

Forord

Statleg forvaltning vert i stadig større grad einsretta i valet sitt av IT-produkt. Statskonsult ynskjer å peike på kva for ulempe ein for stor einsretting kan medføre, og vi vurderar alternativ til dei løysingane som allereie eksisterar.

Denne rapporten tek for seg eit spesielt fenomen innafor programvareutvikling som vert kalla "Open Source Software", eller open programvare som vi kallar det. Eit døme på open programvare er operativsystemet Linux.

I rapporten har vi vurdert om open programvare kan gjere at forvaltninga ikkje vert avhengig av leverandørane, og om ho kan redusere dei totale IT-kostnadene.

Rapporten har vorte skriva over eit lengre tidsrom samstundes som vi her i Statskonsult har gjort eigen utprøving av ulike typar open programvare. Dei fleste i IT-avdelinga i Statskonsult har vore med samt intern IT-drift og interesserte i andre avdelingar. I tillegg har vi henta inn synspunkt frå andre delar av forvaltninga. Prosjektleder har vore seniorrådgjevar Endre Grøtnes. Anne-Lise Halgunset har omsett rapporten til nynorsk.

Oslo, desember 2001

Jon Blaaid

Innhald

1	Samandrag	4
2	Innleiing	5
2.1	Bakgrunn	5
2.2	Leverandøruavhengig	6
2.3	Mål og gjennomføring	7
2.4	Om rapporten	7
3	Open programvare	8
3.1	Kva er open programvare?	8
3.2	Hovudprinsippa til open programvare	8
3.3	Forskjellen mellom open programvare og andre programvarar	9
3.4	Kort om opne programvarelisensar	9
3.5	Døme på open programvare	10
3.5.1	Linux	11
3.5.2	Apache	11
3.5.3	Bind	11
3.5.4	Perl	12
3.5.5	Sendmail	12
3.6	Utviklingsmetodikken for open programvare	12
3.7	Kva for programvare er mest eigna for utvikling som open programvare	13
3.8	Tenkjelege fordelar med open programvare	14
3.9	Tenkjelege ulemper med open programvare	15
4	Linux	17
4.1	Kva er ein Linux-distribusjon?	17
4.2	Linux-kjernen	17
4.3	Designmål for Linux	17
4.4	Grafisk brukargrensesnitt	18
4.5	Kompilatorar	18
4.6	Programvareutvikling	18
4.7	Lisensvilkåra for Linux	19
4.8	Ulike distribusjonar	19
5	Økonomien ved open programvare	21
5.1	Forretningsmodellar for open programvare	21
5.1.1	Skape eit merkenamn og distribuere ei pakke med open programvare	21
5.1.2	Selje open programvare som ein sjølv er hovudutviklar av	21
5.1.3	Proprietære produkt som aukar verdi av den opne programvaren	21
5.1.4	Konsulenttenester	21
5.1.5	Selje utstyr som høyrer til	22
5.2	Finansiering av open programvareutvikling	22
5.2.1	Eksternt finansierte utvikling	22
5.2.2	Internt finansierte utvikling	22

5.3	Tydinga som open programvare har for dei totale IT-kostnadene	23
6	Døme på bruk av open programvare i forvaltninga	25
6.1	Universitetet i Oslo	25
6.2	Høle barne- og ungdomsskule	25
7	Kommersielle tenester for open programvare i Noreg	26
7.1	Kjøp av programvare	26
7.2	Kjøp av maskinar med ferdig installert programvare	26
7.3	Konsulenttenester	26
7.4	Kurs og opplæring	26
8	Bruk av open programvare som standardiserings- og utviklings-tiltak	27
8.1	IETF i open programvare	27
8.2	Internett og open programvarereferanseimplementeringar	27
8.3	Open programvare som katalysator for utbreiing av nye standardar	27
9	Vurdering av bruksområda til open programvare	29
9.1	Vurderingskriterium for open programvare	29
9.2	Kva for delar av forvaltninga er best eigna for bruk av open programvare?	30
9.3	Kva for type open programvare er eigna i forvaltninga?	30
9.3.1	Sluttbrukarprogramvare	30
9.3.2	Tenarprogramvare	31
10	Konklusjonar, moglege tiltak og tilrådingar	32
10.1	Konklusjonar	32
10.2	Moglege tiltak	33
10.2.1	Påleggje bruk av open programvare	33
10.2.2	Støtte bruken av open programvare	33
10.2.3	Distribuere og utvikle open programvare	34
10.2.4	Støtte forskning og utvikling av open programvare	34
10.3	Tilrådingar	34
Vedlegg I	Referansar	36
Vedlegg II	Forkortingar	37
Vedlegg III	GPL (GNU General Public License)	38
Vedlegg IV	The BSD License (Berkeley Software Distribution)	45

1 Samandrag

Pressa, fagmiljø og politikarar har i den seinare tida fokusert på Linux som eit alternativ til Windows. Av di, og fordi Statskonsult er oppteken av å utgreie høve som kan redusere avhengnaden til leverandørane, og sjå nærare på løysingar som kan redusere kostnadene ved kjøp av programvare, har denne rapporten vorte til.

Linux er eit opent, fritt tilgjengeleg operativsystem som er utvikla som "Open Source"-programvare, eller *open programvare*. Open programvare er ein programvare der kildekoden er fritt tilgjengeleg, og programmet kan nyttast, endrast, forbeholdt, utvidast og redistribuerast av alle som ynskjer det.

Gjennom innhenting av informasjon, samtalar med aktørar og eigen utprøving har Statskonsult vurdert bruken av Linux og annan open programvare. I tillegg til å kunne seie noko om korleis open programvare kan nyttast i forvaltninga, har vi dokumentert kva open programvare har å seie for utviklinga av IT-infrastrukturen.

Statskonsult meiner at Linux og annan open programvare har eit stort potensial. Det er i hovudsak på område som utvikling av infrastruktur, utarbeiding av standardar, og i undervisning og opplæring. Det er førebels ikkje eit tilstrekkeleg tilbod av open sluttbrukarprogramvare, slik at forvaltninga kan erstatte den eksisterande programvaren sin med desse. Det finnst til dømes ingen system for Linux når det gjeld rekneskaps- eller verksemdstyring.

Open programvare står i dag fram som ein viktig bidragsytar til vidareutviklinga av Internett-infrastrukturen. Dei grunnleggjande standardane for overføring av data mellom maskinar, sending av e-post og omsetjing av adresser på Internett er alle tilgjengelege som open programvare. Standardane er også utvikla og spesifisert gjennom ein open prosess.

Tilrådingane om open programvare kan oppsummerast i desse punkta:

- Bruk av open programvare for å redusere kostnader vil vere føremålstenleg i verksemdar der dei totale IT-kostnadene i hovudsak er knytt til kjøp av lisensar, og der kostnadene ved opplæring av brukarane på "ny" programvare er relativt liten.
- Linux er veileigna som tenaroperativsystem, også for offentleg sektor. Vurderinga vår er at open sluttbrukarprogramvare per i dag ikkje kan erstatte dei kommersielle alternativa når det gjeld verkemåte og høve for integrasjon.
- Det offentlege bør oppmuntrast til bruk av Linux og annan open programvare innafor skule- og utdanningssystemet.
- Det offentlege bør støtte utviklinga av open programvare. Støtta kan kome gjennom forskings- og utviklingstiltak. Ved tildeling av forskings- og utviklingsmidlar kan det vere eit krav at programvare som vert utvikla, vert gjort tilgjengeleg som open programvare.

