

Gjennomgang av Unix og Linux filsystemet Basert på FHS 2.3

Jon Langseth

7. mars 2005

Intro

Denne teksten forsøker å ta for seg filsystemets oppbygning og organisering på et Unix[3] basert system, med direkte fokus på Linux[2]. Teksten er basert på standarden for organiseringen av et slikt system, the Filesystem Hierarchy Standard version 2.3 (FHS2.3)[1], men tillater seg enkelte avvik, der virkeligheten ikke stemmer med standarden.

Steg for steg går vi igjennom hvilke kataloger som må finnes i et FHS system, og hva som finnes der. Deretter tar vi en titt på de litt mer "eksotiske" ting, og hvor man kan forvente at ting havner når man installerer programmer. Først en liten presisering av noen sentrale ord.

Som de fleste som har benyttet et Unix basert system i mer enn noen minutter har erfart, så finnes ikke konseptet med "stasjoner", som det gjør under Microsoft[4] sine operativsystemer, og gjorde på eldre Apple[5] og Amiga systemer. Unix systemer, og alle som er i slekt med disse gjennom Posix-standard[6], har kun en "stasjon", og det er rot-katalogen, kjennetegnet med katalognavnet "/". Alle fysiske disk og nettverksstasjoner blir føyd inn i katalogtreet som brer seg ut fra roten, og det er dette som kalles å montere, eller "mounte" ting inn i filsystemet.

Et program utgjøres som regel av en kjørbart fil, som enten er hele programmet, eller startpunkt for en større samling funksjoner fordelt over flere filer. En slik kjørbart fil utgjør en kommando, og selve filen er enten et script bestående av leselig tekst, eller det som kalles en binær-fil. I tillegg til binære filer og scripts, vil de aller fleste programmer benytte seg av biblioteker og delte data. Et bibliotek er en samling funksjoner som er skilt ut av hovedfilen til programmet, for å kunne deles mellom flere programmer. Stort sett er det alltid slik at et program er bygget rundt ett eller flere biblioteker som er skrevet av andre enn de som har skrevet programmet. Delte data er informasjon som benyttes av programmet, men ikke inngår som en del av den kjørende koden. Eksempel på dette er ikoner som

benyttes i et grafisk grensesnitt. Til slutt benytter veldig mange programmer seg av konfigurasjons-filer, som bestemmer programmets oppførsel.

Alle disse delene av et program må lagres på fornuftige plasser på disk, og de bør helst grupperes. Dette er årsaken til at filsystemet oppbygning med kataloger og filer er vesentlig. Vi prøver her å gi et innblikk i hvor de forskjellige ting hører hjemme. Før forvirringen blir total, må det sies at en datamaskin som kjører et operativsystem, kan totalt sett kalles et *system*, mens det samme gjelder også for et program eller en samling programmer som tilsammen utfører en tjeneste på maskinen. Her benyttes disse to uttrykkene temmelig løst om hverandre, så når det nevnes *system*, kan dette bety enten maskinen i sin helhet, eller et programsystem som kjører på maskinen. Hvilken det menes bør gå frem av teksten.

/bin

bin katalogen inneholder de absolutt vesentligste kommandoer (binære filer). Denne katalogen inneholder kommandoer som er tilgjengelige for alle brukere av systemet, altså ikke begrenset til administrator et.al. /bin skal kun inneholde operativsystemets absolutt mest vesentlige kommandoer, altså de som er nødvendige under oppstart av systemet. Dette betyr at de filene som ligger i denne katalogen legges dit under installasjon av selve operativsystemet, og det legges sjelden til nye filer her. Kommandoer som ikke er vesentlige under oppstart havner i en katalog under /usr, som vi omtaler senere. Det er her du finner kommandoer som *bash*, *chmod*, *chown*, *chgrp*, *dd*, *grep*, *gunzip*, *kill*, *more*, *mount*, *ping* og *su*. I FHS2.3 er det spesifisert en rekke filer som *må* forekomme i denne katalogen.

/sbin

På samme måte som /bin er /sbin begrenset til kun å inneholde kommandoer som er vesentlige under oppstart og vesentlig vedlikehold (gjenoppretting/reparasjon) av systemet. Mens /bin katalogen inneholder kommandoer som skal være tilgjengelige for alle brukere, skal /sbin inneholde kommandoer som ikke skal brukes av vanlige brukere, men kun benyttes av system-administrator, root-brukeren eller av systemets initialiseringssystem. Siden denne katalogen kun skal inneholde system-kritiske kommandoer, gjelder det samme som for /bin, at alle ikke-kritiske kommandoer havner under /usr.

/lib

Dette er nok en av de katalogene som kun skal inneholde system-kritiske filer. Denne katalogen inneholder biblioteker som er vesentlige under oppstart og gjenoppbygging av systemet. Hva som er system-kritiske biblioteker og ikke er endel mer diffust enn hva som er kritiske kommandoer, og denne katalogen er endel mer fyldig enn hva `/bin` og `/sbin` er. Men veldig forenklet kan man si at de bibliotekene som inngår i en absolutt minimumsinstallasjon av et Linux system vil for det meste ligge i `/lib`, mens biblioteker som kommer etterhvert som man legger til applikasjoner og større systemer som X[15] ikke havner her.

