Engineer

Datei

benannte Daten, die als eine Einheit gespeichert werden

Was es ist

Eine Datei ist das, was ein Programm vom Betriebssystem anfordert, wenn es morgen dieselben Bytes wiederhaben möchte: Es öffnet die Datei über ihren Namen, liest oder schreibt, schließt sie und vertraut darauf, dass das nächste Programm beim Öffnen dieses Namens das Hinterlassene vorfindet. Dokumente, Fotos, Programme, Einstellungen und Logs sind Dateien. Auf Unix-ähnlichen Systemen gilt das auch für manches, das überhaupt nirgends gespeichert ist.

POSIX, der Standard hinter Linux, macOS und anderen Unix-ähnlichen Systemen, definiert eine Datei als Objekt, das gelesen oder beschrieben werden kann, gegebenenfalls auch beides. Zu seinen Attributen gehören unter anderem Zugriffsrechte und der Dateityp. Es folgen die Typen: reguläre Dateien, Verzeichnisse, symbolische Links, FIFOs, Sockets sowie zeichen- und blockorientierte Gerätedateien, die etwa für ein Terminal oder eine Festplatte stehen. Mit „Datei“ ist fast immer die erste Art gemeint. Der Standard beschreibt sie als eine Bytefolge mit wahlfreiem Zugriff, der das System keine eigene Struktur auferlegt. Das System weiß nicht, ob diese Bytes eine Tabellenkalkulation, ein Foto oder eine Datenbankseite sind, und es spielt für das System auch keine Rolle. Die Struktur gehört zum lesenden Programm. POSIX lehnt sogar eine Trennung von Text- und Binärdateien ab, merkt aber an, dass viele Dienstprogramme nur für Text sinnvolle Ausgaben liefern.

Der Name liegt außerhalb der Datei. Ein Verzeichnis ist eine Liste von Einträgen, von denen jeder einen Dateinamen mit einer Datei verknüpft. Ein Dateiname ist eine einzelne Komponente, die bis zu einer vom System gesetzten Grenze jedes Byte außer Schrägstrich und Nullbyte enthalten darf. Ein Pfad wie `/srv/site/index.html` ist die Namenskette, die das System bis zur Datei abläuft. Weil Namen in Verzeichnissen liegen, kann eine Datei mehrere haben. Das System zählt sie. Wird der letzte Name entfernt und hält kein Prozess die Datei mehr offen, wird der Speicherplatz freigegeben.

Alles andere, was das System über eine Datei weiß, sind Metadaten. Eine ausführliche Dateiliste zeigt das meiste davon:

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

Die erste Zahl ist die innerhalb ihres Dateisystems eindeutige Dateiseriennummer: Diese Nummer ist die Datei, die Namen verweisen nur darauf. Es folgen die Berechtigungsbits für Eigentümer, Gruppe und alle anderen, die Anzahl der Namen der Datei, Eigentümer und Gruppe, die Größe in Bytes und der Zeitpunkt der letzten Datenänderung. POSIX führt daneben zwei weitere Zeitpunkte: den letzten Datenzugriff und die letzte Änderung des Dateistatus. Ein Programm, das eine Datei geöffnet hat, hält einen Dateideskriptor, eine kleine prozessinterne Ganzzahl. Die Deskriptoren 0, 1 und 2 stehen für Standardeingabe, Standardausgabe und Standardfehlerausgabe. Deshalb kann eine Shell die Ausgabe eines Programms in eine Datei leiten, ohne dass das Programm davon weiß.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine Datei?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Server, E-Mail, Konten und Updates werden im Alltag betreut, damit die Dinge, auf die alle angewiesen sind, unauffällig funktionieren.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 10 PostgreSQL-/MS-SQL-Server auf Ubuntu — 10 bereitgestellte und funktionsfähig gehaltene Datenbankserver

10 PostgreSQL-/MS-SQL-Server auf Ubuntu Kontakt aufnehmen →

Warum es wichtig ist

Dateien sind die älteste Schnittstelle zwischen Programmen, die einander nie vorgestellt wurden. Ein vor Jahrzehnten geschriebenes Werkzeug kann einem von gestern Daten über eine Datei übergeben, ohne weitere Vereinbarung als Bytes und Name. Konfigurationen, Logs, Exporte, Sicherungen und die Inhalte der meisten Websites nehmen diesen Weg. Verlässliche Dateien sind deshalb ein Großteil eines verlässlichen Systems.