2 Innleiing

2.1 Bakgrunn

Gjennom undersøkinga "IT i staten 1999"¹ fekk Statskonsult fram desse tala om den dominerande posisjonen Microsoft har som brukarmiljø i statleg forvaltning:

- 96 % nyttar MS Word som teksthandsamar
- 91 % av verksemdene hadde ein Windows NT tenar, og Windows NT utgjorde 60 % av alle tenarmaskinane
- Over 60 % av alle klientmaskinar nyttar ein eller annan versjon av Windows som operativsystem²
- Outlook og MS Mail utgjør 44 % av talet på e-postklientar³

Den avhengnaden statleg sektor, som tala i undersøkinga viser, har til ein leverandør samt dei stendig aukande kostnadene til IT generelt, gjer det naturleg å vurdere moglege alternativ til Windows/Office plattformar.

I den seinaste tida har det i pressa, fagmiljø og blant politikarar vorte fokusert på Linux som eit alternativ til Windows. Linux er eit opent, fritt tilgjengeleg operativsystem som er utvikla som "Open Source"-programvare, eller *open programvare*⁴ som vi vel å kalle konseptet i denne rapporten.

Bakgrunnen for at Statskonsult har utgreia open programvare er dei høva vi trur denne programvaren har i statlige sektor til å

- gjere avhengnaden av leverandørane mindre
- gjere kostnadene til kjøp av programvare mindre

For at forvaltninga skal ta i bruk open programvare, bør programvaren vere enkel å kjøpe, enkel å installere og enkel å drifte. Det må også vere eit programtilfang for operativsystemet, slik at ein til dømes kan nytte programvaren til sakshandsaming eller anna oppgåveløysing. Open programvare til sluttbrukarar, til dømes teksthandsamarar og e-postklientar, må også vere like "brukarvenleg" som proprietære⁵ produkt for å kunne verte teke i bruk i forvaltninga.

Statskonsult er ansvarleg for standardisering av datakommunikasjon med tilhøyrande infrastruktur innafor offentleg forvaltning. Vi har difor også vurdert korleis open programvare kan nyttast for å leggje høva betre til rette for felles standardar for data- og informasjonsutveksling innafor det offentlege og ut mot omverda.

¹ Statskonsultrapport 2000:8 IT i staten 1999

² IT i staten 1999 spurde ikkje direkte om kva slags klientoperativsystem som var nytta, men ut frå svara om bruk av tenaroperativsystem og talet på tilknytt PC-ar/terminalar har vi utleia minstetala

³ Dette var før Ft sin overgang til Outlook

⁴ Her bruker vi nemninga open programvare på den programvaren som vert omhandla i denne rapporten. Engelske nemningar omfattar mellom anna "Free Software" og "Open Source Software". Sjå elles kapitlet om "open programvare" for meir utførleg forklaring

⁵ I denne rapporten bruker vi nemninga proprietær programvare på all programvare som ikkje er open programvare

2.2 Leverandøruavhengig

Ein av årsakene til at Statskonsult utgreier open programvare er for å vurdere om ein større bruk av open programvare i offentleg sektor kan gjere avhengnaden av leverandørane mindre. I dette punktet beskriver vi minusa dersom avhengnaden av leverandørar er for stor.

Omgrepet leverandøruavhengig vert oppfatta som reell konkurranse. Det vil seie at vi som kjøparar kan velje mellom fleire leverandørar og produkt når vi skal kjøpe ein vare eller teneste. Avhengnaden av ein leverandør kan oppfattast som ein monopolsituasjon, eller mangel på konkurranse, og er noko som ikkje er ynskt. Heile innkjøpsregelverket for offentleg sektor er basert på prinsippet om fleire leverandørar, og at det er ein reell konkurranse mellom leverandørane.

Monopolsituasjonar kan føre til saktare produktutvikling, dårlegare tenester, høgre prisar, og at ein vert meir sårbar fordi berre eit bestemt produkt vert nytta. I NOU 2000:24 "Et sårbart samfunn" står det: "*Det har vist seg at noen leverandører har oppnådd tilnærmet monopol for utvikling av programvare. Ulempene ved slikt monopol ligger ikke bare i markedsmakten og økte kostnader i mangel av konkurranse, men også avhengighet og stort nedslagsfelt for negative konsekvenser dersom noe skulle gå galt.*"

Produktutvikling. Generelt trur ein at konkurranse fører til raskare produktutvikling og produkt som er betre tilpassa behovet som sluttbrukarane har. Innafor IT-området er det ein stor diskusjon om samansetting av produkt frå fleire leverandørar, eller om alle produkta frå same leverandør gjev den beste totale løysinga for brukarane. Vi trur at fleire leverandørar gjev brukaren større valfridom slik at det vert lettare å finne produkt og tenester som best er tilpassa han.

Tenester. På dette området er det særskild viktig med konkurranse, slik at brukaren får riktig teneste til ein fornuftig pris. Med monopol på tenester må brukarane berre ta det dei får. På dei fleste områda innafor tele og data har ein oppheva tidlegare monopol for å få betre konkurranse på tenester. Eit døme er opphevinga av telemonopolet, som har ført til at alle kan tilby teletenester, og ein markert nedgang i prisar på teletenester. Her trur vi at konkurranse på tenester er eit absolutt gode for brukarane.

Monokultur. Ved utstrakt bruk av eit bestemt produkt oppstår det vi kallar monokulturar. Monokulturar er svært utsett om det oppstår feil i produktet. Døme frå IT-verda på dette er feilane som har oppstått i Intels prosessorar i det siste, noko som har ført til at leverandørane av datamaskinar ikkje har kunne levere så mange maskinar som kundane vil ha. Intel har tilnærma monopol på tilverknaden av prosessorar til PC-ar.

Statskonsult meiner at open programvare er leverandøruavhengig. Årsakene til dette er mellom anna:

- Fleire leverandørar kan levere det same produktet fordi alle spesifikasjonar, all dokumentasjon og sjølve kildekoden er tilgjengeleg for alle.
- Det finnst ingen patent eller delar av programvaren som andre ikkje kan bruke.

2.3 Mål og gjennomføring

Målet med utgreiinga har vore å vurdere kor eigna ulike former for open programvare er for offentleg sektor.

I utgreiinga har vi spesielt lagt vekt på å vurdere moglege bruksområde for open programvare i forvaltninga, dei økonomiske konsekvensane ved å velje open programvare, og korleis open programvare kan skape mindre leverandøravhengnad.

Vi har testa ulike distribusjonar av operativsystemet Linux, e-posttenaren Sendmail og webtenaren Apache når det gjeld open programvare. Utgreiinga har i hovudsak fokusert på bruk av open programvare på tenarmaskinar, men vi vurderte også klientsida. Vi har også vurdert bruk av open programvare som eit standardiseringstiltak.

Det er innhenta erfaringar frå nokre sentrale miljø og brukarar av open programvare i offentleg sektor. Nokre døme på bruk av open programvare er omtalt i rapporten. Vi har også hatt kontakt med miljø som tilbyr ulike tenester innafor området open programvare.

Under arbeidet med utgreiinga har vi opna for og motteke ei rekkje førespurnader, kommentarar og bidrag frå eksterne aktørar. Så langt det har vore mogleg har vi teke omsyn til desse da vi utarbeida rapporten.

2.4 Om rapporten

I kapitla 3–5 forklårar vi kva open programvare er, dei ulike økonomiske modellar for open programvare, og dei mest utbreidde open programvareprodukt.

I kapittel 6 gjev vi døme på bruk av open programvare i offentleg sektor, og i kapittel 7 forklårar vi kort dei opne programvareprodukt og tenestene som vert tilbudd i Noreg i dag. I kapittel 8 tek vi spesielt for oss det høvet open programvare har som standardiseringstiltak, medan vi i kapittel 9 vurderer korleis vi kan bruke dei ulike opne programvarane. I kapittel 10 oppsummerar vi dei konklusjonane og tilrådingane vi har kome fram til.

3 Open programvare

INFORMATION, NO MATTER HOW EXPENSIVE TO CREATE, CAN BE REPLICATED AND SHARED AT LITTLE OR NO COST.