/boot

Denne katalogen inneholder statiske filer for boot-loaderen, kan inneholde konfigurasjon for boot-loaderen, og det er blitt mer og mer vanlig at også operativsystem-kjernen plasseres i denne katalogen. På et nyere Linux system som benytter grub[7], vil konfigurasjon for denne ligge her, i underkatalogen grub (mao `/boot/grub/`), og det er blitt en de-facto standard at Linux kjernen legges her. Det er altså hit du skal kopiere din nykompilete kjerne-fil, og det er her du skal sette opp grub til å bruke denne nye kjernen ¹. Når det er sagt at denne katalogen inneholder boot-loaderen konfigurasjon, bør det også poengteres at det her ikke skal befinne seg andre konfigurasjons-filer, som ikke er vesentlige for lasting av operativsystemet.

/dev

Denne katalogen henter sitt navn fra det engelske ordet "devices". Katalogen inneholder filer som representerer datamaskinens enheter, på engelsk "Devices". De fleste har vært borti navn som `/dev/hda1`, dette viser altså til at `hda1` er en "device" på dette systemet. `/dev` katalogen var tradisjonelt statisk satt opp, og inneholdt et kjent sett filer, som alle representerte filer som *kunne* være tilstede. De fleste nyere Linux-distribusjoner i dag benytter det som kalles dev-filsystemet[16], et system som legger til og fjerner filer i denne katalogen etter hva som faktisk er tilstede av enheter. Bruk av `devfs`, medfører at det opprettes en lang rekke underkataloger under ved, for å representere typer eller klasser av enheter, eksempelvis `/dev/cdrom` for `cdrom`'er og `/dev/usb` for usb-tilknyttede enheter.

¹Tips: Ta alltid vare på den gamle kjernen din, og gi den nye kjernen din ett navn som forteller klart og tydelig hvilken versjon dette er, og når den ble kompilert. Det er ikke "farlig" å ha mange linjer i grub-konfigurasjonen.

/etc

Navnet på denne katalogen er i grunn litt misvisende, da forkortelsen kommer fra "et cetera", "og så videre / og lignende", mens innholdet er veldig spesifikt. Denne katalogen inneholder bortimot alle konfigurasjons-filer for systemer og programmer som er installert på et system. Denne katalogen er såkalt "host-spesifikk", det vil si at to maskiner kan ikke dele samme `/etc` katalog². I tillegg til å være host-spesifikk, er også katalogen system-kritisk. Dette betyr at den må være tilgjengelig for systemet under oppstart³.

Tidligere var det tillatt å legge små programmer og scripts i `/etc` katalogen, men dette er nå "forbudt". De eneste kjørbare filer som skal befinne seg i denne katalogen, er de scriptene som benyttes under oppstart av systemet, av programmet "init"[18]⁴.

I moderne tid, det vil si fra ca 1997, er det blitt vanlig at hver større program-pakke som trenger en generell konfigurasjon under `/etc`, legger sine konfigurasjons-filer i sin egen katalog herunder. Eksempelvis vil programpakken `openoffice`[8]⁵ legge felles konfigurasjoner i katalogen `/etc/openoffice`. For bruker-programmer som har konfigurasjon lagret under `/etc`, er det ikke korrekt å gjøre bruker-spesifikke endringer til filer i denne katalogen. Skal man overstyre en generell konfigurasjon for en enkelt bruker, skal dette gjøres i en såkalt dot-katalog i brukerens hjemmeområde. For å dra eksempelet videre vil da `openoffice` sine brukerspesifikke endringer skje i brukerens `/.openoffice` katalog. Et par av katalogene under `/etc` fortjener mer detaljert beskrivelse:

/etc/X11

Denne katalogen inneholder konfigurasjonen av motoren bak de fleste grafiske brukergrensesnitt på Linux systemer, nemlig the X Windowing System[15]⁶. Her finner du ikke konfigurasjonen til grafiske applikasjoner, men du finner alt av konfigurasjon for selve X systemet. Det er i denne katalogen du skal lete, dersom du for hånd skal endre oppløsning eller fargedybde, eller løse problemer du har med mus eller tastatur under X.

²Tenk tynn-klienter f.ex.

³At katalogen må være tilgjengelig under oppstart, betyr i praksis at den skal ligge på partisjonen som utgjør rot-filsystemet.

⁴init er et foredrag i seg selv....

⁵OpenOffice.org er korrekt navn

⁶Det er direkte feil å kalle X for X-Windows, det heter enten kun X, X11, X11R6, eller X Windowing System.

/etc/cron.*

Programmer som skal kjøres automagisk, i bakgrunn, for å gjøre automatiskerte vedlikeholdsoppgaver og oppdateringer, styres av cron. I cron-katalogene setter man opp hvilke programmer/scripts som skal kjøres, og når. Med anacron[17], en ”moderne” versjon av cron, benytter man katalogene `cron.hourly`, `.daily`, `.weekly` og `.monthly` for å enkelt sette opp programmer til å kjøre med disse intervallene.