Die subtilen Probleme stecken in den Metadaten. Der Änderungszeitpunkt sagt, wann die Bytes zuletzt geschrieben wurden. Das ist etwas anderes als der Zeitpunkt der letzten inhaltlichen Änderung. Beide fallen auseinander, sobald ein Werkzeug eine Datei mit identischem Inhalt neu schreibt. Eigentümerschaft wird als Zahl gespeichert. Dieselbe Datei kann daher auf einem anderen Rechner einer anderen Person gehören.

Dann ist da die Dauerhaftigkeit. Kehrt ein Schreibaufruf zurück, wurden die Bytes dem Betriebssystem übergeben, das sie in einem Cache halten kann. Laut POSIX erreichen zwischengespeicherte Daten das Speichermedium, wenn ein Programm es mit einem Aufruf wie `fsync` verlangt. Sie können auch vorher von selbst dorthin gelangen. Der Standard warnt, dass nach einem Absturz spätere Operationen auf der Platte liegen können, während frühere fehlen. Als Beispiel nennt er eine Umbenennung, bei der der neue Name angelegt wurde und der alte noch besteht.

Schließlich ist eine Datei eine Kopie auf einem Gerät, bis jemand etwas anderes organisiert. Speichermedien fallen je nach Art unterschiedlich aus. Der einzige übergreifend wirksame Schutz ist eine zweite Kopie an einem anderen Ort, die auch wieder gelesen wurde.

Wie es in der Praxis aussieht

Der Änderungszeitpunkt einer Datei hält fest, wann ihre Bytes zuletzt geschrieben wurden, nicht wann sich ihr Inhalt zuletzt änderte. Die Sitemap dieser Website zeigte den Unterschied einst deutlich: Ihre Datumsangaben ordneten 172 der 176 Seiten denselben Änderungszeitpunkt zu. Die Korrektur verfolgte das Datum bis zur Datei selbst zurück. Der Inhaltsexport schreibt bei jedem Lauf jede Datei neu. Unabhängig vom Seiteninhalt gab eine Synchronisierung der gesamten Website daher einen einzigen Zeitstempel. Das Datum stammt nun über eine Rückfallkette aus der Versionshistorie: zuerst ein ausdrückliches Feld, dann die Commit-Historie, dann die Datei. Die Seite dokumentiert die Falle in dieser Kette: Der wörtliche Feldname muss in der Liste stehen, sonst liest der Generator ihn nie. Ein fehlendes Datum lässt das Feld leer, statt eine Vermutung einzusetzen. Im dokumentierten Implementierungsstand betraf eine erneute Synchronisierung nach einem Monat Bearbeitung acht von 186 Dateien.

Die Implementierung des Wiederherstellungstests im Unternehmen endet mit dem erneuten Lesen einer Datei. Ihre systemd-Units unterstützen einen Datenbankdump, wo er konfiguriert ist, einen verschlüsselten und deduplizierten restic-Snapshot, der per SFTP an Speicher bei einem anderen Anbieter gesendet wird, das Bereinigen nach Aufbewahrungsregeln und eine Integritätsprüfung. Ein separater monatlicher Ablauf stellt eine bekannte Datei an einem temporären Ort wieder her und vergleicht ihre Bytes. Der beabsichtigte Test ist somit eine Wiederherstellung statt eines Uploads. Der Implementierungsnachweis vom 2026-10-27 hält fest, dass die Timer mangels konfigurierten Repositorys deaktiviert waren, und dokumentiert keine abgeschlossene Live-Wiederherstellung. Selbst die Passphrase ist absichtlich eine Datei: Die Units lesen sie aus einer geschützten Datei statt aus einem von systemd ausgewerteten Umgebungswert, dessen Escape-Verarbeitung eine Passphrase mit Backslash so verändern könnte, dass sie von der bei der Repository-Erstellung verwendeten abweicht.