— *Thomas Jefferson*

Å dele og kopiere informasjon er enda enklare i dag enn då Thomas Jefferson levde. Det er verd å merke seg at (kildekoden til) programvare også er informasjon, og at sitatet like gjerne kan nyttast om programvare. Det er høvet til enkel spreining av informasjon, til store brukargrupper over lange avstandar⁶, som er årsaka til at konseptet open programvare har fått den tydinga det har.

3.1 Kva er open programvare?

Open kildekode programvare, eller open programvare som vi bruker i denne rapporten, er ei norsk omsetjing av omgrepet ”Open Source Software”.

Det er ikkje lett å forklåre med nokre få ord kva open programvare er. Årsaka til dette er at det finnst fleire variantar og definisjonar av konseptet. I dette punktet vil vi berre gje ei overordna beskriving, ikkje nokon definisjon, før vi i neste punkt beskriv kva vi forstår med open programvare gjennom å presentere hovudprinsippa i konseptet.

Open programvare er programvare der kildekode til programmet er fritt tilgjengeleg, og at programmet fritt kan nyttast, endrast, forbeholdt, utvidast og redistribuerast av alle som måtte ynskje det.

Det finnst klare definisjonar av både ”Open Source Software”⁷ og ”Free Software”⁸. Vi seier at open programvare er programvare som oppfyller krava vi skriv om i punkt 3.2.

3.2 Hovudprinsippa til open programvare

Open programvare kan karakteriserast gjennom den fridomen kvar einaste brukar⁹ har til å

- bruke programvaren slik han eller ho ynskjer, på så mange maskinar som er ynskjeleg, og i alle situasjonar det er ynskjeleg
- tilpasse programvaren slik at han tilfredsstillar dei spesielle behova han eller ho har. Dette inkluderar forbetringar i programvaren, feilretting, utvidingar av programvaren, og at det er mogleg å studere korleis programmet verker
- redistribuere programvaren til andre brukarar, som atter kan tilpasse programvaren i høvet til sine egne behov

⁶ Dette kan lesast som laglegskap av Internett

⁷ The Open Source Initiatives definisjon av Open Source Software finns på <http://www.opensource.org/osd.html>

⁸ Free Software Foundation definisjon av Free Software finns på <http://www.fsf.org/philosophy/free-sw.html>

⁹ Når ein i open programvaremiljø snakkar om brukarar, tenkjer ein på ein kompetent programvareutviklar og ikkje ein vanleg ”pek-og-klikk”-sluttbrukar. Brukarperspektivet i miljøet er svært utviklarorientert

For å kunne oppfylle vilkåra som er nemnt over, er det enda eit vilkår som må tilfredsstillast:

- Brukaren av eit program må ha tilgang til kildekoden til programmet.

Kildekoden til eit program, som ofte er skriva i eit høgnivå programmeringsspråk, som til dømes Java, C eller C++, er ein absolutt føresetnad for at ein brukar skal kunne forstå korleis programmet er bygd opp og verker, for så å kunne tilpasse, utvide og rette opp feil i programmet.

For å kunne oppretthalde fridomane vi har skriva om i dette punktet, er det naudsynt å verne programvaren med lisensvilkår og/eller copyright. Dette kan virke noko paradoksalt, men er programvaren utan lisensvilkår, så kalla public domain, kan folk til dømes ta eigen copyright på programvaren, redistribuere sjølve programmet utan kildekoden, eller bruke kildekoden som basis for proprietær programvare. Ulike opne programvarelisensar vert omtalt i punkt 3.4.

3.3 Forskjellen mellom open programvare og andre programvarar

Open programvare er ikkje det same som gratis programvare. Gratis programvare eller "freeware" er nemninga på kompilerte¹⁰ program som vert distribuert fritt. Det finnst mykje gratis programvare, alt frå små hjelpeprogram til store kjende produkt som Microsofts Internet Explorer.

Kommersiell programvare som vert selt utan tilgang til kildekoden, og at det er mogleg å endre og redistribuere han, kallast proprietær programvare. Det er spesielt lisensvilkår til proprietær programvare som gjer at folk utviklar opne alternativ. Dei fleste lisensar har avgrensingar på kor mange maskinar programmet kan installerast på, kor mykje ein kan endre i programmet, og om ein kan distribuere programmet vidare.

Lisensvilkåra til proprietær programvare inneber at ein kjøper ein rett til å bruke programmet på visse vilkår, ikkje sjølve programmet.

3.4 Kort om opne programvarelisensar

Lisensvilkåra knytt til programvare er eit omfattande tema. Her gjev vi døme på innhaldet i opne programvarelisensar og forklarar kvifor det er knytt lisensvilkår til open programvare.

Den fyrste lisensen knytt til open programvare vart utvikla av Free Software Foundation i samband med deira GNU^{11 12} programvare. Ei forklaring av prinsippa finnst på <http://www.gnu.org/copyleft/copyleft/htlm>. Lisensen til GNU programvaren heiter GNU General Public License og vert berre kalla for GPL Heile

¹⁰ Eit kompilert program kan lesast av maskinvaren som skal bruke programmet, og ikkje av menneske. Eit program som er skriva i eit språk som menneske forstår, vert kompilert til eit språk som maskinvaren forstår. Når ein berre distribuerer det kompilerte programmet, er det berre maskinvaren som kan lese og forstå programmet

¹¹ GNU er ei forkorting av det sirkulære uttrykket Gnus Not Unix

¹² Originalteksten finns på <http://www.gnu.org/copyleft/gpl.html>

lista finnast som eit vedlegg til denne rapporten. GPL er fundamentet for dei fleste opne programvarelisensar. The Opens Source Initiative har lagd ei forklaring av kva som skal til for å kalle ein lisens ein open programvarelisens [3].

To døme på opne programvarelisensar vert presentert i vedlegg til denne rapporten; GPL (vedlegg III) og BSD (vedlegg IV).

Ei av hovudårsakene til at det er knytt lisensar til open programvare, er for å verne om programvaren slik at han framleis skal vere open. Som nemnt tidlegare kan open programvare verte gjort om til proprietær programvare dersom det ikkje er knytt lisensvilkår til programvaren. Lisensvilkåra kan mellom anna

- ivareta fridomane som er nemnt i punkt 3.2
- ivareta vilkåra som dei opphavlege rettshavarane ynskjer (at namn på opphavlege utviklarar følgjer med, at lisensen vert distribuert med programvaren osv)
- passe på at rettskrava til dei som har utvikla programvaren, eller har copyright på namnet til programmet, vert ivareteke
- påleggje at ny programvare som er basert på den opne programvaren, framleis vert open programvare
- påleggje at all programvare som nyttar den opne programvaren, også skal vere open programvare

Dei ulike lisensane varierer mest når det gjeld dei to siste kulepunkta. Medan GPL er svært streng når det gjeld at all programvare som er basert på den opphavlege koden, også skal vere fritt tilgjengeleg, tillet BSD-liknande lisensar at ein kan lage proprietær programvare basert på den opphavlege kildekoden.

Opne programvarelisensar er ikkje til hinder for at verksemdene sel programvaren og tener pengar på dette. Dei som sel open programvare, kan derimot ikkje hindre andre i å selje den same programvaren. Gjennom lisensvilkåra kan ein ikkje avgrense den einskildes rett til å kopiere eller endre programvaren, og ein må også la kildekoden følgje med dei programma ein sel. GPL seier at endringar og forbetringar i koden også skal vere open kildekode, og at alle kan bruke dette i eigen verksemd. Eit godt døme på dette er installasjonsprogrammet RPM (RedHat Packet Manager) som er utvikla av Linux-distributøren RedHat. Dette programmet er også ein del av distribusjonen til Caldera og SuSE. Dermed vert forbetringar gjort av ein distributør som vert nytta av dei andre distributørane.

Det skal tydeleg gå fram på opne programvarepakker som vert distribuert, at distribusjonen er omfatta av ein open programvarelisens. Alle kan setje saman pakker med open programvare og distribuere desse.