/etc/init.d og /etc/rc*

`init.d` og `rc?.d` katalogene benyttes under oppstart av systemet. `rc?.d` inneholder kun linker til scripts som befinner seg i `init.d` katalogen. Disse scriptene tar seg av å sette opp systemet, og å starte og stoppe tjenester under oppstart og avslutning av systemet, meget forenklet ⁷

/media

Denne katalogen er kommet til i senere tid, sammen med de automatiserte systemene for håndtering av enheter i systemet. I samarbeid med `devfs`[16] og hotplug, vil du i denne katalogen finne fjernbare media, så som `cdrom`, `floppy`, `firewire` og `usb-disker`. Innholdet i `/media` vil endre seg alt ettersom hva som kobles til og fra systemet. I en ideell verden, vil systemet ditt automatisk oppdage at du har koblet til eller fra en ekstern disk, og mounte eller unmounte denne under `/media`, så det er her du skal inn dersom du skal få tilgang til den nylig tilkoblede iPod'en din.

/mnt

Ikke alle fjernbare media lar seg håndtere automatisk, og nettverks-disker er heller ikke alltid like enkle å ha med å gjøre. Derfor trenger du av og til (eller kanskje ofte) å manuelt mounte et filsystem midlertidig. Dette er nettopp hva katalogen `/mnt` er til for. Endel synes det er begrensende å kun ha en katalog for midlertidig mounting, og oppretter underkataloger av denne slik at de kan ha flere ”mounts” samtidig, og det er ikke noe direkte feil med dette. Men dersom du har en disk eller nettverksressurs som du benytter ofte, eller har mountet kontinuerlig, bør du vurdere å finne en annen lokasjon til dette, da `/mnt` skal være forbeholdt

⁷ref fotnote ang. `init`.

midlertidige mounts. Det er for eksempel en god praksis å flytte faste nettverks-mounts ut av `/mnt`, og opprette en annen katalog, f.ex `/net` til slikt.

`/srv`

Her skal etter hvert alle filer som er relatert til tjenester (services) tilgjengelig på maskinen plasseres. Med dette menes filer for ftp, http og cvs, som eksempler. Dette er en helt ny katalog i FHS2, og det er slik sett ikke satt noen gjeldende standard for hvordan innholdet i denne skal struktureres. Med andre ord, her kan det hende, hvis maskinen er satt opp med apache web-serveren, at du finner alle filer som er tilgjengelig via http. Men bare hvis distribusjonen din har satt opp apache[19] til å faktisk benytte `/srv` katalogen. Vi håper dette går seg til etter hvert, slik av vi får en fast logikk i dette. Tanken bak denne er ikke direkte ulik den gamle `/exports` katalogen, som i teorien skulle inneholde alle kataloger som skulle deles ut via NFS. Om ideen med `srv` følges ut, bør nå ting som skal være tilgjengelige via nfs havne i `/srv/nfs`

`/home`

Siden Unix systemer har et såpass strukturert og restriktivt oppsett på hvilke filer som havner hvor, og siden Unix systemer støtter flere brukere, trenger man å gi hver bruker sitt område, hvor han kan legge sine filer, uten å være redd for å rote til for systemet eller andre brukere. Dette kalles brukerens hjemmekatalog, og finnes (vanligvis) med brukeren brukernavn under `/home`. Slik sett, dersom din bruker heter `ola-norm` har du hjemmekatalogen din som `/home/ola-norm`. Det er i denne katalogen alle *dine* filer havner, og det er her du finner alle program-konfigurasjoner som gjelder deg. Du kan sjelden se disse, da de er "skjult", men kommandoen `ls -la` vil vise deg at de finnes. For de som tør å begi seg inn i et kommando-shell (noe du bør), er `$HOME` og `~` praktiske kort-versjoner av banen til hjemmekatalogen din. Du kan med andre ord raskt benytte kommandoen `cp en.fil ~` for å kopiere filen "en.fil" til hjemmekatalogen din. Hva du gjør og hvilket rot du har i hjemmekatalogen din, er din sak, men Gnome[9] og KDE[10] oppretter katalogen `/Desktop` for deg, som er skrivebordsflaten din. Denne bør du ikke slette.

`/root`

Dette er hjemme-katalogen til root-brukeren, og er i grunn en katalog som er blitt lagt til for å pynte opp litt. Det utsagnet krever nok litt forklaring. Opprinnelig

var det slik at root-brukeren hadde, slik han har nå, all makt på systemet. Han var bokstavelig talt rot-brukeren, og hadde hele systemet som sin hjemmekatalog. Når root logget inn, ble han plassert direkte i / katalogen, og dette var hans \$HOME, eller ~. Dersom en villfaren systemadministrator så begynte å logge inn som root, og benytte systemet, ble / fylt med masse brukerdata tilhørende root, noe som ikke er direkte heldig, og heller ikke særlig pent. Derfor ble /root katalogen opprettet, og root brukeren flyttet inn der. Akkurat som alle andre hjemme-kataloger bør være satt opp, skal /root være satt slik at kun root-brukeren kan aksessere denne katalogen og dens innhold⁸.

/tmp

Alle datamaskiner har behov for å opprette midlertidige filer av forskjellige slag, og /tmp er plassen for midlertidige filer som ikke trenger å taes vare på når maskinen slås av eller rebootes. Denne katalogen skal være skrivbar for alle, og bør strengt tatt ikke inneholde kjørbare filer. Ofte benyttes tempfs på denne katalogen, dette er en form for ram-filsystem, som eksisterer fullstendig i RAM, og dermed automagisk tømmes når maskinen rebooter eller slås av. Midlertidige filer som ikke skal forsvinne selv om maskinen mister strøm eller rebootes, skal ikke ligge her, de hører hjemme i /var/tmp.

