LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

LoongArch Biweekly

Сессия #1 (Май 13, 2026)

Discord Server

@LOONGSON_USERS

Прежде чем мы начнем

- Эта встреча организована сообществом энтузиастов Loongson – сторонней организацией, не связанной с Loongson Technology или какими-либо другими компаниями, структурами и учреждениями, упомянутыми в этой презентации.
- Пожалуйста, ограничьтесь обсуждением только открытой информации – мы не будем отвечать на вопросы о продуктах и разработках, находящихся на стадии разработки или тестирования.
- Воздержитесь от политических заявлений и комментариев на протяжении всего собрания.
- Мнения, высказанные на этом собрании, являются исключительно личными точками зрения выступающих.

Как проводятся встречи

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Большинство обновлений поступили из китайского biweekly, который состоялся в прошлое воскресенье.
  - Дополнительные обновления за последние три дня.
- Если Вы хотите сделать презентацию, пожалуйста, обратитесь за правами на редактирование.
  - Telegram: @JeffBai, Discord: @mingcongbai
  - Или просто напишите в групповых чатах!
  - Вы можете вносить правки до начала встреч.
- Презентации будут на английском языке, но по запросу доступны перевод на русский или пояснения. После каждого блока мы будем оставлять немного времени на вопросы.
  - Если Вы хотите помочь с переводом на другие языки, пожалуйста, свяжитесь с нами..
- Встречи транслируются в прямом эфире на YouTube…
  - … а также будут записаны и доступны на YouTube, VK и Bilibili.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

“Развитие апстрима

Binutils

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Zhou Zhao исправил проблему, из-за которой GNU Assembler (gas) мог не выравнивать инструкции LoongArch, когда выравнивание не было явно указано.
  - Проблема была решена путем принудительного выравнивания по 4 байтам (выравнивание по слову).
  - Это необходимо было исправить, поскольку невыровненные инструкции вызвали бы ошибки выборки инструкций на LoongArch (AArch64 требует выравнивания по 4 байтам, а RISC-V (с расширением C) - выравнивания по 2 байтам).
  - Технически это исправление не является строго обязательным, поскольку компилятор всегда добавляет директиву .align для секции .text, и опытные разработчики также знают, что её нужно писать. Однако «Ошибка сегментации» (Segmentation Fault) может удивить новых разработчиков.
  - С другой стороны, ситуация, когда ассемблерный файл «нормально» работает с gas >= 2.44, но вызывает «Segmentation Fault» с gas <= 2.43, всё равно будет удивительной :(
- Затем Qinggang Meng добавил тестовый кейс insn_align_4.s для верификации этого исправления.

GCC (16.x)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- GCC 16.1 был выпущен, следующее должно войти в версию 16.2.
  - Тем не менее, мы рекомендуем дистрибутивам соответствующим образом бэкпортировать следующие исправления.
- Lulu Cheng:
  - Исправлен PR 125057: Внутренняя ошибка компилятора (ICE), вызванная неполными условиями разделения в loongarch_split_vector_move - GCC 16 добавил векторное перемещение __lasx_cast_128 (перемещение из 128-битного в 256-битный вектор), которое не было учтено в RTL, из-за чего оно обрабатывалось как перемещение из GPR (регистра общего назначения) в FPR (регистр с плавающей точкой).
- xry111 (Xi Ruoyao):
  - Исправлен PR 125049: Жизненный цикл истинного значения и адреса, в котором оно хранилось в стековой канарейке (stack canary) (в рамках реализации защиты стека (stack protector) в GCC), был слишком большим, что позволяло злоумышленникам читать значение, находящееся в регистре, или, при высокой нагрузке на регистры, перезаписывать адрес, вытесненный в стек, делая тем самым защиту неэффективной.
  - Добавлена возможность использовать форматирование %c для вывода имён символов во встроенном ассемблере (расширенный ассемблер верхнего уровня, представленный в GCC 16, предположительно, будет использовать эту возможность).

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

GCC (16.x, issue)

- Yang Yanjun (setarcos) сообщил, что GCC 16.1 падает (с внутренней ошибкой компилятора — ICE) при компиляции VTK 9.6.1
  - Не воспроизводится в последних релизах / ветке gcc-16.
  - Биссекция/деление пополам указала на коммит r16-8886 как на тот, который устранил эту внутреннюю ошибку.
  - Этот коммит является исправлением целевой-независимой ошибки PR 125185, поэтому данный вопрос, скорее всего, является примером PR 125185 (докладчик это пока не подтвердил).