3.5 Døme på open programvare

Det finnast fleire opne programvarar som er tilgjengelege. Mange av dei mest nytta programma på Internett er open programvare. Programma vert ofte utvikla i opne prosjekt. Prosjekt betyr her ein liten kjerne med utviklarar som publiserar nye versjonar av programmet, og som koordinerar og bruker innsatsen til ei rekkje friviljuge bidragsytarar for å vidareutvikle programmet. Vi vil no kort beskrive nokre av programma.

3.5.1 Linux

Linux er det programmet som får folk utafor open programvarekulturen til å verte merksam på open programvare, og er det programmet som ofte vert trekt fram når ein skal forklåre kva open programvare er.

Linux, eller meir korrekt GNU/Linux, er eit UNIX-liknande operativsystem. Linux prøver å vere i samsvar med POSIX-standarden for å vere mest mogleg kompatibel med andre UNIX-system¹³. Eit program utvikla for eit POSIX-operativsystem, skal i teorien kunne nyttast på alle operativsystem som følgjer POSIX-standarden.

Opphavleg var Linux designa for å kunne kjørast på Intel-baserte PC-ar, men finnast nå for andre maskinvareplattformer som Macintosh, Alpha¹⁴ og Sun SPARC.

Linux er utvikla og vert distribuert under GPL. Dette betyr at Linux kan distribuerast fritt så lenge kildekoden følgjer med. Alle endringar og modifikasjonar av Linux som det er ynskjeleg å distribuere, må også innehalde kildekode for endringane. På denne måten sørgjer GPL for at Linux framleis kan vere open, og at ingen kan lage ein proprietær versjon av operativsystemet. Sjå elles kapittel 4 for meir informasjon om Linux.

3.5.2 Apache

Apache-prosjektet er ein friviljug felles innsats på programvareutvikling med sikte på å etablere ein robust kildekode for http-tenar (her kallar vi han noko unøyaktig ein webtenar), rik på funksjonar og fritt tilgjengeleg, med eigenskapar som er tilstrekkelege for bruk i næringslivet.

I følgje undersøkingar gjord av Netcraft er Apache-tenaren den mest nytta webtenaren på Internett med om lag 60 prosent av marknaden. Det er over 5 millionar Apache webtenarar på Internett i dag¹⁵. For ei full og oppdatert oversikt, sjå <http://www.netcraft.com/survey/>.

Apache kan nyttast på ei rekkje plattformer og med mange operativsystem. Blant dei mest nytta operativsystema for Apache webtenar finn vi FreeBSD, Sun Solaris og Linux. Apache prøver å vere ei referanseimplementering for http-standarden for alle plattformer som han vert utvikla for.

Brukarane av Apache er mange. I Noreg nyttar mellom anna Aetat, Datatilsynet, SSB, Statskonsult, Dagbladet, VG, Startside, Sol, Stocknet og Telenor¹⁶ Apache til sine eksterne webtenester.

3.5.3 Bind

BIND (Berkeley Internet Naming Daemon) er ei implementering av DNS-protokollen (Domain Name System) og inneheld ei referanseimplementering av dei viktigaste komponentane i DNS som ein fritt kan distribuera, mellom anna ein

¹³ POSIX er utgjeve som "IEEE Standard 1003.1-1988 Portable Operating System Interfaces for Computer Environments og som ISO/IEC 9945.1:1990

¹⁴ Tidlegare var dette DEC Alpha, men er nå Compaq Alpha

¹⁵ Dette er Netcraft sine tal for oktober 2000

¹⁶ Opplysningane vart framskaffa gjennom bruk av Netcraft sine tenester i oktober 2000

namntenar (nameserver), eit DNS-bibliotek for å omsetje mellom namn og IP-adresser, og verkty for å verifisere at ein DNS-tenar fungerer i forhold til standarden.

BIND er den dominerande namnetenaren på Internett og er ein føresetnad for at ein skal kunne bruke namn, som til dømes www.statskonsult.dep.no, for å finne fram til IP-adressa til ein maskin på Internett.

3.5.4 Perl

Perl er eit programmeringsspråk som har spesielt gode eigenskapar når det gjeld tekstmanipulering, og er difor spesielt eigna til å lage interaktive og dynamiske websider. Dei fleste kommersielle websider som tilbyr dialog med brukarane og tilgang til databasetenarar, er utvikla ved bruk av Perl. Eit konkret døme er Amazon.com.

Perl kan integrerast i webtenarar, noko som fører til store forbetringar i prosesseringssnøggleiken. Kort sagt kan bruken av Perl gjere kommunikasjonen med system som ligg attom, enklare og raskare.

3.5.5 Sendmail

Sendmail er den mest nytta tenaren for Internett e-post. Om lag 75 prosent av alle e-posttenarar på Internett er ulike Sendmail-implementeringar. Sendmail er fritt tilgjengeleg, og han kan også fritt distribuerast vidare.

Sendmail har i hovudsak tre typar tenestar: e-post ruting (inkludert ulike former for autentiserings- og krypteringsmekanismer), lagring av e-post (inkludert støtte for POP og IMAP), og høve for utviding av e-postprogrammet (utvidingar som leveringskvitteringar og integrasjon mot ein PKI).

Sendmail følgjer og vert utvikla i samband med eksisterande e-poststandardar innafor IETF. Dei siste versjonane av Sendmail har støtte for mellom anna ESMTP [RFC 1869].

3.6 Utviklingsmetodikken for open programvare

Utviklingsmetodikken for open programvare er svært forskjellig frå utviklinga av proprietær programvare. Boka "The Cathedral & the Bazaar" [2] forklår utviklingsmetodikken for, drivkraftane bak, og kulturen innafor open programvare. Det er viktig å vite korleis open programvare vert til for å kunne forstå den omgrepsmessige forskjellen mellom open og proprietær programvaren.

Dei fleste opne programvareprosjekta startar med at ein eller fleire personar føler at ein programvare manglar, eller at ein eksisterande programvare er utilstrekkeleg eller dårleg. Dei startar då med å lage ny programvare, gjerne basert på programvare som allereie finnast der kildekode er tilgjengeleg. Når tilstrekkelege delar av programmet verker, ikkje når det er ferdig, legg dei det ut på ei nyheitsgruppe eller liknande, slik at andre kan bruke programmet og eventuelt kome med kommentarar og forbetringar. Er det mange nok som har interesse av programmet og kjem med kommentarar, er eit nytt opent programvareprosjekt fødd.

Alle opne programvareprosjekt har ein eigar. Eigaren er han som mottok forslag til endringar i programmet, forslag til ny kode, rettingar, forbetringar, og som gjev ut nye versjonar av programmet. Dette er ein kontinuerleg prosess, og i starten av eit prosjekt kan det kome nye versjonar av eit program så ofte som kvar einaste dag.

All kildekode vert publisert, og det er fritt for alle å bruke den publiserte kildekoden til eksisterande eller nye opne programvareprosjekt.

Eigarens oppgåve vert å lage ein oversiktleg og god struktur i programmet, administrere alle endringar og ny kode som vert foreslått, og ikkje minst lage nye versjonar av programmet som har med dei endringane og koden eigaren meiner er vettig. Dette betyr at ikkje alle kodar kjem med, eller at alle forslag til nye funksjonar vert implementert. Alle program har ei read-me-fil som fortel om kven som har gjeve bistand med kva til programmet. Det er viktig å kreditere bidragsytarane.

Det er viktig for eit prosjekt å få så mange brukarar av programmet som mogleg, for derigjennom å få så mange bidragsytarar som mogleg til vidareutviklinga av programmet.

3.7 Kva for programvare er mest eigna for utvikling som open programvare?

Kva kan utviklast som open programvare, og kva har vorte utvikla som open programvare? Dette er viktige spørsmål, og svaret vil gje klare føringar på kva open programvare er eigna til.

