Engineer

Fichier informatique

ensemble de données nommé et stocké comme une seule unité

Ce que c’est

Un fichier informatique est ce qu’un programme demande au système d’exploitation lorsqu’il veut retrouver les mêmes octets le lendemain : il ouvre le fichier par son nom, lit ou écrit, le ferme et compte sur le fait que le prochain programme qui ouvrira ce nom y trouvera ce qui a été laissé. Documents, photos, programmes, réglages et journaux sont tous des fichiers. Sur les systèmes de type Unix, plusieurs choses qui ne sont stockées nulle part en sont également.

POSIX, la norme à la base de Linux, macOS et des autres systèmes de type Unix, définit un fichier comme un objet permettant l’écriture, la lecture ou les deux, avec des attributs comprenant ses droits d’accès et son type. Elle énumère ensuite les types : fichiers ordinaires, répertoires, liens symboliques, FIFO, sockets, et fichiers spéciaux de caractères ou de blocs, qui représentent des périphériques tels qu’un terminal ou un disque. Quand on dit fichier, on désigne presque toujours le premier type, que la norme décrit comme une séquence d’octets à accès direct à laquelle le système n’impose aucune structure. Le système ne sait pas, et n’a pas à savoir, si ces octets sont une feuille de calcul, une photo ou une page de base de données ; la structure appartient au programme qui les lit. POSIX ne distingue même pas les fichiers texte des fichiers binaires, tout en notant que beaucoup d’utilitaires ne donnent de résultat pertinent que sur du texte.

Le nom est conservé à l’extérieur du fichier. Un répertoire est une liste d’entrées, chacune associant un nom de fichier à un fichier. Un nom de fichier est un composant unique qui peut contenir n’importe quel octet sauf une barre oblique et l’octet nul, dans une limite fixée par le système. Un chemin tel que `/srv/site/index.html` est la chaîne de noms que le système parcourt pour l’atteindre. Puisque les noms vivent dans les répertoires, un fichier peut en avoir plusieurs, et le système en tient le compte. Lorsque le dernier nom est supprimé et qu’aucun processus ne garde le fichier ouvert, l’espace est libéré.

Tout ce que le système sait d'autre sur un fichier relève des métadonnées, dont un affichage détaillé montre la plupart :

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

Le premier nombre est le numéro de série du fichier, unique au sein de son système de fichiers : ce numéro représente le fichier, les noms ne font que pointer vers lui. Viennent ensuite les bits de permission pour le propriétaire, le groupe et tous les autres, le nombre de noms du fichier, le propriétaire et le groupe, la taille en octets et l'heure de la dernière modification des données. POSIX conserve deux autres dates à côté : le dernier accès aux données et le dernier changement de l'état du fichier. Un programme qui a ouvert un fichier détient un descripteur de fichier, un petit entier propre au processus. Les descripteurs 0, 1 et 2 correspondent respectivement à l'entrée standard, à la sortie standard et à la sortie d'erreur standard ; c'est pourquoi un shell peut envoyer la sortie d'un programme dans un fichier sans que le programme le sache.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un fichier informatique ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. L'entretien quotidien des serveurs, de la messagerie, des comptes et des mises à jour, pour que les outils indispensables restent sans histoire.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 10 serveurs PostgreSQL et MS SQL sur Ubuntu — 10 serveurs de bases de données déployés et maintenus

10 serveurs PostgreSQL et MS SQL sur Ubuntu Contactez-nous →

Pourquoi c'est important

Les fichiers sont la plus ancienne interface entre des programmes qui n'ont jamais été présentés l'un à l'autre. Un outil écrit il y a des décennies peut transmettre des données à un outil écrit hier par l'intermédiaire d'un fichier, sans autre accord que les octets et le nom. Configuration, journaux, exports, sauvegardes et contenu de la plupart des sites web passent ainsi ; préserver la fiabilité d'un système consiste donc en grande partie à préserver celle de ses fichiers.

Les problèmes subtils se nichent dans les métadonnées. La date de modification indique quand les octets ont été écrits pour la dernière fois, ce qui est différent du dernier changement de ce qu'ils expriment. Les deux divergent dès qu'un outil réécrit un fichier au contenu identique. Le propriétaire est enregistré sous forme de numéro ; le même fichier peut donc appartenir à une personne différente sur une autre machine.

Vient ensuite la durabilité. Une écriture qui rend la main a transmis les octets au système d'exploitation, lequel peut les garder en cache. POSIX indique que les données en cache atteignent le stockage lorsqu'un programme le demande par un appel tel que `fsync`, et qu'elles peuvent y arriver spontanément plus tôt. La norme avertit qu'après un plantage, des opérations ultérieures peuvent être sur disque alors que des opérations antérieures manquent. Elle donne l'exemple d'un renommage qui laisse le nouveau nom créé et l'ancien encore en place.

Enfin, un fichier reste une copie sur un périphérique tant que personne ne prévoit autre chose. Les défaillances du stockage dépendent du support ; la seule protection valable pour toutes est une deuxième copie ailleurs, qui a été relue.