Jahre zuvor ging es darum, Dateien aus defekter Hardware zu retten, oft die einzige Kopie: Datenrettung von SD-Karten, rotierenden Festplatten, SSDs, RAID-Verbünden, externen Laufwerken und Mac-Systemen. Jedes dieser Medien fällt anders aus und gibt seine Daten anders preis. Ein Mac-Dateisystem hat zudem eigene Besonderheiten. Die Arbeit bestand deshalb ebenso sehr im passenden Vorgehen für das Medium wie in einer einzelnen Technik. Der Maßstab war schlicht: unversehrt zurückgegebene Dateien.

Wo es schiefgeht

Alle Seiten scheinen sich gleichzeitig geändert zu haben. Kopieren, Auschecken, Wiederherstellen oder Exportieren schreibt die Datei neu und setzt ihren Änderungszeitpunkt neu. Jedes nachgelagerte System, das diesem vertraut, sieht eine Änderung, die nie stattfand. Eine Kopie mit erhaltenen Zeitstempeln tut das Gegenteil und verbirgt eine tatsächliche Änderung. Entnehmen Sie Datumsangaben zum Inhalt dem Inhalt selbst oder der Versionshistorie und lassen Sie den Dateizeitpunkt das bedeuten, was er bedeutet.

Ein Leser findet eine halbe Datei. Schreibt ein Programm eine bestehende Datei direkt neu, gibt es ein Zeitfenster, in dem ein anderer Prozess oder der nächste Systemstart Teile der alten und der neuen Fassung sieht. Schreiben Sie eine temporäre Datei im selben Verzeichnis, rufen Sie dafür `fsync` auf und

benennen Sie sie dann so um, dass sie die Originaldatei ersetzt. POSIX verlangt, dass der ersetzte Name während der gesamten Umbenennung sichtbar bleibt und entweder auf die alte oder auf die neue Datei verweist.

Ein Skript stolpert über einen zulässigen Namen. Ein Dateiname darf Leerzeichen, Zeilenumbrüche und einen führenden Bindestrich enthalten. Ein Skript, das Namen an Leerraum zerlegt oder ohne Anführungszeichen an einen Befehl übergibt, bearbeitet die falschen Dateien. POSIX empfiehlt selbst, einen führenden Bindestrich zu vermeiden und für übertragbare Namen bei Buchstaben, Ziffern, Punkt, Unterstrich und Bindestrich zu bleiben. Setzen Sie jede Expansion in Anführungszeichen und beenden Sie die Optionsauswertung vor einer Namensliste mit `--`.

Eine geheime Datei ist für alle lesbar. Die Berechtigungsbits einer neuen Datei stammen aus den Voreinstellungen des erzeugenden Prozesses. Diese erlauben üblicherweise der Gruppe und anderen das Lesen. Ein so geschriebener Schlüssel oder eine Passphrase wird mit jedem Konto auf dem Host geteilt. Erstellen Sie geheime Dateien von Anfang an mit Zugriffsrechten nur für den Eigentümer, statt die Rechte nachträglich einzuschränken.

Weiterführende Lektüre

- NIST-CSRC-Glossar: Datei — die Definitionen aus NIST-Sicherheitsveröffentlichungen, jeweils mit ihrem Ursprungsdokument.
- POSIX.1-2024 Basisdefinitionen: Definitionen — die formale Bedeutung von Datei, regulärer Datei, Dateiname, Pfadname, Verzeichniseintrag und Dateideskriptor.
- POSIX.1-2024 Basisdefinitionen: Allgemeine Konzepte — Berechtigungen, die drei Dateizeitpunkte, übertragbare Dateinamen und das, was ein Absturz im Cache hinterlassen kann.
- POSIX.1-2024: rename() — die Garantie, dass ein ersetzter Name durchgehend entweder auf die alte oder auf die neue Datei verweist, auf der sicheres Speichern beruht.



Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist eine Datei?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/file/