Spekteret for programvare går frå infrastrukturprogramvare på den eine sida til sluttbrukarprogramvare på den andre sida. Infrastrukturprogramvare kan vere alt frå iverksetjingar av ulike tenarprogramvarar, operativsystemkjerner til DNS og TCP/IP. Sluttbrukarprogramvare er i denne samanheng programvare som den vanlege ikkje-tekniske brukaren nyttar som teksthandsamarar, rekneark, rekneskapsprogram og ulike verksemdspesifikke system.

Tradisjonelt har utviklinga av open programvare skjedd på infrastrukturensida av programvarespekteret. Bind, Sendmail og Linux er alle gode døme. Andre program som Apache og Perl ligg litt meir mot midten, men kan ikkje kallast ikkje-teknisk sluttbrukarprogramvare.

Det ein kan sjå, er at open programvare ofte er programvare som andre program skal køyrast på toppen av, eller verktøy for å utvikle andre program. I følgje Brian Behlendorf, hovudutviklar av Apache, er det fleire årsaker til at open programvare har vorte utvikla på infrastruktur og på tenarsida av programvarespekteret. Nokre av desse er:

- Sluttbrukarapplikasjonar er vanskelege å lage. Det er ikkje berre fordi ein programmerar må halde seg til stendig skiftande ikkje-standardisert vindaugutviklingsmiljø, men også fordi dei fleste utviklarane ikkje er gode grafiske designarar på grensesnitt. Dette krev testlaboratorium, brukarstudie osv, noko som open programvaremiljø ikkje plar ha tilgang til.

-
- Kulturelt sett har open programvare vore fokusert på nettverksprogram og operativsystem.
 - Utviklinga av open programvare trivs best der inkrementell utvikling er mogleg. Historisk har dette vore enklare på tenarsida enn på klientsida, fordi struktur og arkitektur på programvaren er betre kjend her, og det er lettare å få eit grunnleggjande operativt program som ein seinare kan lage utvidingar av.
 - Mykje av den opne programvaren vart skriva av teknikarar for å løyse eit problem som dei hadde, medan dei utvikla annan programvare. Målgruppa for programvaren var andre teknikarar.

For å utvikle eit program treng ein god kjennskap til problemet, eller ein god kravspesifikasjon. Tradisjonelt har ikkje open programvaremiljø hatt kjennskap til kva dei ulike ikkje-tekniske sluttbrukarane har trunge, og dei har heller ikkje hatt tilgang til gode kravspesifikasjonar. Måten open programvareprosjekt startar på, og programma vert utvikla på, er heller ikkje så godt eigna for brukarmedverknad frå personar som ikkje medverkar med kode. Men i ei utviklingsgruppe for proprietær programvare er det derimot vanleg å ha med brukarar.

Det sitatet som kjem no er meint å illustrere kor naudsynt det er å ha kunnskap om eit område for å kunne utvikle gode program for det. Sitatet er av Larry Augustin, administrerande direktør (CEO) i VA Linux Systems¹⁷:

”Open-source developers understand UNIX. This is part of what made it possible to create a better UNIX-Linux. In order to create a better MS Office, open-source developers need to understand MS Office in as much detail as they understood UNIX. My fear is that the open-source developer community doesn’t understand Office. It can’t create what it doesn’t understand. What we need is more developers using windows and Office.”[5]

”*Dei kan ikkje lage noko dei ikkje forstår*”. Dette er den sentrale budskapet. Skal dei som utviklar open programvare til dømes lage eit rekneskapssystem, må dei kunne rekneskap, alt regelverk og alle rutinar som er knytt til dette, samt vere gode programmerarar.

3.8 Tenkjelege fordelar med open programvare

Motivasjonen for å bruke og utvikle open programvare er forskjellig. Alt frå filosofiske og etiske til reint praktiske årsaker er ofte representert. I dette punktet presenterar vi nokre av dei tenkjelege fordelane ved bruk av open programvare.

Den fyrste fordelan med open programvare er at den er gratis og fritt tilgjengeleg. Denne fordelan er ikkje unik for open programvare. Som vi har skive i punkt 3.3 finnast det mykje proprietær programvare gratis tilgjengeleg. Det som skil open programvare frå annan programvare, og gjev dei tenkjelege fordelane, er korleis den einskilde kan utnytte hovudprinsippa som ligg bak utviklinga av open programvare. Nedanfor omtalar vi desse:

¹⁷ VA Linux Systems sel maskinar med ferdig installerte Linux-system som er tilpassa dei spesielle behova kundane har. Firmaet tilbyr også “alle typar konsulenttenester i samband med Linux

-
- *Tilgjengeleg av kildekoden og retten til å tilpasse han.* Tilgjengeleg av kildekoden gjev det mogleg å tilpasse og forbetre programmet for eigen bruk. Det gjer det også mogleg å lage versjonar for nye maskinvareplattformer og få detaljert kjennskap til korleis programmet fungerer. Treng ein rettingar i programmet, kan ein gjere dette sjølv berre ein har kunnskapen. Lokale tilpassingar vert mogleg, til dømes å få programmet på sitt eige språk om leverandørane ikkje lagar ein slik versjon.
 - *Høva til å redistribuere rettingar og modifikasjonar.* Det er mogleg å lage nye program basert på eksisterande kode for sida å distribuere dette programmet fritt. Trur ein at det er feil og manglar i programmet, kan ein rette desse og sida redistribuere den modifiserte versjonen. Viser det seg at den nye versjonen er betre, vil denne som regel verte teke inn i det opphavlege programmet, og det opphavlege programmet vert betre.
 - *Fridomen til å bruke programmet på den måten det passar den einskilde.* Mange gratis program har avgrensingar på korleis programmet kan nyttast. Eit vanleg døme er at programmet ikkje kan nyttast for kommersiell bruk. Slike restriksjonar finn ein ikkje under programvare som vert distribuert under GPL. Ynskjer ein å ta i bruk programmet på nye tekniske plattformer, i "embedded systems", eller anna, er dette heilt i orden.
 - *Det er ingen ukjente "svarte boksar" i programmet.* I proprietær programvare kan delar av funksjonsevna vere ukjent. Det kan vere verknader i programmet som ikkje er dokumentert for omverda som dei som har lagd programmet kan utnytte. Dette er ofte gjort som udokumenterte systemkall.
 - *Utviklinga av programvaren er ikkje avhengig av berre ein verksemd.* Om utviklaren av programmet ikkje ynskjer å lage nye versjonar, eller ikkje ynskjer å vidareutvikle for plattformen di, er det mogleg å få andre firma til å overta eller du kan overta sjølv. I tilfellet Linux er det mange leverandørar av det same produktet som på sin eigen måte vidareutviklar det, og tek det beste frå det dei andre gjer. Utviklinga og vedlikehaldet av programmet er ikkje avhengig av eitt firma, og ein minskar dermed leverandøravhengnaden.

3.9 Tenkjelege ulemper med open programvare

Nedanfor nemner vi nokre tenkjelege ulemper knytt til open programvare:

- *Det finnast ingen garanti for at det vil skje noka utvikling av ein spesiell programvare eller ei vidareutvikling av programvaren i ynskja retning.* Anna programvareprosjekt treng menneskelege eller økonomiske ressursar for å kome i gang. Ein veit aldri om produktet vil kome så langt at det kan nyttast, eller om det er nok interesse for produktet. Dette gjeld alle typar programvareutvikling, men utryggleiken er vesentleg større når det gjeld open programvareprosjekt.

I alle programvareprosjekt er det ein periode frå ideen vert utforma og arbeidet startar til det føreligg noko konkret som kan nyttast. Dette er den

skjøraste fasen i eit opent programvareprosjekt. Særleg dersom eit opent programvareprosjekt vert starta utan sterk støtte frå ein eller fleire verksemder, kan det vere vanskeleg å få koden opp på eit slikt nivå at han trekkjer til seg andre programmerarar som kan bruke og utvikle produktet vidare. Er det ikkje mogleg å få nok støtte eller nok programmerarar interessert, vil prosjektet berre dø eller sakte forsvinne.