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

GCC (17.x)

- xry111 (Xi Ruoyao):
  - Устранил лишнюю инструкцию расширения знака (slli.w …, 0) после инструкции ctz.w, которая расширяет 32-битные целочисленные CTZ..
  - Внёс операцию spaceship, которая оптимизирует выражение (a == b) ? 0 : (a < b) ? -1 : 1 как две инструкции slt[u] и одну инструкцию вычитания.
    - В своём обзоре Lulu Cheng указала, что маркер SRP_SIGNED_AND_UNSIGNED для результата может быть некорректным и требует доработки.
  - Введена оптимизация (по аналогии с RISC-V zbb), преобразующая a ^ b ^ (a | c) в (c & ~a) ^ b, чтобы задействовать инструкцию andn (AND NOT)
    - Lulu Cheng и Waterman указали, что сообщение коммита содержало ошибки и требовали исправления.

LLVM

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- lrzlin (Runze Lin):
  - Внёс пользовательское понижение (custom lowering) для знакового расширения LSX-векторов — путём комбинации инструкций vslti и vilvl/vilvh, что позволило LLVM генерировать более эффективные последовательности инструкций.
  - Добавил тестовые кейсы для операций расширения LSX/LASX для sitofp и uitofp, охватывая различные разрядности и длины векторов.
    - Проверки CI указали, что тестовый кейс использует устаревшую директиву undef, требуется доработка.
- heiher (Rui Wang):
  - Внёс поддержку векторного сложения/вычитания (ADD/SUB) для типов vNi128.LLVM в настоящее время поддерживает ADD/SUB как для v1i128 (одно 128-битное целое в 128-битном векторе), так и для v2i128 (два 128-битных целых в 256-битном векторе) — это позволит LLVM генерировать собственные векторные инструкции ADD/SUB с Q-размерными элементами: VADD.Q, VSUB.Q (LSX), и XVADD.Q, XSUB.Q (LASX).
- CSharperMantle (Rong Bao):
  - Реализовал защиту от атак типа stack clash (-fstack-clash-protection) для LoongArch (с отсылкой на реализацию в бэкенде RISC-V), используя ту же стратегию для allocation-unrolling для распределений фиксированной длины.

Rust

- heiher (Rui Wang):
  - Преимущественно внутренние изменения.
  - Проведен рефакторинг имен переносимых вспомогательных SIMD-функций и интринсиков путей, переименовав SimdL в SimdExt, а также сгруппированы пути кода как is:: (внут. интринсики), cs:: (основные SIMD-модули) и ls:: (родные вспомогательные SIMD-функции для архитектуры LoongArch).
  - Выполнен рефакторинг реализаций SIMD-битовых операций, таких как vbitclr, vbitrev, vbitset и другие — теперь все они используют crate::intrinsics::simd.

Linux Kernel

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Поддержка платформы
  - **Huacai Chen:** Проверить маркеры ACPI PCIH для каждого корневого моста PCIe, пропуск ненужного исправления ресурсов памяти (ревизия 1)
  - **Hongliang Wang:** Добавить свойства тактовой частоты и настроить частоту шины для i2c-ls2x (контроллер I2C семейства Loongson 2) (ревизия 3)
  - **Qunqin Zhao:** Проверить Security Engine на прерывания на node0 (ревизия 2)
  - **Binbin Zhou:** Шина Loongson CAN-FD (ревизия 1)
- Подсистема KVM
  - **Tao Cui:** Упростить _kvm_pte_init() с помощью memset64() (ревизия 1, отклонено из-за возможного снижения производительности)
  - **Qiang Ma:** Ограничить KVM_CAP_NR_VCPUS (рекомендуемое максимальное количество vCPUs) в зависимости от значения KVM_MAX_VCPUS (максимальное количество vCPU для платформы), для предотвращения недопустимых конфигураций (ревизия 2)
  - **Bibo Mao:** Перенести некоторые объявления переменных в paravirt.h для предотвращения предупреждений компилятора при !CONFIG_SMP, поскольку asm/qspinlock.h не включается в таких конфигурациях (ревизия 1)
  - **Bibo Mao:** Удалить ненужную interrupt injection при срабатывании программного таймера (ревизия 1)

