Engineer

Комп'ютерний файл

іменований набір даних, що зберігається як одне ціле

Що це таке

Комп’ютерний файл — це те, що програма просить в операційної системи, коли хоче отримати ті самі байти назад завтра: вона відкриває файл за іменем, читає чи пише, закриває його й розраховує, що наступна програма, яка відкриє це ім’я, побачить те, що там залишили. Документи, фотографії, програми, налаштування й журнали — усе це файли, а в Unix-подібних системах файлами є й деякі речі, які взагалі ніде не зберігаються.

POSIX, стандарт, на якому стоять Linux, macOS та інші Unix-подібні системи, визначає файл як об’єкт, у який можна писати, з якого можна читати або і те, і інше, з атрибутами, серед яких права доступу й тип. Далі він перелічує типи: звичайні файли, каталоги, символьні посилання, FIFO, сокети, а також символьні й блокові спеціальні файли, що представляють пристрої на кшталт термінала чи диска. Кажучи «файл», люди майже завжди мають на увазі перший вид, який стандарт описує як послідовність байтів із довільним доступом, без жодної додаткової структури з боку системи. Системі байдуже, чи ці байти є електронною таблицею, фотографією чи сторінкою бази даних; структура належить програмі, яка їх читає. POSIX навіть відмовляється розрізняти текстові та двійкові файли, хоча й зауважує, що багато утиліт дають осмислений результат лише на тексті.

Ім’я зберігається поза файлом. Каталог — це список записів, і кожен запис пов’язує ім’я файлу з файлом. Ім’я файлу — це одна складова, що може містити будь-який байт, крім скісної риски й нульового байта, у межах, які встановлює система; шлях на зразок `/srv/site/index.html` — це ланцюжок імен, яким система проходить, щоб дістатися до файлу. Оскільки імена живуть у каталогах, один файл може мати їх кілька, і система веде їм лік. Коли останнє ім’я видалено й жоден процес більше не тримає файл відкритим, місце звільняється.

Усе інше, що система знає про файл, — це метадані, і довгий список показує більшість із них:

```
$ ls -li notes.txt
1048577 -rw-r----- 2 ana staff 4096 Oct 16 09:14 notes.txt
```

Перше число — серійний номер файлу, унікальний у межах його файлової системи: цей номер і є файлом, а імена лише вказують на нього. Далі йдуть біти прав для власника, групи й усіх інших, кількість імен файлу, власник і група, розмір у байтах і час останньої зміни даних. Поряд із ним POSIX зберігає ще два часи: останнього доступу до даних і останньої зміни статусу файлу. Програма, що відкрила файл, тримає дескриптор файлу — невелике ціле число, дійсне лише всередині процесу. Дескриптори 0, 1 і 2 — це стандартний ввід, вивід і потік помилок, тому оболонка може спрямувати вивід програми у файл так, що програма про це й не знатиме.

Застосуйте поняття до рішення

Застосуйте «Що таке комп'ютерний файл?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Щоденний догляд за серверами, поштою, обліковими записами й оновленнями, щоб те, на що всі покладаються, працювало тихо й без сюрпризів.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Підтримувала 10 серверів PostgreSQL і MS SQL — 10 серверів баз даних розгорнуто й підтримувано в справному стані

Підтримувала 10 серверів PostgreSQL і MS SQL Зв'язатися з нами →

Чому це важливо

Файли — найстаріший інтерфейс між програмами, які ніколи не були знайомі одна з одною. Інструмент, написаний десятиліття тому, може передати дані інструментові, написаному вчора, через файл, без жодної домовленості, крім байтів та імені. Конфігурація, журнали, експорти, резервні копії і вміст більшості вебсайтів подорожують саме так, тож тримати файли надійними — це здебільшого й означає тримати надійною систему.

Метадані — це місце, де живуть підступні проблеми. Час зміни каже, коли байти востаннє записали, а це інший факт, ніж те, коли востаннє змінилося те, що в них написано. Ці два моменти розходяться, щойно якийсь

інструмент перезаписує файл з ідентичним вмістом. Власника записано числом, тож той самий файл на іншій машині може належати іншій людині.

Далі — надійність запису. Запис, що повернув керування, передав байти операційній системі, а вона може тримати їх у кеші. POSIX каже, що дані з кешу потрапляють у сховище, коли програма просить про це викликом на кшталт `fsync`, а можуть потрапити туди й раніше самі, і попереджає, що після збою на диску можуть бути пізніші операції, тоді як раніших там немає. Як приклад він наводить перейменування, після якого нове ім'я вже створене, а старе ще залишається.

І нарешті, файл — це одна копія на одному пристрої, доки хтось не подбає про інше. Єдиний захист, що працює, хоч би як підвів носій, — це друга копія деінде, яку вже прочитали назад.