- *Det er av og til vanskeleg å vite om eit spesielt opent programvareprosjekt eksisterar, og finne ut kor langt det har kome.* Det skjer lita marknadsføring av open programvare, difor er det vanskeleg å finne ut om det finnast open programvare som dekkjer behova dine.
- *Problem knytt til patentrettar og intellektuelle rettar.* Opphavsrettar, eller med andre ord brot på opphavsrettar, kan vere eit stort problem for open programvare. Det er vanskeleg å vite om ein bestemt algoritme eller måte å løyse eit problem på, er patentert. Patent på programvare er eit problem for alle som utviklar programvare, men dei rammer open programvare meir enn annan programvare fordi tilgjenge av kildekode gjer det lettare å oppdage brot på opphavsrettar. Dei fleste opne programvareprosjekt har ikkje finansielle krefter til å skaffe seg rettsleg hjelp i tvistar som gjeld brot på opphavsrett.

4 Linux

Som vi har skrivt i punkt 3.5.1 er Linux, eller meir korrekt GNU/Linux, eit UNIX-liknande operativsystem. Linux kjem i fleire distribusjonar frå ulike leverandørar og er også fritt tilgjengeleg på Internett.

I dette kapitlet skriv vi meir detaljert om Linux, om ulike distribusjonar av Linux, og om programvare som er tilgjengeleg for Linux. Kapitlet vil til dels vere teknisk orientert, og ein føresetnad er difor at lesaren har litt teknisk kunnskap.

4.1 Kva er ein Linux-distribusjon?

Ein Linux-distribusjon er ei samansetjing av mange ulike programvaremodular, kalla pakker, som til saman utgjer eit komplett operativsystem. Det vi får når vi kjøper ein distribusjon frå eitt av selskapa som distribuerar Linux, er i realiteten ei rekkje pakker som dei har satt saman for oss. RedHat Linux består av over 400 ulike pakker frå mange forskjellige utviklingsprosjekt.

Ein Linux-distribusjon er meir enn berre eit operativsystem. Det er ei samling av opne programvarekomponentar som gjer at brukarane kan utnytte operativsystemet betre og meir lagom. Blant komponentar som følgjer med er grafiske brukargrensesnitt, kompilatorar, teksthandsamarar og ulik tenarprogramvare. Alt dette er open programvare. I tillegg er det ofte med proprietær nytteprogramvare som kontorstøttepakker og spill.

Pakking av Linux saman med dei installasjonsprogramma som ofte følgjer med, er ein hovudgrunn til at ein kjøper ein distribusjon og ikkje lastar ned og set det saman sjølv.

4.2 Linux-kjernen

Den sentrale komponenten i Linux er kjernen. Skal ein vere presis, så er det kjernen som er Linux. Kjernen er det systemet som kontrollerer kommunikasjonen (eller grensesnittet) mellom andre program og maskinvaren.

Kjernen i eit operativsystem sørgjer mellom anna for filhandtering, I/O, signalering, separering av prosessar, mottak av avbrotmeldingar frå dei ulike maskinvaredelane, og kommunikasjonen mellom andre delar av operativsystemet og maskinvaren.

4.3 Designmål for Linux¹⁸

Linux vart utvikla for Intel 386-plattformen og var i utgangspunktet ikkje designa for å vere transportabelt. Linux-kjernen er ikkje skriven for å vere transportabelt, men vart designa for å vere effektivt.

Linux-kjernen vart til dels utvikla ved å sjå på fellestrekk ved eksisterande maskinvarearkitektur, og designe operativsystemet ut frå kva som var felles for desse arkitekturane. Kjernen er kva ein kallar ein monolittisk kjerne, og ikkje ein mikrokjerne, som var hovudtrenden da Linux vart utvikla.

¹⁸ Teksten i dette punktet er eit omarbeida utdrag av eit essay Linus Torvalds har i boka Open Sources

Ved vidareutviklinga av Linux-kjernen har Linus Torvalds lagt til grunn nokre få prinsipp:

- Utgangspunktet er at ein skal bruke sunne generelle designprinsipp i staden for å designe for transport. Bruken av god effektiv design og kode vil også føre til at systemet vert portabelt.
- Ein skal så langt det er mogleg unngå grensesnitt som kan programmerast til kjernen. Årsaka er at med ein gong ein får eit grensesnitt inn mot kjernen, må dette oppretthaldast i nye versjonar fordi folk begynner å programmere mot dette.
- Ein skal vere varsam med å tilføye nye eigenskapar til systemet. Nye eigenskapar må nøye vurderast før dei kan takast inn i kjernen, sida kjernen påverkar resten av systemet.
- Kjernen må lagast så modulær som mogleg av fleire årsaker. Ei årsak er at modulære system er lettare å tilpasse spesifikke maskinvarearkitekturar sidan ein ikkje treng å endre heile systemet. Ei anna årsak er at folk kan arbeide uavhengig av kvarandre på forskjellige delar av kjernen.

4.4 Grafisk brukargrensesnitt

Linux er originalt utan eit grafisk brukargrensesnitt, men det finnast fleire grafiske brukargrensesnitt som kan brukast saman med kjernen for å utgjere eit komplett moderne operativsystem. Linux har over ti ulike grafiske brukargrensesnitt. Desse vert kalla vindauehandteringssystem (Windows managers).

Dei grafiske brukargrensesnitta køyrar på toppen av Xfree86, ei fri utgåve av X Windows System. X-systemet er infrastrukturen for alle Unix-liknande vindauehandteringssystem. X-systemet er svært enkelt å konfigurere, og i stor grad kan ein sjølv avgjerde korleis brukargrensesnittet skal framstå. Det er mogleg å få eit X-system til å etterlikne både Windows 95/98 og Macintosh.

Dei to mest populære brukargrensesnitta for Linux for tida er KDE (the K Desktop Enviroment) og GNOME (the GNU Network Object Modell Enviroment). Alle Linux-distribusjonar kjem med eitt eller fleire grafiske brukargrensesnitt.

4.5 Kompilatorar

Får å kunne tilpasse kildekoden og lage ein eigen versjon treng ein ein kompilator som omsetjar frå kildekode til maskinkode. Jo betre kompilatoren er, jo lettare er det å lage tilpassa versjonar, eller versjonar for andre maskinvareplattformer. Linux er utvikla i programmeringsspråket C. Det følgjer med kompilatorar i alle distribusjonar av Linux. Den vanligaste kompilatoren er gcc (GNU C Compiler).

4.6 Programvareutvikling

Linux er ei utmerkt plattform for programvareutvikling. Linux-distribusjonar kjem som standard med ein C og C++-kompilator og assembler. I tillegg finnast det utviklingsverktøy for nyare språk som Perl og Python. Årsaka til at det vert lagd så stor vekt på programvare for programvareutvikling, er at høvet til å tilpasse

programvare er eitt av dei store føremona ved Linux. Utan gode programutviklingsverktøy ville tilgjengen av kildekode og høvet til å endre koden hatt liten praktisk nytte.

4.7 Lisensvilkåra for Linux

Linux er utvikla, og vert distribuert, under lisensen GNI General Public License (GPL). Se elles punkt 3.4.

Linux er ein open programvare, men alle som ynskjer det, kan setje saman ein distribusjon og selje han. Ein føresetnad er sjølvstendig at lisensvilkåra i GPL vert overhalde. Linux-distributørar kan ikkje avgrense vilkåra i GPL eller vidaredistribusjon. Skal du installere Linux på fleire maskinar, kan du berre kjøpe ei pakke (CD) som kan nyttast for å installere Linux på alle maskinane. Men det er eitt unntak frå denne regelen. For å skape meirverde for sin eigen distribusjon, legg nokre leverandørar med kommersiell programvare i pakka. Den kommersielle programvaren kan ha avgrensingar på kor mange maskinar dei kan installerast på. Dette bør kome klårt fram i sjølve pakka som vert distribuert.