/proc

Under Linux dukker katalogen /proc alltid opp. Linux benytter denne katalogen til kernel, system og prosessinformasjon. Dette er en katalog som har svært dynamisk innhold, og det er en svært dum ting å prøve å kopiere innholdet av denne katalogen, da du vil finne blant annet en direkte tilgang til hele minnet som er i bruk av systemet, også til kopieringsjobben du eventuelt prøver deg på, samt en direkte kloner av hele filsystem-treet, slik kjernen ser det. Prøver du å kopiere /proc, vil du forsøke å kopiere alt som foregår i din maskins minne, og du vil gjøre det rekursivt. Denne katalogen inneholder ingen filer som lagres på disk, alt er en virtuell representasjon av hvordan kjernen oppfatter systemet. /proc er en uvurderlig katalog dersom man med litt kunnskap vil finne ut hva som skjer internt i systemet. Blant annet finner du her filer som /proc/interrupts som forteller deg hvilke kjente IRQ'er som finnes, og hva de brukes til, du finner /proc/devices som viser hvilke kjente enheter som finnes. En ekskursjon i /proc med kommandoen less ved din side, vil kunne avsløre ganske spennende, og nyttige ting. Men vær

⁸Den personen som oppretter en public_html eller annen generell lesbar katalog under /root bør skytes

forsiktig. Disse filene skal du *ikke endre* med mindre du vet nøyaktig hva du driver med, og noen av disse filene er potensielt *enorme*.

/usr

En del av de katalogene som er omtalt hittil, har hatt kommentar om at de kun inneholder vesentlige, kritiske, eller host-spesifikke filer, og ikke skal inneholde generell programvare. Det er her `usr` treet kommer inn i bildet. `/usr` inneholder et fil-hierarki under `fil`-hierarkiet, og gjentar en del av de kjente navnene fra tidligere. Det er her alle filer tilhørende programmer og pakker som installeres på toppen av det helt grunnleggende havner. Navnet kom opprinnelig fra "user", og indikerte at her lå "user-accessible packages" og "user installed data".

Dette er plassen man skal begynne å lete, for å finne igjen filer tilhørende programmer man har installert, med unntak av konfigurasjons-filene. Disse havner som vanlig i `/etc`, eller i brukerens hjemmekatalog. Under `/usr` gjentas som sagt en del av de katalogene man finner direkte på roten, og betydningen av disse er noenlunde lik. `/usr/bin` inneholder alle ikke-kritiske binære filer, altså kommandoer eller programfiler, som er kjørbare av alle brukere på systemet, `/usr/sbin` inneholder alle ikke-kritiske binære filer som kun skal benyttes av systemadministrator.

`/usr/lib` inneholder alle biblioteker som ikke er påkrevet under oppstart, så hvis du installerer en pakke med et bibliotek for grafiske applikasjoner under X, så vil du finne dette biblioteket igjen enten som filer direkte under `/usr/lib`, eller som en egen katalog her. For de som benytter script-språk til å programmere eller utvikle forskjellige verktøy, er katalogene `/usr/lib/perl` og `/usr/lib/python*` seg vel verdt å merke, da det er inn i disse alle tillegg, moduler og pakker for disse språkene skal havne.

Ut over `/usr/bin`, `/usr/sbin` og `/usr/lib` finnes det endel kataloger som er verdt å titte på, en etter en.

/usr/games

Denne katalogen ble i "gamle dager" brukt til spill som ble installert, og til en viss grad er dette fortsatt tilfelle. Gnome legger nemlig binær-filene til sine spill i denne katalogen. Ut over dette er katalogen lite i bruk, men det anbefales at den er der av historiske hensyn. KDE benytter ikke denne katalogen.

/usr/include

Dersom du driver med programutvikling, spesielt om du benytter C eller C++, er denne katalogen essensiell. Den inneholder nemlig de aller fleste header-filer

som benyttes under kode-utvikling og kompilering. På distribusjoner som skiller mellom rene bibliotek-pakker og utvikler-pakker (-dev), vil innholdet av utvikler-pakken være header-filer som havner i denne katalogen. Her finner du også den veldig vesentlige katalogen `/usr/include/linux`, som er vesentlig for all kompilering av kjerne-relaterte programmer og tillegg.

`/usr/src`

Hvis vi ser bort fra at RPM[11] baserte Linux distribusjoner benytter `/usr/src` til å behandle SRPM pakker, er den definitivt viktigste grunnen til å merke seg `/usr/src`, at det er en de-facto standard⁹ å legge kildekoden til Linux kjernen her, i en katalog med navnet `/usr/src/linux-<ver>` med en link til `/usr/src/linux`. Det finnes verktøy som faktisk baserer seg på dette, og spesielt er det enkelte driver-pakker som ikke er standard del av kjernen, som må kompileres fra bunn, og avhenger av dette oppsettet. Laster du ned Linux-kjernen som en kildekode-pakke fra f.eks. `kernel.org`, er det slik du bør plassere den, siden de aller fleste dokumenter som forklarer hvordan du kompilerer din egen kjerne, baserer seg på at dette er tilfelle.