À quoi cela ressemble en pratique

La date de modification d'un fichier consigne la dernière écriture de ses octets, pas le dernier changement de son contenu. Le sitemap du site a montré cet écart : ses dates plaçaient 172 des 176 pages à un même instant de modification. La correction a retracé la date jusqu'au fichier lui-même : l'export de contenu réécrit tous les fichiers à chaque exécution ; quel que soit ce que disait une page, une synchronisation attribuait donc un horodatage unique à tout le site. La date vient maintenant de l'historique des versions, par une chaîne de repli qui essaie un champ explicite, puis l'historique des commits, puis le fichier. L'entrée consigne le piège de cette chaîne : le nom littéral du champ doit figurer dans la liste, sinon le générateur ne le lit jamais. Une date manquante laisse le champ vide au lieu de fournir une supposition, et, au stade d'implémentation documenté, une resynchronisation après un mois de modifications ne touchait que huit fichiers sur 186.

L'implémentation du test de restauration de l'entreprise se termine par la relecture d'un fichier. Ses unités systemd prennent en charge un export de la base là où il est configuré, un instantané restic chiffré et dédupliqué envoyé par SFTP vers un espace de stockage chez un autre fournisseur, la purge selon la politique de conservation et une vérification d'intégrité. Un workflow mensuel

distinct restaure un fichier connu dans un emplacement temporaire et compare ses octets : le test prévu est donc une restauration, pas un envoi. Le compte rendu d'implémentation daté du 2026-10-27 indique que les minuteurs étaient désactivés faute de dépôt configuré et ne consigne aucune restauration réelle achevée. Même la phrase secrète est volontairement conservée dans un fichier : les unités la lisent dans un fichier protégé plutôt que dans une valeur d'environnement analysée par systemd, dont le traitement des échappements pourrait, en présence d'une barre oblique inverse, produire une phrase secrète différente de celle utilisée pour créer le dépôt.

Des années auparavant, le travail consistait à sauver des fichiers dont il ne restait souvent qu'une copie, à partir de matériel défaillant : de la récupération de données sur cartes SD, disques durs mécaniques, SSD, ensembles RAID, disques externes et systèmes Mac. Chacun tombe en panne et livre ses données différemment, et un système de fichiers Mac a ses particularités. Le travail reposait donc autant sur l'approche adaptée au support que sur une technique particulière. Le résultat se mesurait simplement : des fichiers rendus intacts.

Où les problèmes surviennent

Toutes les pages semblent changer à la fois. Une copie, une extraction depuis le gestionnaire de versions, une restauration ou un export réécrit le fichier et réinitialise sa date de modification. Chaque système en aval qui se fie à cette date voit alors un changement qui n'a jamais eu lieu. Une copie qui conserve les dates fait l'inverse et masque un changement réel. Prenez les dates du contenu dans le contenu lui-même ou dans l'historique des versions, et laissez la date du fichier signifier ce qu'elle signifie.

Un lecteur trouve un demi-fichier. Un programme qui réécrit un fichier sur place laisse un intervalle pendant lequel un autre processus, ou le prochain démarrage, voit une partie de l'ancienne version et une partie de la nouvelle. Écrivez un fichier temporaire dans le même répertoire, appelez `fsync` sur lui, puis renommez-le à la place de l'original. POSIX exige que le nom remplacé reste visible pendant tout le renommage et pointe soit vers l'ancien fichier, soit vers le nouveau.

Un script trébuche sur un nom valide. Un nom de fichier peut contenir des espaces, des retours à la ligne et un tiret initial. Un script qui découpe les noms sur les caractères d'espacement ou les passe à une commande sans guillemets agit sur les mauvais fichiers. POSIX conseille d'éviter le tiret initial et de s'en tenir

aux lettres, chiffres, point, trait de soulignement et tiret pour les noms qui doivent être portables. Mettez chaque expansion entre guillemets et terminez l'analyse des options par `--` avant une liste de noms.

Tout le monde peut lire un fichier secret. Les bits de permission d'un nouveau fichier viennent des réglages prédéfinis du processus qui le crée, et ceux-ci laissent généralement le groupe et les autres lire. Une clé ou une phrase secrète écrite ainsi est partagée avec tous les comptes de l'hôte. Créez les fichiers secrets avec des droits réservés au propriétaire dès le départ, au lieu de les restreindre ensuite.

Pour aller plus loin

- Glossaire NIST CSRC : fichier — les définitions des publications de sécurité du NIST, chacune avec son document d'origine.
- Définitions de base POSIX.1-2024 : définitions — le sens formel de fichier, fichier ordinaire, nom de fichier, chemin, entrée de répertoire et descripteur de fichier.
- Définitions de base POSIX.1-2024 : concepts généraux — les droits, les trois dates du fichier, la portabilité des noms et ce qu'un plantage peut laisser dans le cache.
- POSIX.1-2024 : rename() — la garantie qu'un nom remplacé pointe à tout moment vers l'ancien fichier ou le nouveau, sur laquelle repose l'enregistrement sûr.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'un fichier informatique ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/file/