Engineer

# Fil

et navngitt sett med data lagret som én enhet

## Hva det er

Et program som vil finne de samme bytene igjen i morgen, ber operativsystemet om en fil: det åpner filen ved navn, leser eller skriver, lukker den og stoler på at det neste programmet som åpner det navnet, ser det som ble lagt igjen der. På engelsk heter den file eller computer file, og på norsk sier mange også datafil. Dokumenter, bilder, programmer, innstillinger og logger er filer, og på Unix-lignende systemer gjelder det også flere ting som ikke er lagret noe sted i det hele tatt.

POSIX, standarden bak Linux, macOS og de andre Unix-lignende systemene, definerer en fil som et objekt som kan skrives til, leses fra eller begge deler, med attributter som bl.a. omfatter tilgangsrettighetene og typen. Deretter lister standarden opp typene: vanlige filer (regular files), kataloger, symbolske lenker, FIFO-er, sockets og tegn- og blokkspesialfiler, som står for enheter som en terminal eller en disk. Når folk sier fil, mener de nesten alltid den første typen, som standarden beskriver som en rekke byte med direkte tilgang (random access), der systemet ikke legger noen egen struktur på innholdet. Systemet vet ikke og bryr seg ikke om bytene er et regneark, et bilde eller en side i en database; strukturen hører til programmet som leser dem. POSIX vil ikke engang skille tekstfiler fra binærfiler, men noterer at mange verktøy bare gir meningsfull utdata når de får tekst.

Navnet ligger utenfor filen. En katalog, i dagligtale ofte en mappe, er en liste over oppføringer, og hver oppføring knytter et filnavn til en fil. Et filnavn er én enkelt komponent som kan inneholde hvilken som helst byte unntatt skråstrek og nullbyten, opp til en grense systemet setter; et stinavn som `/srv/site/index.html` er kjeden av navn systemet går gjennom for å komme fram. Siden navnene bor i kataloger, kan én fil ha flere av dem, og systemet holder tellingen. Når det siste navnet fjernes og ingen prosess lenger har filen åpen, blir plassen frigjort.

Alt annet systemet vet om en fil, er metadata, og en lang listing viser det meste av det:

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

Det første tallet er filens serienummer (file serial number), ofte kalt inodenummeret, og det er unikt innenfor filsystemet: det tallet er filen, og navnene peker bare på det. Så kommer tilgangsbitene for eieren, gruppen og alle andre, antallet navn filen har, eier og gruppe, størrelsen i byte og tidspunktet for siste endring av dataene. POSIX holder rede på to tidspunkter til ved siden av det: siste tilgang til dataene og siste endring av filens status. Et program som har åpnet en fil, holder en fildeskriptor, et lite heltall som bare gjelder inne i prosessen. Deskriptor 0, 1 og 2 er standard inndata, utdata og feilutdata, og derfor kan et skall sende utdataene fra et program inn i en fil uten at programmet vet om det.

Bruk begrepet i en beslutning

Bruk «Hva er en fil?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Daglig ettersyn av servere, e-post, kontoer og oppdateringer, slik at det alle er avhengige av, forblir kjedelig.

**Se tjenestene våre** Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu — 10 databaseservere produksjonssatt og holdt i god stand

**Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu** Ta kontakt →

## Hvorfor det er viktig

Filer er det eldste grensesnittet mellom programmer som aldri er blitt presentert for hverandre. Et verktøy skrevet for flere tiår siden kan gi data videre til et som ble skrevet i går, gjennom en fil, uten noen annen avtale enn bytene og navnet. Konfigurasjon, logger, eksporter, sikkerhetskopier og innholdet på de fleste nettsteder reiser alle på denne måten, så å holde filene pålitelige er det meste av å holde et system pålitelig.

Metadataene er der det subtile trøbbelet bor. Endringstidspunktet (mtime) sier når bytene sist ble skrevet, og det er et annet faktum enn når det de sier, sist endret seg. De to skilles i det øyeblikket et verktøy skriver om en fil med nøyaktig

samme innhold. Eierskap lagres som et tall, så den samme filen kan tilhøre en annen person på en annen maskin.

Så er det holdbarheten. Et skrivekall som returnerer, har bare gitt bytene til operativsystemet, og det kan holde dem i minnet en stund. POSIX sier at data som venter i minnet, når fram til lagringsmediet når et program ber om det med et kall som `fsync`, og at de kan komme dit tidligere av seg selv. Standarden advarer også om at etter et krasj kan senere operasjoner ligge på disken mens tidligere mangler, og eksempelet den gir, er en omdøping som etterlater det nye navnet opprettet og det gamle fortsatt på plass.

Til slutt er en fil én kopi på én enhet, helt til noen sørger for noe annet. Lagringsmedier svikter på måter som avhenger av mediet, og den eneste beskyttelsen som holder på tvers av alle, er en kopi nummer to et annet sted, som noen faktisk har lest tilbake.