`/usr/share`

Under denne katalogen finner du et vell av underkataloger, som hver for seg tilhører installerte programmer og pakker. Dette er plassen for såkalt "application shared data" som ikke endres av brukere. Merk at dette er delte data, det er ikke data som genereres og endres under kjøring. Dette har en annen plassering, som vi kommer tilbake til. I tillegg til å holde på application shared data, er det er også kilden som bruker-spesifikk data som havner i brukernes *dot-kataloger* stort sett kopieres ut fra. Et ofte tilbakevendende spørsmål blant brukere som ikke ennå er helt kjent med Linux og moderne Unix systemer, er "hvor havner alle data som hører til dette programmet hen?", og svaret på dette spørsmålet er "her". Dersom man benytter KDE[10] er dette ikke helt sant, da applikasjonsdata for KDE havner i underkatalogen `apps`.

I `/usr/share` finner man i tillegg katalogene som inneholder ikoner, bakgrunnsbilder og themes som er tilgjengelige for alle brukere på systemet, og man finner katalogene som inneholder dokumentasjon, i form av manual-sidene til programmet man, og i form av tekst-filer i katalogen `/usr/share/doc`. Dokumentasjonen her skal være inndelt slik at hver installert program-pakke har sin katalog med dokumentasjon, med samme navn som pakken.

⁹Faktisk er det endel som anbefaler å ikke gjøre dette

/usr/X11R6

X Windowing System har fått lov til å få nok et hierarki under et hierarki¹⁰, i katalogen `/usr/X11R6`¹¹. Her har X sine egne `bin`, `lib`, `include` og man kataloger. Disse er igjen linket til `/usr/bin/X11`, `/usr/lib/X11` og `/usr/include/X11`, respektivt. Disse katalogene inneholder alt som er inkludert i den grunnleggende X11 pakken, deriblant X-serveren og driver-modulene, grunnbibliotekene (så som `Xaw` og `Xv`), og de enkleste grunnprogrammene som `xterm` og `xload`. Inn i disse katalogene er det strengt tatt kun ting som er en integrert del av X11R6 som skal legges, mens programmer som benytter X11 fremdeles skal havne i `/usr/bin` med familie.

/usr/local

Her dukker det opp noe som ved første øyekast ser ut som en kopi av strukturen i `/usr`, og som ligger under den igjen? Hva kan denne være godt for? Om man skal være riktig så pedantisk, eller i det minste ryddig, så er det slik at innholdet i de katalogene som ligger direkte på roten, med unntak av `/etc`, bør styres og oppdateres av operativsystemet, og innholdet i `/usr` bør styres av operativsystem distribusjonens pakkesystem. Av denne grunn bør man ha en egen plass hvor man installerer såkalt lokalt installerte programmer og biblioteker, altså ting man installerer for hånd, fra kildekode. Derfor finnes `/usr/local`.

Inn i denne katalogen, i en struktur som er lik den man finner på roten, og i `/usr`, installerer man altså alt som man legger inn for hånd. Dette gjør det veldig ryddig å skille mellom programmer som andre har pakket for deg, og programmer du legger inn. I tillegg sikrer dette at ting du legger inn manuelt, ikke blir overskrevet, eller endret av systemets pakke-håndtering. Faktisk er det av denne grunn, at du her også finner katalogen `/usr/local/etc`. Her havner konfigurasjonsfilene til ting som er installert under `/usr/local`, i stedet for i `/etc`. Dette er en veldig vanlig ting å ikke være klar over, og har ført til at man ofte leter lenge og vel etter konfigurasjoner, eller ender opp med konfigurasjoner som ikke har noen virkning, når disse rett og slett havner på feil plass.

/opt

Her kommer en litt snodig sak. Vi har `/usr` for å installere programmer og slikt som ikke er en del av det vesentlige systemet, og vi har `/usr/local` til

¹⁰Det er verdt å nevne at dette er gjengs praksis for andre vindus-systemer. Eksempelvis finner man grunnlaget for X, i form av OpenWindows på Solaris, i katalogen `/usr/openwin`

¹¹Både XFree86 og X.Org er basert på X Windowing System version 11 release 6.

ting vi installerer for hånd, fra kildekode. Og i tillegg har vi katalogen `/opt`, som eksisterer for installasjon av "optional packages". I denne katalogen havner nemlig program-pakker som ønsker å samle alle filer og elementer som inngår i programmets oppbygning på en plass. Dersom en programvareleverandør ikke ønsker å spre programmet sitt utover i `/usr` katalogene, og slik sett "blande" sine filer med andres, vil de isteden legge inn pakken her, i en egen katalog, med samme navn som programpakken. Eksempelvis er det på enkelte Linux versjoner, og spesielt på Unix versjoner som Solaris[20], normalt at *Sun Java*[21] installeres i `/opt/j2re-1.4.x` og *Mozilla*[22] installeres i `/opt/mozilla`. Dette er en praksis som er mye brukt av kommersielle programvareleverandører. For å underlette tilgangen til programmene, finnes normalt sett `/opt/bin` som inneholder linker til de programfilene som trengs for å starte programmet, og `/opt/lib` som inneholder linker til de vesentligste biblioteks-filene. Det er verdt å nevne at på endel distribusjoner finnes `/opt`, men er tom.