4.8 Ulike distribusjonar

Vi har sett på fire ulike Linux-distribusjonar og vurdert ulike eigenskapar ved distribusjonane. Distribusjonane er forskjellige både når det gjeld pris og programvaren som følgjer med. Målet vårt har ikkje vore å kåre ein vinnar, men å sjå på om dei ulike distribusjonane yter noko ekstra til Linux.

Nedanfor er ei oppsummering av eigenskapar til nokre distribusjonar:

Distribusjon/ eigenskapar	Caldera	Corel	RedHat	SuSE
Pris	Kr 345	\$ 89 + \$ 63 i frakt = kr 1260	\$ 80 + \$ 38 i frakt = kr 977	Kr 345
Type	Caldera Open Linux 2.3 Standard	Corel Linux OS Deluxe	RedHat Linux 6.1 Deluxe	SuSE Linux 6.4.
Kjøp	Nett – via Akademika	Nett – direkte frå Corel	Nett – direkte frå RedHat	Nett – via Akademika
Tid for levering	2 dagar	7 dagar	5 dagar	2 dagar
Fyrste gangs installasjon	Partition Magic, Lizard	Corel Install Express	Linuxconf	YAST 2
Installasjon av nye pakker	COAS	Corel package installer	RPM	RPM, SaX
Tenarprogramvare som følgjer med	Apache, Samba, Sendmail, DB2	Apache, Samba	Apache, Samba, Sendmail, MySQL	Apache, Samba, Sendmail, MySQL
Kontorstøtteprogram som følgjer med	Star Office, Applixware, WordPerfect (prove)	WordPerfect	Star Office	Star Office, Applixware, WordPerfect (prove)
Drift og administrasjon	COAS			
Brukarstøtte inkludert	90 dagar e-post	30 dagar e-post og tlf.	90 dagar web, 30 dagar tlf.	60 dagar e-post og tlf.
Anna brukarstøtte tilgjengeleg	Telefon mot ein avgift			

5 Økonomien ved open programvare

I dette kapitlet forklårar vi både korleis ein kan tene pengar på open programvare, korleis det offentlege kan finansiere utviklinga av open programvare, og om bruken av open programvare kan redusere dei totale IT-kostnadene.

5.1 Forretningsmodellar for open programvare

Eit interessant spørsmål er korleis det er mogleg å tene pengar på open programvare. Programvaren er fritt tilgjengeleg og kan lastas ned gratis. Dermed må ein tene pengar på ein annan måte enn å selje den opne programvaren. I dette punktet skriv vi om ulike forretningsmodellar for å tene pengar på open programvare.

5.1.1 Skape eit merkenamn og distribuere ei pakke av open programvare

Dette er den mest synlege måten å tene pengar på. Bedrifter som RedHat og SuSE set saman distribusjonar av Linux og sel desse pakkene saman med avgrensa brukarstøtte og trykte brukarrettleiingar (bøker). Brukaren betaler ikkje for sjølve programvaren, men for samanstillinga av dei ulike programma, brukarstøtte og for pakkinga og distribusjonen. Kjøparen held seg til eit merkenamn og kjøper eit produkt frå dette firmaet i staden for å kjøpe eit liknande produkt utan merkenamnet.

5.1.2 Selje open programvare som ein sjølv er hovudutviklar av

Som for punktet over er det eit slags merkenamn som sjølv er produktet. Som eit tillegg her er firmaet som sel den opne programvaren, også hovudutviklar av programvaren. Dermed er det dette firmaet som har den beste kompetansen på produktet, og det har ein stor fordel når det skal selje tenester som brukarstøtte og spesialtilpassingar, sida firmaet kjenner kildekoden best. Ei ulempe her er at ein bruker mykje ressursar på å utvikle sjølve programvaren, utan å få direkte tilbakebetalt desse kostnadene. Poenget er å ha betre kunnskap enn det konkurrentane har, og alltid vere nokre månader før når det gjeld nye finessar i koden. Cygnus, hovudutviklaren av gcc- (Gnu C Compiler)kompilatoren, er eit godt døme på ei verksemd som lever av å ha betre kompetanse enn konkurrentane sine.

5.1.3 Proprietære produkt som aukar verdi av den opne programvaren

Mange verksemder lager både open og proprietær programvare som er basert på den opne programvaren. Han er meint for forbrukarmarknaden, mens den proprietære programvaren er meint for forretningsmarknaden. Eit døme på dette er Sendmail, som har ei proprietær utgåve der nye funksjonar kjem 4–8 månader før den same funksjonen i den opne utgåva.

5.1.4 Konsulenttenester

Dette er ei tradisjonell verksemd; gje bort produktet og selje tenester som opplæring, drift og vedlikehald. Distributørane av Linux baserer seg på at dei skal få ein stor del av inntektene sine på sal av tenester. Dette er også eitt av dei viktigaste argumenta som vert brukt for open programvare. IT-selskap må fokusere på å tilby gode

tenester til brukarane når dei ikkje kan basere eksistensen sin på inntekter frå lisensar, og tenestene til brukarane vil dermed verte betre¹⁹.

5.1.5 Selje utstyr som høyrer til

Dette kan verke trivielt, men sal av ”utstyr som høyrer til” er viktig for marknaden rundt open programvare. Forlaget O'Reilly støttar utviklinga av open programvare ved å løne sentrale utviklarar av open programvare. Forlaget tener inn dette på sal av bøker om Linux, Perl og anna det utgjev. Andre i denne kategorien er dei som sel maskinvare med open programvare preinstallert, eller som sel kommersiell programvare for Linux. Som ein kuriositet kan vi nemne at det er ein stor marknad for ulike former for maskotar knytt til Linux.

5.2 Finansiering av open programvareutvikling

Dette punktet handlar om korleis ein kan finansiere utviklinga av open programvare. Det vil vere eit grunnlag for dei forslaga rapporten framsetjar på korleis det offentlege kan støtte opp om utviklinga av open programvare, sjå kapittel 10. Dette punktet er i hovudsak basert på rapporten om open programvare, som er sponsa av EU [4].

5.2.1 Eksternt finansiert utvikling

Her vert utviklinga av open programvare betalt av ei ekstern verksemd. Det kan vere fleire årsaker til at somme er viljuge til å finansiere utvikling av open programvare. Fleire av bidragsytarane er verksemder som lever av open programvare på ein eller annan måte, sjå punkt 5.1.

Ein viktig bidragsytar er det offentlege, fordi det gjennom midlar til forskning og utvikling kan sponse utviklinga av ny programvare for spesielle bruksområde. I Noreg vil dette typisk vere Noregs forskingsråd og Statens nærings- og distriktsutviklingsfond.

Andre bidragsytarar er selskap som sel tilhøyrande produkt, og som vil tene på utviklinga av ein spesiell open programvare. På Linux finnast det ein del proprietær programvare (spesielt programvare for kontorstøtte) som har behov for drivarar eller annan programvare, for å virke korrekt på Linux. Desse betaler gjerne for utviklinga av denne opne programvaren.

5.2.2 Internt finansiert utvikling

Verksemdar som sel open programvare eller tenester direkte knytt til open programvare, vil støtte utviklinga av open programvare for å auke marknaden for slike produkt. Er dei gode, vil ein større marknad bety større høve for å tene pengar. Alle dei store distributørane av Linux gjer det slik. Ved sjølv å betale for utviklinga av produktet vonar ein å skape ein større marknad og der igjennom større høve for inntening.