Як це виглядає на практиці

Карта сайту колись повідомляла пошуковим системам, що 172 зі 176 його сторінок змінилися в ту саму мить. Виправлення простежило дату до самого файлу: дата кожної сторінки бралася з часу зміни файлу, а експорт вмісту переписує всі файли під час кожного запуску, тож одна синхронізація проставляла ту саму дату всьому сайту. Тепер дата береться з історії версій через ланцюжок резервних варіантів: спершу явно задане поле, потім історія комітів, потім файл. Сторінка фіксує пастку в цьому ланцюжку: назву поля треба буквально вказати в переліку, інакше генератор його ніколи не прочитає. Якщо дати немає, поле лишається порожнім, а не містить здогадку, і на зафіксованому етапі реалізації повторна синхронізація після місяця правок зачепила вісім файлів зі 186.

Реалізація тестового відновлення в компанії завершується зворотним читанням файла. Її юніти systemd підтримують дамп бази даних, якщо його налаштовано, надсилання зашифрованого знімка restic із дедуплікацією через SFTP до сховища іншого провайдера, очищення за політикою зберігання та перевірку цілісності. Окремий щомісячний процес відновлює відомий файл у тимчасове місце й порівнює його байти, тож задумана перевірка стосується відновлення, а не завантаження до сховища. У записі про реалізацію від 2026-10-27 зазначено, що таймери були вимкнені за відсутності налаштованого репозиторію; жодного завершеного реального відновлення не зафіксовано. Навіть парольна фраза навмисно зберігається у файлі: юніти читають її із захищеного файла, а не зі значення змінної

середовища, яке розбирає systemd. Обробка екранування в такому значенні могла б перетворити фразу зі зворотною скісною рискою на іншу, ніж та, з якою створили репозиторій.

Роками раніше робота полягала в тому, щоб рятувати файли, часто єдину копію, з обладнання, яке вийшло з ладу: відновлення даних з SD-карток, класичних HDD, SSD, масивів RAID, зовнішніх дисків і систем Mac. Кожен із цих носіїв по-своєму ламається й по-своєму віддає дані, а файлова система Mac має свої особливості, тож знати правильний підхід до носія було не менш важливо, ніж володіти окремою технікою. Мірило було просте: файли, повернені цілими.

Де виникають проблеми

Усі сторінки водночас виглядають зміненими. Копіювання, checkout, відновлення чи експорт переписують файл і скидають його час зміни, і все далі в ланцюжку, що довіряє цьому часу, бачить зміну, якої не було. Копіювання, яке зберігає час, робить навпаки й ховає зміну, що справді відбулася. Беріть дати вмісту з самого вмісту або з історії версій, а часу файлу дайте означати те, що він означає.

Читач знаходить половину файлу. Програма, яка переписує файл на місці, лишає проміжок, у якому інший процес або наступне завантаження системи бачить частину старої версії й частину нової. Запишіть тимчасовий файл у тому самому каталозі, викличте для нього `fsync`, а тоді перейменуйте його поверх оригіналу. POSIX вимагає, щоб ім'я, яке замінюють, лишалося видимим протягом усього перейменування і вказувало або на старий файл, або на новий.

Скрипт спотикається об допустиме ім'я. Ім'я файлу може містити пробіли, переноси рядка й дефіс на початку, і скрипт, який ділить імена за пробілами або передає їх команді без лапок, зачепить не ті файли. POSIX сам радить уникати дефіса на початку імені й обмежуватися літерами, цифрами, крапкою, підкресленням і дефісом в іменах, які мають переноситися між системами. Беріть кожну змінну в лапки й завершуйте опції через `--` перед списком імен.

Секретний файл може прочитати кожен. Біти прав нового файлу беруться з налаштувань за замовчуванням процесу, який його створив, а ті зазвичай дозволяють читати групі та всім іншим. Ключ чи парольна фраза, записані

так, стають спільними для кожного облікового запису на хості. Створюйте секретні файли одразу з правами лише для власника, а не звужуйте права згодом.

Що почитати далі

- Глосарій NIST CSRC: file — визначення з публікацій NIST з безпеки, кожне з документом, звідки воно походить.
- POSIX.1-2024 Base Definitions: Definitions — формальне значення файлу, звичайного файлу, імені файлу, шляху, запису каталогу й дескриптора файлу.
- POSIX.1-2024 Base Definitions: General Concepts — права доступу, три часи файлу, переносні імена файлів і те, що збій може залишити по собі в кеші.
- POSIX.1-2024: rename() — гарантія, що ім’я, яке замінюють, увесь час вказує або на старий файл, або на новий, і саме на ній тримається безпечне збереження.



Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке комп'ютерний файл?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/file/