Linux Kernel

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Прочие исправления функциональности или кода
  - **Huacai Chen:** Включить CONFIG_64BIT по умолчанию, поскольку в версии 7.1 появилась опция 32BIT, что приводит к нарушению работы существующих конфигураций (defconfig's) (ревизия 1)
  - **Huacai Chen:** Уточнить -m32/-m64 с помощью cc-option для исправления сборки в Clang (ревизия 1)
  - **Huacai Chen:** Исправить определение SYM_SIGFUNC_START в 32-разрядных сборках (ревизия 1)
  - **Wentao Guan:** Исправить проблему, из-за которой loongson_gpu_fixup_dma_hang() может вызывать исключения доступа ADE (ревизия 3)
  - **Rui Wang:** Перенести логику KASLR в заглушку EFI, устранить перекрытия сегментов памяти между initrd и boot services/kernel (которое могло привести к ошибкам загрузки) (ревизия 4)
- Прочий рефакторинг и чистка кода
  - **Qiang Ma:** Удалить неиспользуемое определение cpu_has_perf (ревизия 1)
  - **Qiang Ma:** Упростить логику show_cpuinfo() (ревизия 1, отклонено как неверное)
  - **Qiang Ma:** Исправить определение CSR_CPUID_COREID{,_WIDTH} (ревизия 1, отклонено как неверное)

Box64

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Версия 0.4.2 была выпущена, и вскоре за ней последовал оперативный релиз с исправлениями 0.4.3-1.
- Продолжена работа по улучшению инфраструктуры модульного тестирования.
- Исправлены несколько ошибок трансляции инструкций в LoongArch JIT.
  - … благодаря улучшенной инфраструктуре модульного тестирования.
- Оптимизирована производительность эмуляционной функции PCLMUL на C.
  - Архитектуре LoongArch не хватает инструкции умножения без переноса, и предыдущая программная эмуляция работала медленно.
  - Задействована реализация из OpenSSL и применён метод 4-битной таблицы поиска для повышения производительности.
- Исправлены мелкие опечатки в обёртках библиотек.
- Добавлен китайский перевод документации по использованию переменных среды.
  - Подготовительная работа для будущего графического инструмента конфигурации, для которого мы планируем добавить многоязычную поддержку.
- Исправлены несколько проблем с механизмом кэширования файлов кода.
  - Цель: Развить эту экспериментальную функцию и включить её по умолчанию перед следующим релизом.
- Изменено отложенное вычисление EFLAGS — от C-функции в JIT-встраивание.
  - Эта функция может быть горячей точкой в некоторых играх.
  - JIT-встраивание снижает издержки на переключение контекста.
  - Полностью использует расширение LoongArch LBT (Loongson Binary Translation) для дальнейшего повышения производительности.

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

UEFI (EDK II)

Прошивка UEFI от Loongson основана на EDK II, и поддержка общих платформ постепенно добавляется в основной репозиторий инженерами Loongson.

- kilaterlee (Chao Li) исправил и оптимизировал перемещение, связанное с PC, чтобы обеспечить поддержку кейсов, когда HI не примыкает к LO, а также когда один HI соответствует нескольким LO.
  - В рамках этого pull-запроса он также добавил поддержку компиляторов CLANGDWARF, что позволяет собирать прошивку UEFI с помощью Clang. Это должно дать возможность выполнять сборку UEFI с Link-Time Optimization с потенциальными преимуществами в размере бинарного файла.

Прочие обновления (feature запросы/ошибки)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Этот раздел посвящён обсуждению потенциальных запросов новых функций и, возможно, поможет вам найти следующую задачу ;-)

- bh1xaq (XiaoTan) запросил полную поддержку LoongArch в BambuStudio — программе для нарезки моделей (слайсере) для 3D-принтеров. Приложение в данный момент собирается на LoongArch, но не поддерживает бинарный сетевой плагин.
  - В связи с этим было высказано пожелание, чтобы апстрим-разработчики помогли реализовать эту поддержку.
  - Также bh1xaq отметил, что существует открытая альтернатива (open-bambu-networking).
- wojiushixiaobai (Xiaobai Wu) сообщил об ошибке в python-build-standalone: предварительно скомпилированные бинарные файлы Python «падают» с ошибкой сегментации (segfault) при запуске. Причина в том, что выравнивание ELF (по страницам 4KiB) не соответствует более распространённым страницам 16KiB.
  - **Разработчики апстрима исправили эту проблему на прошлой неделе: они обновили patchelf до версии 0.14 и установили выравнивание страниц по умолчанию на 64KiB — так же, как это делается по умолчанию в последних версиях Binutils.**