`/var`

Det ble tidligere nevnt at data og filer som genereres og endres av programmer som kjører på systemet, ikke havner i `/usr/share`. Slike data havner enten i brukerens hjemmekatalog, dersom det er et program som kjøres interaktivt av en bruker. Men dersom det er et program som kjøres av systemet, så som en server-tjeneste eller et vedlikeholdsprogram, så ender disse data opp under katalogen for "variable data", altså `/var`. Her havner administrative data, logger, cache-filer og spoling-filer, blant annet. Katalogen har en fast struktur, som sorterer forskjellig type informasjon.

`/var/run`

Alle programmer som kjører på en maskin, og som er startet av operativsystemet gjennom `init`, skal opprette en liten fil i denne katalogen, som skal hete *programnavn.pid*. Denne filen skal inneholde en linje, som kun skal inneholde prosessnummeret til programmet. Dette benyttes til å holde styr på programmene, og til å forenkle jobben med å starte/stoppe tjenester.

`/var/cache`

Dette er plassen for såkalt chached data, eller "bufret informasjon" som skulle bli en passende oversettelse. Hit lagres programgenererte data som kan, og bør gjenbrukes for å spare tid. Eksempler på dette er fonter som genereres av X, man-sider som genereres ved første gangs lesning, pakker som lastes ned og behandles

av pakkesystemer[25] og filer tilhørende web-proxy[23], om en slik kjører på maskinen.

/var/lib

All data som genereres av et kjørende program, eller et bibliotek som genererer sentrale variable data, skal lagres i en katalog med samme navn som applikasjonen eller biblioteket under `/var/lib`, med unntak av logg-filer, og spool-filer. Det er med andre ord her du finner de variable data til for eksempel MySQL[24] og APT[25].

/var/log

Dette er katalogen hvor alle logg-filer skal havne. De mest vesentlige er nok de som genereres av `syslog`, men her finner du også loggene til andre programmer. For eksempel er det anbefalt at logg-filene til apache havner her, i sin egen katalog. Kort sagt, dersom du er på jakt etter en eller annen logg-fil, så bør den ligge her.

/var/lock

Dersom et system-program av en eller annen grunn trenger å opprette en form for lås, for eksempel for å sikre at ikke mer enn en instans av programmet kan kjøre om gangen, skal denne låsen opprettes gjennom en lock-fil i `/var/lock`. Dette gjelder altså ikke brukerprogrammer, med unntak av de som kjører med rettighetene til root. Eksempel på dette er `minicom`[26], en terminalemulator, som må sikre at maksimalt en bruker forsøker å benytte en serieport om gangen.

/var/spool

Spool-filer er eksempelvis mellomagringsfiler for det systemet som håndterer printing på en maskin. Data "spoles" opp i en fil, for så å sendes litt etter litt ut til printer. Alle systemer som benytter seg av spooling prinsippet, skal ha en katalog under `/var/spool`, som skal benyttes til å holde slike filer.

/var/mail

Denne katalogen er faktisk på veg ut, og er i FHS 2.3 valgfri. Tradisjonelt var det her alle brukeres mailbox-filer (e-post-kasser) befant seg. Siden bortimot alle moderne epost-systemer nå benytter seg av det såkalte *Maildir*[27] formatet, med en katalog i brukerens hjemmekatalog, trenger ikke lenger `/var/mail` å eksistere. For å unngå forvirring, må det sies at det er snakk om mail som er levert til

brukeren av f.eks. sendmail eller postfix lokalt det er snakk om, altså ikke de mail du plukker ned med en epost-leser.

/var/tmp

Under omtalen av /tmp nevnte vi hensikten med katalogen /var/tmp. Denne katalogen skal altså inneholde temporære, transiente data, som skal beholdes selv om maskinen rebootes eller slås av.

/var/local

Dette er "var-tree" til ting som er installert i /usr/local. Her finner du altså en replikering av /var sin oppbygning, som benyttes av "lokalt installerte programmer og pakker". Dersom du husker tilbake til omtalen at /usr/local, er du sikkert med på hva jeg snakker om.

/var/opt

På samme måte som /usr/local har en ekvivalent /var/local, finnes /var/opt som en katalog som skal benyttes av programmer installert i /opt. Hver pakke skal her benytte sin egen underkatalog, tilsvarende prinsippet for /opt

Partisjonering

Som det ble nevnt innledningsvis, er det slik med Unix familien at disker, partisjoner og andre former for logiske lagringsenheter ikke gis stasjonsnavn, men føyes inn i det resterende tree. Dette gir oss muligheten til å spre systemet utover flere disker eller partisjoner, eller tilogmed hente deler inn fra en server. Av forskjellige grunner er det veldig utbredt å stykke opp en fysisk disk i flere partisjoner, og fordele filsystemet utover disse. Og da dukker spørsmålet opp "Hvordan bør jeg gjøre dette mest, for å utnytte disken best mulig"? Akkurat dette er det like mange meninger om, som det finnes litt over middels erfarne Unix brukere. Her tas utgangspunkt for argumentasjon i forbruket på en gjennomsnittlig installert Ubuntu[12] Warty arbeidsstasjon, en ikke fullt så standard oppsatt Debian[13] web, sql og mail server, en temmelig generisk Debian fil-server (NFS, samba[28] og FTP) samt forfatterens personlige preferanser.

