Выпущен Git 2.56.0

Выпущен Git 2.56.0

Git 2.56.0 — крупный релиз с заметными изменениями в повседневных командах, partial clone, работе с refs, объектной базой, reftable, sparse index и сборкой под Windows и macOS. Также в выпуск вошло много исправлений после Git 2.55.

Главное для пользователей

- `git status` теперь даёт более полезную подсказку, если локальная ветка отстаёт от push-ветки или разошлась с ней: предлагается команда вида `git pull <remote> <branch>`.
- `git add` получил опцию `--resolved`, которая добавляет в индекс только пути с разрешёнными конфликтами и оставляет остальные локальные изменения нетронутыми. Перед добавлением Git проверяет, что в файлах не остались маркеры конфликтов.
- `git branch` получил опцию `--delete-merged` для удаления локальных веток, уже влитых в их tracked remote-tracking branches.
- `git bisect` получил опцию `--reset-when-found[=<where>]`: после нахождения проблемного коммита можно автоматически выполнить `git bisect reset` и вернуться в исходное состояние или в указанное место.
- `git replay` получил опцию `--linearize`, которая отбрасывает merge-коммиты и линеаризует переигрываемую историю, похоже на `git rebase --no-rebase-merges`.
- `git imap-send` получил опцию `--draft`, чтобы загруженные письма помечались как черновики и корректнее открывались в почтовых клиентах.
- `git rev-list` получил опцию `--missing-only`, которая показывает только отсутствующие объекты. Это удобно вместе с `--missing=print` и `--missing=print-info`.
- `git repack` получил опцию `--drop-filtered` для partial clones: она позволяет удалять локальные promisor blobs, превышающие заданный лимит, и освобождать место.

Новые и расширенные команды

- Набор инструментов `git refs` расширен подкомандами `create`, `delete`, `update` и `rename` для создания, удаления, обновления и переименования refs.
- Экспериментальная команда `git history` получила подкоманду `drop`, которая удаляет коммит и переигрывает его потомков поверх родителя.
- `git repo info` научился выводить новые ключи с абсолютными и относительными путями для `gitdir` и `commondir`.
- В `git cat-file --batch-command` добавлена команда `remote-object-info`: клиент может запросить у удалённого сервера метаданные объекта через protocol

v2, пока поддерживается размер объекта. Позже для неё добавили placeholder `%(objecttype)`.
- `git cat-file --batch-all-objects` переведён на общий интерфейс обхода object database, что упрощает код и позволяет backend-ам эффективнее использовать reachability bitmaps.

Работа с ветками, worktree и историей

- `git log --follow` лучше обрабатывает нелинейную историю, где один и тот же путь переименовывался по-разному в разных ветках истории.
- `git log --graph` теперь визуально отделяет root-коммиты и коммиты, которые стали root из-за упрощения истории. Это помогает не принять несвязанные коммиты за связанные.
- В `git blame` оптимизировано выравнивание сокращённых object names: отдельная колонка под специальные отметки резервируется только тогда, когда они действительно выводятся.
- `git worktree add` больше не пытается угадывать remote-tracking branch, если явно указаны `-b` или `-B` для создания новой ветки.
- `git checkout` и `git worktree add` теперь сообщают об ошибке, если подходящую remote-tracking branch нельзя определить однозначно.
- `git worktree repair` корректнее переписывает `.git` file рабочего дерева с относительного пути на абсолютный, если команда запущена внутри этого рабочего дерева.
- `git branch -d` теперь сообщает, если ветку нельзя удалить, потому что она используется в активном `bisect`.
- `git push origin/main` и `git branch origin main` распознаются как возможные опечатки, и Git предлагает исправление.

Конфигурация и protocol v2

- Добавлена настройка `fetch.followRemoteHEAD`, задающая значение по умолчанию для `remote.<name>.followRemoteHEAD`.
- Условные include-блоки `[includeIf "condition"]` теперь умеют учитывать расположение worktree.
- Поддержка capability `promisor-remote` обновлена: сервер может добавлять promisor remotes, если это разрешено настройкой `promisor.acceptFromServerURL`.
- Запрос `promisor-remote` без значения теперь корректно отклоняется, вместо того чтобы приводить к обращению к null pointer.
- Клиентский парсер server-advertised bundle URI теперь дочитывает оставшийся ответ при ошибочно настроенном списке, чтобы не ломать состояние протокола. Серверная сторона больше не отправляет пустые значения конфигурации как некорректные пары key-value.
- Если push remote задан URL-ом, Git теперь использует fetch refspec подходящего настроенного remote, чтобы найти и обновить remote-tracking branch, например для `@{push}`.