Прочие обновления (вклады)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- MarsDoge и yetist исправили проблему, из-за которой gnu-efi мог компилироваться с инструкциями LSX/LASX, которые не поддерживаются прошивкой UEFI от Loongson и вызывают ошибку утверждения (assertion error) при загрузке.
  - Проблема была решена путём указания флагов компилятора -mno-lsx и -mno-lasx.
  - **Сейчас этот патч находится на стадии проверки и тестирования.**
- phprus (Владислав Щапов) добавил поддержку LoongArch в /delta workflow, используемый в разработке библиотеки zlib-ng, который позволяет проанализировать влияние PR на размеры бинарного файла библиотеки, экспортируемые символы, ABI и степень сжатия.
  - Данный рабочий процесс уже включён; отчёты о степени сжатия, размерах бинарных файлов, размерах символов и т.д. также будут формироваться для LoongArch по запросу мейнтейнера, чтобы помочь с ревью кода.
- zevorn (Chao Liu) добавил поддержку LoongArch для machina, эмулятор полной системы с JIT, написанный на Rust. Этот эмулятор способен загружать ядра Linux (с включённым MMU) и поддерживает периферийные устройства.


ホーム
POINT
30/30 (100.0%) Battery: 1%
リズムゲーム
キャラクターセレクト
DIVAルーム
ショップ
リズムゲームをプレイすることができます。
:移動
:選択
SELECT:キャラ / カメラ位置リセット
:キャラ回転 / ズーム

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

“Обновления дистрибутивов / ОС

Советы мейнтейнерам

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Рассмотрите возможность бэкпортирования вышеупомянутого исправления для GCC PR 125049 (исправление будет бэкпортировано в версии 16.2, 15.3, 14.4 и 13.5; дистрибутивам, использующим GCC 12, придётся выполнять бэкпорт локально).
  - **Затронутые пакеты необходимо пересобрать для получения усиленной защиты.**
- Коммит в Linux 6.19 удалил *временное решение* для исключения FPU в dml21_map_dc_state_into_dml_display_cfg().
  - Окончательное исправление теперь входит в состав Linux 7.1 (в настоящее время на стадии RC), что означает, что на версиях 6.19 и 7.0 поддержка AMD RDNA 4 была нарушена.
  - Возможно, это произошло из-за путаницы, когда инженеры AMD отправляли внутренние патчи в основной репозиторий.
  - Ruoyao Xi рассматривает возможность повторного внедрения временного решения или, как альтернативу, бэкпортирования окончательного исправления, которое сейчас находится в версии 7.1.

Linux From Scratch (Linux с нуля)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Пользовательская сборка LFS 12.0 была создана для Chips & Cheese, чтобы собрать бинарные файлы тулчейна LoongArch для SPEC CPU2026 (и, возможно, для последующей передачи в SPEC для включения в следующее обновление).
  - Мы выбрали LFS 12.0, поскольку эта версия представляет собой один из самых старых базовых уровней upstream для LoongArch — glibc 2.38 и GCC 13.2, что позволяет предварительно собранным бинарным файлам работать на большинстве дистрибутивов.
  - Обратите внимание, что этот тулчейн не влияет на результаты бенчмарков — он содержит инструменты, необходимые для запуска теста на целевом устройстве.
- Добавлены пояснения о поддержке загрузчиков и прошивок с разделением по протоколу загрузки, а не просто по концепции «нового/старого миров» (new/old worlds), что позволяет избежать смешивания между реализациями ABIs и загрузки/прошивки.

AOSC OS

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Linux 6.18.27 уже продвинут, включая исправления безопасности.
  - **Dirty Frag (две уязвимости LPE) и Copy Fail (LPE).**
  - Пожалуйста, обновите систему при первой возможности (устройства под управлением AOSC OS, установленные на серверах, а также многопользовательские и общедоступные устройства следует обновить немедленно).
- Linux 7.0 теперь доступен для тестирования.
  - Содержит последние возможности LoongArch, включая оптимизацию 128-битных атомарных операций.
  - Частые проблемы: графические процессоры Intel не работают (чёрный экран с большим количеством ошибок тайм-аута операций GPU); карты AMD могут отображать графические артефакты при высокой нагрузке на процессор (?!).
  - **oma topics --opt-in linux-kernel-7.0.3**
- OpenJDK 25 теперь доступен в качестве версии OpenJDK по умолчанию.
  - Наблюдается значительный прирост производительности в Minecraft и Ghidra.
- За последние две недели выпущено **19** рекомендаций по безопасности, **15** из которых содержат уязвимости высокой или критической степени серьёзности!