Men før vi går i gang med et partisjoneringsoppsett, må det fortelles at enkelte av katalogene i filsystemet ikke kan eller skal skilles ut på egen partisjon. Dette er de katalogene som er vesentlige for systemet under oppstart, etter at kjernen

er lastet. Disse katalogene må ligge på den partisjonen som utgjør roten for filsystemet:

- /bin
- /dev *
- /etc
- /lib
- /proc *
- /sbin
- /tmp *

* 12

De katalogene som derimot er aktuelle å skille ut på egne partisjoner, er oppgitt i tabellen nedenfor, med anslått størrelse på henholdsvis en gjengs brukt arbeidsstasjon og en liten server. Merk at det er langt fra nødvendig å partisjonere ut alle disse, men at man da må passe på å gi tilstrekkelig rom på rotpartisjonen. Tallet for rot-partisjonen er gitt, under forutsetning at de nedenforstående katalogene er skilt ut. Der det kun står strek, indikerer dette at katalogen er tom.

Katalog	Arbeidsstasjon	Web/mail-server	Fil-server
/	150 MB	50 MB	100 MB
/home	9 GB	4 GB	2 GB
/opt	50 MB	–	–
/srv	–	1 GB	350 GB
/usr	2 GB	700 MB	700 MB
/var	400 MB	2 GB	600 MB

Et godt prinsipp er å skille /boot ut på en egen partisjon, og helst den første på den primære disken. Dette for å sikre at kjernen og boot-loader konfigurasjonen alltid ligger først på disk, og det gjør systemet noe mindre sårbart for feil i filsystemet andre steder. Denne partisjonen kan være veldig liten, men for å gi rom, anbefales det at denne havner mellom 25 og 50 MB. Personlig setter jeg av 75 MB til dette. I tillegg til de synlige partisjonene, må du alltid sette av plass til swap. Dette bør ikke være den første partisjonen på disk, jeg pleier å legge den inn som partisjon nummer tre, etter /boot og roten. Størrelsen på denne ble det historisk

¹²Disse katalogenes innhold vil stort sett uansett bli satt opp og administrert av de dynamiske ram-baserte filsystemer *devfs*, *procfs* og *tempfs*

sagt at skulle være to ganger størrelsen på RAM, i disse dager uttaler mange¹³ at swap kan tendere til å fungere effektivt dersom RAM og SWAP har ulik størrelse, og RAM mengden er 384 MB eller mer. Resten av teksten tar forutsetning i at disse to partisjonene, med deres størrelse, er satt av.

På en arbeidsstasjon, er å gjøre hele denne oppstykkningen i min mening langt fra nødvendig. Hovedgrunn til å gjøre det, er for å oppnå mindre blokkestørrelser, og dermed kunne utnytte plassen mer effektivt. Realiteten er at den plassen du tjener inn ved å lage mindre partisjoner er minimal, og du må fremdeles ta hensyn til at selve partisjoneringen tar plass. Det er langt mer effektivt å tenke over hvilken filsystem-type du velger, enn å ha mange små partisjoner. Mye plass kan for eksempel vinnes ved å benytte reiserfs[29] på partisjoner som inneholder mange små filer, og ext3 på disker med store filer.

På en arbeidsstasjon vil jeg anbefale at kun /home skilles ut. Dette gjør at du kan re-installere systemet ditt, uten å være redd for at data går tapt, mens du slipper hodebry med å passe på hvorvidt en programpakke kommer til å få plass eller ikke på de forskjellige partisjonene. Dersom du skulle få trøbbel med rot-filsystemet ditt på en arbeidsstasjon, er ikke det så kritisk, hvis du enkelt kan redde innholdet av hjemmekatalogene. Sett av rikelig med plass til rot-filsystemet, og bruk resten til brukerne. Med utgangspunkt i tabellen over, ser vi at et slikt oppsett vil gi et minstekrav i størrelse på 2.6 GB for rot-partisjonen. Gi systemet noe ekstra plass, for eksempel et par GB om du har det til rådighet, og gi resten til brukeren din.

Når det gjelder servere begynner partisjoneringsspørsmålet å bli mye mer komplekst. Hvilke størrelser som bør velges avhenger veldig av hvilke typer tjenester som skal tilbys, hva slags data disse tjenestene håndterer, og hvordan serveren legges opp. Det som er sikkert, er at for servere, har det en større hensikt å stykke opp i flere partisjoner og/eller disker. Ved å separere data på forskjellige logiske enheter, oppnår man bedre data-separasjon, bedre data-sikring og potensielt bedre data-integritet. Disse siste to momenter kan forklares ved at filer som ligger på en logisk enhet ikke er offer for feil som oppstår i filsystemet på en annen logisk enhet, og systemet blir lettere å gjenopprette etter for eksempel innbrudd, hvis man klarer å sikre de separate logiske enhetene godt. I tillegg oppnås en potensiell hastighetsforbedring, ved at operativsystemet har begrensede områder å lete etter filer, dersom man skulle få en høy grad av fragmentering¹⁴ i filsystemene.