Индекс, sparse checkout и partial clone

- `git stash push` лучше работает со sparse index: если pathspec полностью находится внутри sparse-checkout cone, лишнее расширение sparse index не выполняется.
- Исправлено потенциальное чтение за пределами массива в проверке необходимости расширения sparse index.
- В `git mv` проверка отсутствующей ведущей директории назначения выполняется на этапе проверки, поэтому `git mv -n` теперь тоже сообщает об ошибке. Сообщение об ошибке `rename(2)` стало понятнее и показывает источник и назначение.
- Concurrent downloads packfiles через packfile URIs и dumb HTTP стали безопаснее: Git избегает одновременного append в staging file, а `fetch-pack` теперь терпим к уже существующим `.keep` files.
- Загрузка packfile URIs в protocol v2 теперь размечена Trace2 region, поэтому можно отслеживать суммарное время загрузки внешних packs и количество advertised URIs.
- Обход object database стал лучше различать отсутствующие и повреждённые объекты, возвращая более точные статусы ошибок.

Производительность

- `git cat-file --batch` на стандартном формате вывода стал быстрее за счёт более прямого формирования строк.
- `git branch` и `git tag` быстрее работают в репозиториях с большим количеством unrelated loose refs: iterator теперь сразу получает namespace prefix.
- `git branch --contains` и `git for-each-ref --contains` используют memoized commit traversal, который раньше применялся только в `git tag --contains`. Проверки связности для большого числа refs стали заметно быстрее.
- Перечисление untracked и ignored files в `git status` избавлено от квадратичной сложности при добавлении в string lists.
- Сравнение рабочего дерева с коммитом ускорено для pathspec, совпадающих с ранними index entries: Git больше не пересканирует уже распакованные записи index.
- Вычисление merge-base останавливается раньше, когда на одной стороне заканчиваются exclusive commits, что ускоряет запросы с односторонней историей.
- Добавление большого числа новых packfiles стало быстрее: для заведомо новых packfiles появился fast path, который избегает лишнего обхода.
- Reftable быстрее обновляет refs: если добавление уже держит `list_file` lock, лишний `stat` и reload stack не выполняются.
- В reftable оптимизированы чтение и обновление refs при большом количестве deletion tombstones.
- `pack-objects` теперь может совместно использовать reachability bitmaps, delta-islands и `--path-walk`, выбирая более быстрый путь упаковки.
- `pack-objects` начал записывать в Trace2 общий размер данных, записанных в pack files. Это помогает сравнивать настройки сжатия.

Документация и удобство

- Справка `git rev-parse --parseopt` и большинства подкоманд Git теперь возвращает код 0 при явном запросе `-h` или `--help`, что соответствует обычной Unix-практике.
- Usage и SYNOPSIS для `git fast-export`, `git fast-import`, `git format-patch`, `git imap-send`, `git send-email` и `git request-pull` приведены к более единообразному виду.
- Документация `git cherry-pick` уточняет, что `--no-commit` намеренно не устанавливает ref `CHERRY_PICK_HEAD`.
- Документация `git interpret-trailers` стала понятнее: описан формат ключей trailer, уточнена терминология и поведение comment lines.
- Ограничения миграции ref format в `git refs migrate` теперь показаны заметным предупреждением рядом с описанием подкоманды.
- В документацию добавлены ссылки на страницу `gitdatamodel`.
- Completion-скрипты обновлены: bash completion знает про экспериментальную `git history`, zsh completion корректнее находит команду Git после глобальных опций вроде `-C`.
- Userdiff patterns для Swift теперь поддерживают Swift-специфичные конструкции: attributes, modifiers, failable initializers и generics.