@AOSC_MAIN

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Новости и события сообщества

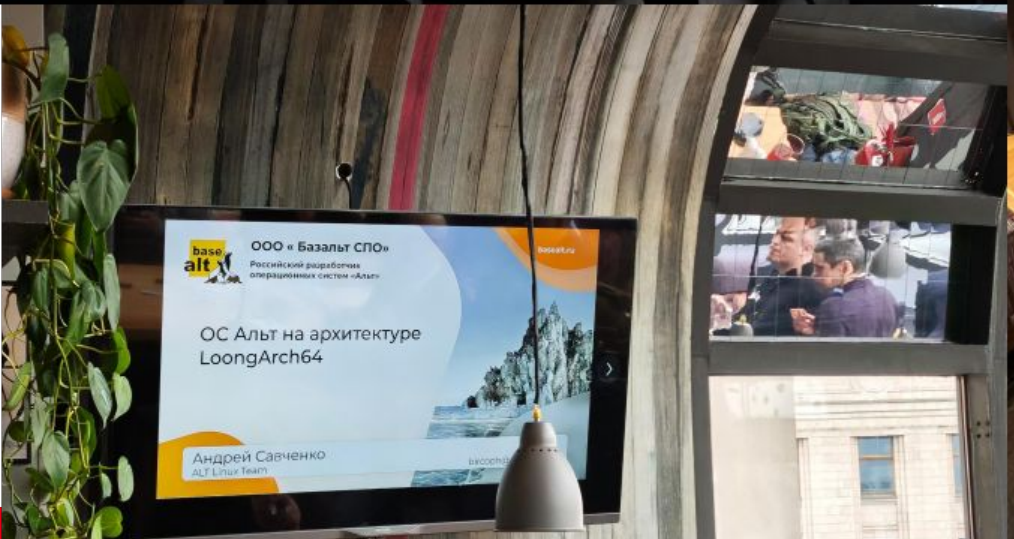
Итоги поездки в Москву

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- С 12 по 18 апреля наши сотрудники сообщества совершили поездку в Москву.
  - Мы посетили компанию Tramplin Electronics и помогали им на их стенде на выставке Expo Electronica — мы собрали игровой компьютер Irtysh C616 с Box64.
  - Мы прочитали два доклада на Irtysh Meetup ‘26 — о построении сообщества и о бинарной трансляции.
  - Много новых друзей, новой информации (и новых задач)!
- Многие наши коллеги в России проявляют большой интерес к LoongArch …
  - … но им не хватает технической поддержки, чтобы более эффективно использовать устройства на LoongArch.
  - **Нам нужно больше сессий технического обмена, подобных biweekly встречам!**
  - В ALT Linux и ROSA хорошо разбираются в LoongArch и им удалось собрать одни из первых дистрибутивов для LoongArch на основе upstream. **Но им не хватало знаний о не upstream-исправлениях и особенностях конкретных платформ.**
  - Производителям, в свою очередь, требовалась информация о поддержке периферийных устройств и обновлениях прошивок.
  - Box64 стал изюминкой, привлекая внимание энтузиастов, производителей и представителей СМИ.


Трамплин
Электроникс

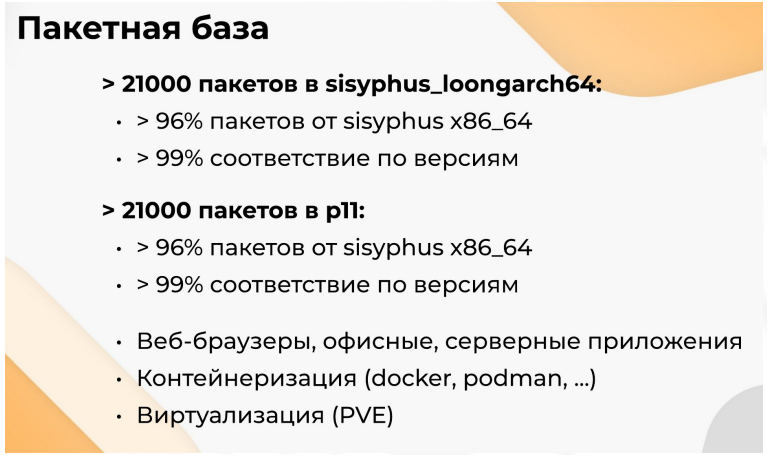

иртыш


ООО « Базальт СПО»
Российский разработчик операционных систем «Альт»
ОС Альт на архитектуре LoongArch64
Андрей Савченко
ALT Linux Team