## Slik ser det ut i praksis

Avstanden mellom de to tidspunktene viste seg en gang i nettstedets sitemap: datoene der plasserte 172 av de 176 sidene i ett og samme endringsøyeblikk. Rettelsen fulgte datoen tilbake til selve filen: hver fil får et nytt endringstidspunkt hver gang innholdseksporten kjører, fordi eksporten skriver alle filene på nytt, så uansett hva en side sa, stemplet én enkelt synkronisering hele nettstedet med samme tidspunkt. Datoen kommer nå fra versjonshistorikken gjennom en reservekjede som først prøver et eksplisitt felt, så commit-historikken, så filen. Siden skriver ned fellen i den kjeden: det bokstavelige feltnavnet må stå i listen, ellers leser generatoren det aldri. En manglende dato lar feltet stå tomt i stedet for å holde en gjetning, og på det dokumenterte implementeringstrinnet berørte en ny synkronisering etter en måned med innholdsendringer åtte filer av 186.

Selskapets implementering av gjenopprettingstester ender med at en fil leses tilbake. systemd-enhetene har støtte for en databasedump der dette er konfigurert, et kryptert, deduplisert restic-øyeblikksbilde sendt over SFTP til lagring hos en annen leverandør, opprydding etter oppbevaringsregelen og en integritetssjekk. En separat månedlig arbeidsflyt gjenoppretter en kjent fil til et midlertidig arbeidsområde og sammenligner bytene, slik at den tiltenkte testen gjelder gjenoppretting og ikke opplasting. Implementeringsdokumentasjonen datert 2026-10-27 sier at timerne var deaktivert uten et konfigurert sikkerhetskopirepo, og dokumenterer ingen fullført gjenoppretting i drift. Selv

passfrasen er en fil, med vilje: enhetene leser den fra en beskyttet fil i stedet for fra en miljøverdi tolket av systemd, der behandlingen av escape-sekvenser kunne få en omvendt skråstrek til å endre passfrasen i forhold til den som ble brukt til å opprette repoet.

År tidligere handlet jobben om å redde filer, ofte den eneste kopien, ut av maskinvare som hadde sviktet: datagjenoppretting fra SD-kort, roterende HDD-er, SSD-er, RAID-oppsett, eksterne disker og Mac-systemer. Hver av dem svikter og gir fra seg dataene sine på sin egen måte, og et Mac-filsystem har sine egne særheter, så arbeidet handlet like mye om riktig tilnærming for mediet som om noen enkelt teknikk. Målestokken var den enkle: filer levert tilbake i behold.

## Hvor det går galt

**Hver side ser ut til å være endret på én gang.** En kopiering, en checkout, en gjenoppretting eller en eksport skriver filen på nytt og setter endringstidspunktet til nå, og hvert system lenger ned i kjeden som stoler på det tidspunktet, ser en endring som aldri skjedde. En kopi som bevarer tidspunktene, gjør det motsatte og skjuler en endring som faktisk skjedde. Hent datoer for innhold fra innholdet selv eller fra versjonshistorikken, og la filtidspunktet bety det det betyr.

**En leser finner en halv fil.** Et program som skriver om en fil der den ligger, åpner et tidsvindu der en annen prosess, eller neste oppstart, ser en del av den gamle versjonen og en del av den nye. Skriv en midlertidig fil i den samme katalogen, kall `fsync` på den, og gi den deretter nytt navn over originalen. POSIX krever at navnet som erstattes, forblir synlig gjennom hele omdøpingen og hele tiden peker enten på den gamle filen eller på den nye.

**Et skript snubler i et lovlig navn.** Et filnavn kan inneholde mellomrom, linjeskift og en bindestrek helt først, og et skript som deler navn på blanktegn eller gir dem videre til en kommando uten anførselstegn, kommer til å handle på feil filer. POSIX råder selv til å unngå bindestrek først i navnet og til å holde seg til bokstaver, sifre, punktum, understrek og bindestrek for navn som skal kunne flyttes mellom systemer. Sett anførselstegn rundt hver ekspansjon, og avslutt tolkningen av opsjoner med `--` foran en liste med navn.

**En hemmelig fil kan leses av alle.** Tilgangsbitene på en ny fil kommer fra standardverdiene til prosessen som opprettet den, og de standardverdiene lar som regel gruppen og alle andre lese. En nøkkel eller passfrase som skrives på

den måten, deles med hver konto på verten. Opprett filer med hemmeligheter slik at bare eieren har tilgang fra første stund, i stedet for å stramme inn etterpå.

## Les mer

- NIST CSRC glossary: file — definisjonene NISTs sikkerhetspublikasjoner bruker, hver med dokumentet den kommer fra.
- POSIX.1-2024 Base Definitions: Definitions — den formelle betydningen av fil, vanlig fil, filnavn, stinavn, katalogoppføring og fildeskriptor.
- POSIX.1-2024 Base Definitions: General Concepts — tilgangsrettigheter, de tre filtidspunktene, filnavn som tåler å flyttes, og hva et krasj kan etterlate av data som ennå ikke er skrevet til disken.
- POSIX.1-2024: rename() — garantien om at et navn som erstattes, hele tiden peker på enten den gamle filen eller den nye, som trygg lagring bygger på.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

**Se hva det koster** Ta kontakt →

## Gjør begrepene praktiske

Spør hvordan «Hva er en fil?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

**Se tjenestene våre**

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

**Laget med kjærlighet av D.**

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

**Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.**

**Skrevet ut fra https://engineer.company/nb/glossary/file/**