Внутренние изменения и сборка

- Продолжается избавление от глобального `the_repository`: конфигурация, refs, worktree API, pathspec, object database, tempfile и lockfile API постепенно получают явный repository context.
- Глобальные настройки вроде `core.ignorecase`, `core.filemode`, `protect_hfs`, `protect_ntfs`, `editor_program`, `pager_program`, `askpass_program` и `push_default` перенесены в per-repository структуры.
- Object database продолжает становиться pluggable: loose backend, housekeeping, fsck, packfile generation, stream APIs и transaction interfaces выделены в более самостоятельные слои.
- `git receive-pack` переведён на ODB transaction interfaces для записи входящих объектов и packfiles.
- Alternate object directories теперь загружаются при инициализации object database, а не лениво при первом lookup.
- In-memory alternate object sources удалены: submodule object databases теперь доступны через собственные repository structures.
- Sequencer освобождает object database перед запуском `git commit`, чтобы открытые file handles не мешали auto-maintenance, особенно на Windows.
- Сборка macOS universal binaries с Rust теперь поддерживается через отдельные static archives для target triples из `RUST_TARGETS` и объединение через `lipo`.
- Начиная с Git 2.56.0 сборка Windows переходит с MINGW64 на UCRT64 runtime; CMake-сборка обновлена соответственно.
- При cross-compiling с Cargo система сборки учитывает `CARGO_BUILD_TARGET`, чтобы находить артефакт в target-specific подкаталоге.

- Meson-сборка исправлена: hook-list.h генерируется до компиляции файлов, которые от него зависят.

Исправления после Git 2.55

- Исправлен возможный crash при диагностике некорректных .git files.
- Улучшена поддержка hashing loose и packed objects больше 4 ГБ на Windows и других LLP64-платформах.
- git status корректнее показывает rebase todo list и сокращает object IDs для большего числа команд.
- Парсинг GPG и SSH signatures теперь удаляет \r только перед \n, а не все carriage returns подряд.
- Reftable укреплён против повреждённых таблиц: исправлены out-of-bounds записи и чтения, а также аварийные abort при parsing.
- git merge-base исправлен для репозиториев с commit graph v1 и clock skew: ранний выход включается только при подходящем порядке queue.
- git submodule update -i снова работает как сокращение для --init.
- Rebase больше не копирует notes с dropped commit на несвязанный commit.
- git clone --revision больше не падает при работе с сервером без protocol v2.
- git diff --relative --cached больше не падает на unmerged paths вне prefix.
- git cat-file --batch-command больше не падает при запросе contents без type.
- В git rebase исправлены два случая со skipped fixup и squash: неверный счётчик коммитов в template message и неоткрывающийся editor для цепочки с fixup -c.
- Git for Windows больше не пытается автоматически определять тип symlink, если target path начинается со slash. Это защищает от утечек NTLM credentials при checkout специально подготовленных symbolic links.
- Sparse-index collapse больше не падает при invalidated cache-tree node из-за intent-to-add path.
- Trailer parser больше не принимает строки, начинающиеся с URL, например https://..., за trailer metadata.
- git worktree add исправлен для путей с trailing slashes и больше не читает за пределами строки при извлечении имени ветки или worktree.
- git history в повреждённом репозитории теперь завершает работу понятной ошибкой, если нужный tree object отсутствует.
- git checkout -m аккуратнее использует autostash fallback и отделяет совет по конфликтам пустой строкой.
- Pathspec matching больше не обращается за пределы памяти, когда negative pathspec короче общего prefix положительных pathspecs.
- Исправлены многочисленные утечки памяти и ресурсов в format-patch, merge, reftable, Bloom filters, hash backends, remote rewrite, submodule helper, Rust-интерфейсе к C hash functions и других местах.
- Исправлены проблемы с unchecked return values, invalid file descriptors и process handles, найденные статическим анализом.

Для разработчиков Git

- В проект добавлена конфигурация для b4, а документация рекомендует использовать её для отправки patches.
- `SubmittingPatches` обновлён: уточнены standard trailers, реакция на review, ограничения reroll и ожидание, что автор явно отзовёт patch series, если больше не занимается ею.
- Тесты активнее используют helper-ы вроде `test_grep`, `test_stdout_line_count` и `commit_body`, чтобы не терять exit status команд в pipelines.
- Добавлен linter `greplint`, запрещающий новые bare `grep` assertions в тестах.
- CI обновлён: статический анализ переведён на Ubuntu 24.04, Debian 11 заменён на Debian 12, pull request workflow отменяет старые runs после нового push.
- В test suite уменьшено потребление диска, часть дорогих тестов отключена на слабых платформах, а long test suite включён в GitLab CI.