Фидбэк от ALT Linux

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

- Работа по портированию началась в 2023 году, LoongArch показывал хорошие результаты и на сегодняшний день достиг **уровня покрытия пакетами более 95% в их репозиториях**.
- Однако устройства на LoongArch всё ещё в 2–3 раза медленнее сборочных серверов для «устоявшихся портов» (x86-64, i586 и AArch64).
  - Андрей отмечает, что ALT Linux использовал серверы 3C5000L, которые, как известно, не очень хорошо масштабируются на многоядерных нагрузках. Tramplin предоставил серверы Irtysh C632, которые должны были быть (гораздо) более производительными, но им не хватило модулей ОЗУ, чтобы ввести их в эксплуатацию (имеющиеся у них 4-ранговые модули не поддерживаются).
- LoongGPU вызывает проблемы из-за своей закрытости (закрытый исходный код) и наличия ошибок.
- Таким образом, LoongArch в настоящее время считается одним из «догоняющих портов» (catch-up ports) наряду с riscv64 и e2k (Эльбрус), фактически архитектурой второй категории (tier 2).



LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Фидбэк от других производителей

- Поддержка Mellanox NIC вызывает сомнения — было бы хорошо получить официальные заявления о совместимости и другие примечания.
- Порт VPP (Virtual Packet Processing) крайне необходим для программных маршрутизаторов и файерволов — текущий порт собирается, но ему не хватает SIMD-оптимизаций для LoongArch.
- Прошивки для материнских плат, особенно от ASUS, трудно найти.
- Компания XSquare, занимающаяся решениями для баз данных, с оптимизмом смотрит на LoongArch, но надеется на большее внимание к оптимизации посредством SIMD/ассемблера.
- Было бы полезно иметь более консолидированные каналы для получения технической информации и последних обновлений…
  мы работаем над этим!

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Небольшая реклама

Пожалуйста, посетите наш общественный портал loongfans.cn! Вскоре будут добавлены переводы на французский, немецкий и русский языки …

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

Как читать и понимать логи UEFI

…и почему это спасёт вам жизнь

(Спикер: Мария)

LOONGSON HOBBYISTS COMMUNITY
Collaborate / Architect / Forward

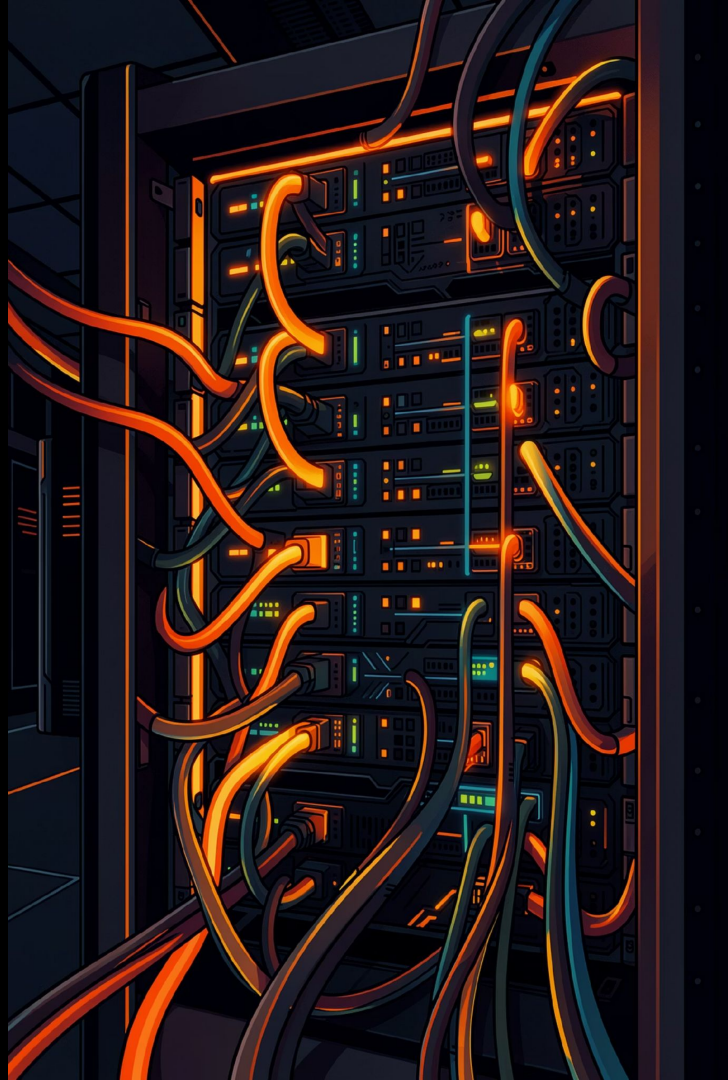
Краткое содержание

- В этой презентации разработчики встраиваемых систем и прошивок учатся считывать и понимать логи UEFI, которые выводятся через последовательный порт (UART) уже в первые микросекунды после включения питания. На примере реальной платформы Loongson (файл putty.log) мы проходим весь путь загрузки: от холодного старта и первых POST-кодов, через пробуждение ядер процессора, тренировки высокоскоростных линков, обнаружение проблемного порта PCIe и инициализацию чипсета. Главный вывод: вместо того чтобы гадать, почему плата не загружается, вы за 5 минут находите коренную причину в логе и спокойно идёте пить чай.