Inndelingen i tabellen er en god pekepinn på hvordan man bør partisjonere for en server, med tanke på hvilke kataloger som skal skilles ut. Men størrelsen kommer som sagt veldig mye an på hvilke tjenester som tilbys, og hvor disse systemene lagrer sine filer. Det som kan sies sikkert, er at det bør settes av en

¹³Dette har forfatter selv erfart

¹⁴Fragmentering innebærer at en fil er stykket opp og lagret på flere plasser på disk, og at filer som egentlig ligger i samme katalog, ligger logisk spredd blant andre på disk.

liten partisjon til rot-filsystemet. En størrelse på denne på 250 MB holder, og gir man den godt med plass, klarer man seg lenge med 500 MB, under forutsetning at oppsettet i tabellen følges. Katalogen `/usr` avgjøres av hvor mange programmer og lignende som kommer til å installeres og kjøres på serveren, men per i dag bør man revurdere hvordan serveren er konfigurert, dersom plassbehovet her overstiger et par GB.

Deretter må man tenke etter hva slags tjenester man tilbyr. Skal man tilby tjenester basert på data lagret i brukeres hjemmekataloger, bør denne få størst plass. Dersom derimot man kjører tjenester som baserer seg på `/var`, så som MySQL[24] eller postgresql[30], bør man sette av en rimelig porsjon til brukere, og deretter den største plassen til `/var`. Om tjenestene man tilbyr i all hovedsak holder data i `/srv` katalogen, bør den største plassen legges der.

Om jeg skal gi et konkret eksempel på partisjonering av en 250 GB stor disk, for bruk til en "hjemme-server" med 512 MB RAM, som skal håndtere innkommende mail for fire brukere, hvor mail lagres i brukerens `/Maildir`, kjøre apache med PHP[31] og MySQL for å håndtere en blogg med databasen i `/var` og html-dokumenter i `/srv/http`, og en NFS/samba/ftp server med diverse mediafiler og lignende, vil oppsettet bli omtrent slik:

Katalog / partisjon	Størrelse
<code>/boot</code>	75 MB
<code>/</code>	500 MB
<code>swap</code>	512 MB
<code>/home</code>	20 GB
<code>/srv</code>	215 GB
<code>/usr</code>	2 GB
<code>/var</code>	10 GB

De manglende megabytene i dette oppsettet, vil også sannsynligvis bli manglende i et forsøk på å gjennomføre dette oppsettet, da angitt størrelse på disk, og reell utnyttbar plass, sjelden stemmer overens. Mest sannsynlig vil `/srv` partisjonen bli mindre enn hva som er angitt her.

Referanser

- [1] The Filesystem Hierarchy standard version 2.3
<http://www.pathname.com/fhs/pub/fhs-2.3.htm>
- [2] The Linux operating system official web site,
<http://www.linux.org/>
- [3] Unix operativsystem familien, samt et registrert varemerke
<http://www.unix.org/>
- [4] Microsoft Corporation, utviklere av bl.a. MS DOS og MS Windows
<http://www.microsoft.com/>
- [5] Apple Computers
<http://www.apple.com/>
- [6] Portable Operating System Interface,
nytt navn er The Open Group Base Specifications
<http://www.opengroup.org/onlinepubs/009695399/toc.htm>
- [7] Grand Unified Boot Loader
<http://www.gnu.org/software/grub/>
- [8] Kontorpakken OpenOffice.org
<http://www.OpenOffice.org/>
- [9] GNU Network Object Model Environment
<http://gnome.org/>
- [10] the K Desktop Environment
<http://www.kde.org/>
- [11] RPM Package manager
<http://www.rpm.org/>
- [12] Ubuntu, Linux for Humans
<http://www.ubuntulinux.org/>
- [13] the Debian Project, opphavet til Debian GNU/Linux bl.a.
<http://www.debian.org/>
- [14] RedHat *<http://www.redhat.com/>*

- [15] the X Windowing System 11 release 6
tilgjengelig fra XFree86 på <http://www.xfree.org>
og fra the X Consortium som X.org på <http://www.X.org/>
- [16] Device File system
<http://seth.positivism.org/man.cgi/devfsd>
- [17] Anacron, anachronistic scheduler
<http://anacron.sourceforge.net/>
- [18] System V init
<http://www.freeos.com/articles/3243/>
<http://freshmeat.net/projects/sysvinit/>
- [19] Apache web server
<http://www.apache.org/>
- [20] Sun Solaris operating system <http://www.sun.com/software/solaris/>
- [21] Sun Java technology
<http://java.sun.com/>
- [22] The Mozilla project
<http://www.mozilla.org>
- [23] Squid Web cache
<http://www.squid-cache.org/>
- [24] MySQL database server
<http://www.mysql.com/>
- [25] Debian Advanced Package Tool
<http://www.debian.org/doc/manuals/apt-howto/index.en.html>
- [26] Minicom communication program
<http://alioth.debian.org/projects/minicom/>
- [27] Maildir format for lagring av innkommet epost
<http://cr.yp.to/proto/maildir.html>
- [28] samba SMB/CIFS server
<http://www.samba.org/>
- [29] Reiserfs, by Hans Reiser
<http://www.namesys.com/>

[30] PostgreSQL database
<http://www.postgresql.org/>

[31] PHP (Hypertext preprocessor)
<http://www.php.org/>

creative commons

Public Domain Dedication

Copyright-Only Dedication (based on United States law) or Public Domain Certification

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.