcd-io/etcd/issues/21877
xuxiaowei-com-cn - add loongarch64 build support for coredns: https://github.com/coredns/coredns/pull/8137
xuxiaowei-com-cn - add loongarch64 support for CI testing and releasing workflows for CNI: https://github.com/containernetworking/plugins/pull/1262

Slide 24 (AOSC OS):
Xi Ruoyao - GCC PR 121875: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=121875

Slide 26 (OpenWrt):
xinmu - 3 kernel patches for OpenWrt's 6.12 kernel: https://github.com/openwrt/openwrt/pull/23366

Slide 27 (Proxmox VE):
pve-loong64-port GitHub organization: https://github.com/pve-loong64-port