Как читать и понимать логи UEFI

И почему это спасёт вам жизнь

ПОЛНЫЙ РАЗБОР НА ПРИМЕРЕ ПЛАТФОРМЫ LOONGSON



О чём этот доклад и зачем он вам

Проблема

Вы — разработчик встраиваемых систем, прошивок или низкоуровневого ПО — часто сталкиваетесь с ситуацией: **«Плата не грузится».** Светодиоды мигают, вентиляторы крутятся, экран пуст. Что делать? Гадать? Менять плату? Перепаивать процессор?

Решение

Единственный надёжный способ понять, что происходит на самых ранних этапах загрузки — **это читать логи UEFI через последовательный порт (UART).**

Цель

1. Понимать структуру лога загрузки UEFI на платформе Loongson
2. Находить признаки проблем и понимать их причины
3. Связывать сообщения лога с конкретными файлами в исходном коде EDK2
4. Применять эти знания для отладки, оптимизации и разработки

Вместо того чтобы часами гадать, вы сможете за 5 минут определить по логу, что плата зависла из-за плохо вставленного модуля памяти — и пойти пить чай вместо перепайки чипсета.

Toolkit: Как получить лог

1.1 UART

На любой серьёзной встраиваемой плате есть разъём UART (универсальный асинхронный приёмопередатчик). Он работает с первых микросекунд после подачи питания — задолго до инициализации видео, USB и сети.

Что вам понадобится

- USB-UART адаптер (например, на чипе CP2102 или FT232)
- Три провода: TX, RX, GND
- Программа-терминал: PuTTY, minicom, screen
- Настройки: 115200 бод, 8 бит, 1 стоп-бит, без контроля чётности

1.2 POST-коды: язык «молчаливой» системы

До инициализации видео UEFI выводит POST-коды в порт 0x80 и дублирует их в UART. Последний POST-код точно указывает, где зависла система.

Диапазон	Фаза
0x00–0x0F	SEC Phase (очень ранняя инициализация)
0x10–0x1F	PEI Phase (до инициализации памяти)
0x20–0x3F	PEI Phase (после инициализации памяти)
0x40–0x4F	DXE Phase
0x50–0x5F	BDS Phase

Анатомия лога: SEC Phase — мир до памяти

Мы прослеживаем весь путь загрузки от первой инструкции процессора до появления меню BIOS Setup, анализируя файл putty(1).log строка за строкой.

Этап 0: Холодный старт и первые байты

```
=~= PuTTY log 2026.03.18 11:03:46 =~=
Shut down slave cores done!
```

Сообщение `Shut down slave cores done!` — это **хвост предыдущей загрузки**, завершившейся программным сбросом. Это вторая итерация загрузки.

Ключевые моменты SEC Phase

```
POSTCODE=<0x00>
CPU Type: 3C6000/D
AVS: Get Vddn value is: 0000044c
CPU CLK SEL : 00000004
```

- **Определение процессора:** Прошивка определяет модель процессора, чтобы выбрать правильный путь инициализации
- **AVS:** Адаптивное масштабирование напряжения — чтение заводских калибровочных значений. Некорректные значения вызывают нестабильность
- **CLK SEL:** Индексы множителей частоты, позже преобразуемые в реальные мегагерцы

Cache-as-RAM (CAR): критический момент

На этапе SEC основная память DRAM **ещё не инициализирована**. С-коду нужен стек — поэтому система использует **процессорный кеш как временную оперативную память**.

```
POSTCODE=<0x06>
Copy Sec code to SCache-As-Ram.
Copy PEI code to SCache-As-Ram.
POSTCODE=<0x07>
Entering C environment
```

Если настройка CAR провалится, вы **не увидите ничего**. Ни UART, ни POST-кодов. Это «чёрный экран смерти» на уровне прошивки — самая сложная для диагностики проблема.

PEI Phase: Пробуждение системы

Самый сложный и насыщенный этап: инициализируются ядра процессора, линки между сокетами и контроллер памяти, а также обнаруживается чипсет.

1

Диспетчер PEI-модулей

Находит **0x16 PEI FFS файлов** в Firmware Volume. Каждый модуль идентифицируется по GUID — например, 9B3ADA4F… = PcdPeim.efi. Используйте GUID, чтобы найти модуль в дереве исходников EDK2.

2

Пробуждение всех ядер

`PreCpuMpPei.efi` пробуждает все ядра: 4 узла × 16 ядер = 64 физических ядра. С SMT: **128 логических процессоров**. Отсутствующая строка в логе означает, что ядро не проснулось..

3

Конфигурация PHY

`PhyConfigPpi.efi` управляет электрическими параметрами высокоскоростных линков, выгружая большие таблицы параметров PHY для всех линий. Параметры могут поступать из кода, EEPROM или предыдущих тренировок.

4

Тренировка межсокетных линков

`retry to gen4 ..No.3 times link up GEN4 SUCCESSFULLY!` - Линк не смог подняться с первой попытки. Только **с третьей попытки** Gen4 (~16 GT/s) заработал. Повторы указывают на маргинальное качество сигнала или неоптимальное питание/температуру.

PCIe-сбои и спасительный мост

PCIe-катастрофа: мёртвые порты

```
init pcie 2
ltssm 0x0
error!!! not link up!
```

`ltssm = 0x0` означает состояние **Detect.Quiet** - приёмник не обнаруживает сигнала. На другом конце провода физически нет устройства.

⚠ Это **не ошибка прошивки** — это особенность платы TD622E0. Не все линии PCIe разведены к слотам. Однако каждая неудачная попытка тратит **десятки-сотни миллисекунд**, добавляя секунды к времени загрузки. Простое исправление: добавить список отключённых портов и пропускать их.

Спасительный мост: чипсет 7A2000

```
bridge init start ...
device linked in X8 mode GEN1
device linked in X8 mode GEN3
sub 7A version 0x0
```

После серии неудач линк **CPU-к-7A2000 (южный мост)** поднимается: сначала x8 Gen1, затем переходит на Gen3.

Это один из самых важных позитивных моментов в логе. Если бы этот шаг провалился, мы бы не увидели **ничего дальше** — ни SATA, ни USB, ни видео.

Детективная история: почему произошла перезагрузка?

Последовательность инициализации появляется в логе дважды. Вот полная хронология событий, объясняющая причину.

1

Чтение Flash JEDEC ID

Система считывает идентификатор микросхемы флэш-памяти при первой загрузке.

2

CMOS Cell 54 = 0x0

Подтверждён холодный запуск из G3 (Mechanical Off). `Resume from G3. Boot with Full cfg.`

3

Критично: Bit 10 is Set

`CmosRestoreFactoryInit` обнаруживает **бит 10 = 0x800** в CMOS 54 — интерпретируется как «загрузить заводские настройки по умолчанию».

4

Выполнен сброс к заводским настройкам

Все переменные UEFI сброшены. `Restore Factory Done, Start Reset!!!` Инициирован программный системный сброс.

5

Начинается вторая загрузка

Вторая половина лога — это чистая загрузка с только что сброшенными настройками. Тайна раскрыта.

ⓘ Причиной установленного бита может быть: действие пользователя в меню BIOS, несоответствие структуры переменных после обновления прошивки или **сбой питания RTC.**

Заключение и практические выводы

1

UART — главный инструмент

Лог UART — ваш основной диагностический инструмент, когда система молчит. Он работает с самого первого момента после подачи питания.

2

POST-коды

Мгновенно локализуют зависание до конкретной фазы: SEC, PEI, DXE или BDS. Последний код говорит обо всём.

3

Сбой CAR = тишина

Сбой Cache-as-RAM означает абсолютную тишину. Это самая сложная для диагностики проблема — нет ни UART, ни POST-кодов.

4

GUID и PPI

Отслеживайте любое сообщение лога непосредственно до конкретного драйвера в дереве исходников EDK2, используя GUID модуля.

5

Линки и повторы

Анализируйте сообщения «retry» и «not link up» — они указывают на аппаратные проблемы или простой потенциал для оптимизации времени загрузки.

6

CMOS и NVRAM

Могут вызывать загадочные перезагрузки. Всегда проверяйте флаги сброса, если видите повторяющуюся последовательность инициализации в логе.

> Вместо слепой перепайки вы теперь можете потратить **5 минут с логом**, найти первопричину и пойти пить чай. **Спасибо**.

LOONGSON HOBBYISTS
COMMUNITY
Collaborate / Architect / Forward

Q&A

Discord Server

@LOONGSON_USERS

LOONGSON HOBBYISTS COMMUNITY

Collaborate / Architect / Forward