¹⁹ Vi har til nå ikkje sett dokumentasjon på at tenestene til brukarane er vorte betre eller dårlegare fordi brukaren nyttar open programvare

5.3 Tydinga som open programvare har for dei totale IT-kostnadene

Har open programvare noka vesentleg tyding for dei totale IT-kostnadene i ei verksemd? Dette spørsmålet er ei av årsakene til at Statskonsult har vurdert open programvare. Det er ikkje mogleg å svare generelt på dette, for alle verksemder har ulike føresetnader ved kjøp av nytt IT-utstyr. Det vi vil gje er ei generell forklåring på korleis ein kan vurdere sine eigne totale IT-kostnader og fokusere på dei områda kor det er usemje mellom open og proprietær programvare.

Vi har ikkje vurdert nytteverdet eller om IT-kjøp er føremålstenleg for ei verksemd. Dette er noko alle verksemder bør gjere, men det er ikkje mogleg å gjere det på eit generelt grunnlag. Det vi ser på her, er kostnadssida i ei kost/nyttevurdering.

Med totale IT-kostnader meiner vi kostnadene for kjøp og bruk av eit bestemt IT-system i den tida IT-systemet er i bruk. Vi har delt "livsløpet" inn i fleire periodar som informasjonsinnhenting, innkjøp, utrulling og igangsetjing, opplæring, dagleg bruk og vedlikehald, og sal og overgang til ny teknologi. Nedanfor vert desse fasane nærare omtalt.

- *Informasjonsinnhenting.* Denne fasen inneber å skaffe seg informasjon om produktet ein seinare skal kjøpe. Her kan det vere vanskelegare å finne informasjon om open programvare, sida distributørane ikkje alltid har ei eiga marknadsavdeling. Informasjon som finnast kan vere meir korrekt enn for proprietær programvare, sida opne programvareverksemder i hovudsak ikkje tener pengane sine på sal av produktet. Dessutan er koden tilgjengeleg for inspeksjon.

Gjennom innkjøpsordningane sine kan offentleg sektor sørgje for at informasjon om open programvare er like tilgjengeleg som informasjon om annan type programvare.

- *Innkjøp.* Innkjøp i offentleg sektor må følgje regelverket for offentleg innkjøp. I utgangspunktet skal dette regelverket stille alle potensielle leverandørar likt. Vurderer ein eit kjøp berre ut frå pris, vil open programvare sjølvsagt kome best ut, sida han har ein låg initiell kostnad knytt til seg. Det er verd å merke seg at ein del programvare kjem saman med maskinvare, eller er ei oppgradering av eksisterande programvare. Difor er det ikkje mogleg å setje opp eit reknestykke som er dekkjande for alle verksemder, sjølv på dette isolerte området.
- *Utrulling og igangsetjing.* Dette omfattar installasjon og tilrettelegging av systemet, slik at det er klårt til bruk i heile verksemda. Store verksemder som sel proprietær programvare, er i dag betre rusta for dette enn verksemder som sel open programvare, fordi dei tradisjonelt har hatt eit større fokus på denne fasen. Skal store verksemder nytte open programvare må seljarane av open programvare etablere eit apparat som er i stand til å handtere utrullinga i stor skala.

-
- *Opplæring.* Denne kostnaden bør ikkje vere avhengig av om det er eit opent programvaresystem eller ikkje. Per i dag er det eit større utval av opplæringstilbod for proprietær programvare enn det er for open programvare. Prisen er ikkje forskjellig, men det er lettare å få eit tilbod som passer den ein skilde verksemda med proprietær programvare. Det er også heilt klårt fleire brukarar som er kjende med eksisterande proprietær programvare, enn med open programvare. Difor vil det vere kostnader forbunde med opplæring om ein skal bytte system.
 - *Dagleg bruk og vedlikehald.* Dette er ofte den fasen som utgjer den største kostnaden knytt til dei totale IT-kostnadene. Frå open programvaremiljø vert det hevda at open programvare vil gjere vedlikehald billigare, men så langt har vi ikkje funne dokumentasjon på dette.
 - *Sal og overgang til eit nytt system.* Nye IT-system vert ofte bygd for å erstatte eksisterande. Har ein tilgang til kildekoden (ikkje nødvendigvis open kildekode), vil overgangen til eit nytt system ofte verte enklare, sidan ein har betre oversikt over korleis det førre systemet er implementert. Det som er viktig i denne fasen, er at systemet er mogleg å dokumentere, og at ein kan ta med seg data frå det gamle systemet over til det nye.

Ved ei vurdering av dei ulike fasane i livsløpet til eit IT-system er det berre innkjøpsfasen som klårt skil seg ut til fordel for open programvare. Ein stor kostnad forbunde med overgang til open programvare er opplæring av brukarar i det nye systemet. For dei andre fasane er det vanskeleg å finne nokre store skilnader i tenkjelege kostnader knytt til bruken av open programvare.

På spørsmålet om det offentlege kan kutte kostnader ved ein overgang til open programvare, er konklusjonen vår: Ein vil kunne spare kostnader ved å nytte open programvare i verksemder der dei totale IT-kostnadene i hovudsak er knytt til kjøp av lisensar, og der kostnadene ved opplæring av brukarane på ”ny” programvare er relativt liten.

6 Døme på bruk av open programvare i forvaltninga

Utover bruken av Linux og Apache som webtenester, har vi ikkje så mange døme på medveten bruk av open programvare i forvaltninga. Dei vi har, kjem frå utdanningssektoren.

6.1 Universitetet i Oslo

Universitetet i Oslo nyttar open programvare på mange måtar. Fleire fakultet har Linux tilgjengeleg for sluttbrukarane. Mange har Linux som tenarmaskinar der dei køyrer til dels avanserte tekniske applikasjonar. Apache er utbreidd som webtenar, og infrastrukturprogramvare som Sendmail og Bind, er dominerande.

Det vi har nemnt nå er heilt vanleg bruk av open programvare. Ein annan bruk av Linux som ikkje har vore så vanleg tidlegare, men som er svært interessant, er å nytte Linux som ein tynn klient. Dette betyr at Linux berre kan nyttast for å handtere brukargrensesnitta frå program som vert køyrt på tenarmaskinane.

All brukarprogramvare finnast på tenarmaskinen. Dette er både vanleg kontorstøtteprogramvare og meir fagleg programvare. På tenarmaskinane køyrer ein Windows eller NT. Klienten vert berre nytta for å kommunisere med tenarmaskinen og vise fram skjermbilde. Brukaren ser dei vanlege Windows-skjermbilda, men det er eit Linux operativsystem som vert nytta for å vise fram skjermbilda. På denne måten kan ein unngå å installere Windows på alle PC-ane som finnast ute hjå sluttbrukarane.

Ein kan spare ein del lisenskostnader ved å bytte ut Windows med Linux og samstundes ha den gamle programvaren. Ulempa, eller kanskje fordelene er at ein går over til eit miljø med tynne klientar der sluttbrukaren mistar den direkte kontrollen over kva som skal finnast på eigen maskin.

I miljø der ein ynskjer å innføre tynne klientar til brukarane sine og ikkje la dei ha direkte tilgang til lokal programvare, kan det vere klokt å bytte ut Windows med Linux på alle klientane. Bruk av Linux i miljø der ein ynskjer å innføre tynne klientar, er aukande i universitets- og høgskulemiljø.

6.2 Høle barne- og ungdomsskule

Høle barne- og ungdomsskule har installert Linux på gamle PC-ar som skulen har fått frå næringslivet. PC-ane kjem utan operativsystem, og dette må installerast. Ved å velje Linux som operativsystem, er kostnadene til lisensar reduserte.

7 Kommersielle tenester for open programvare i Noreg

7.1 Kjøp av programvare

I Noreg er det enkelt å kjøpe open programvare. Vi tinga programvaren frå Akademika og fekk den levert to dagar seinare på døra. Det følgde med faktura som kunne handsamast på vanleg måte. Det er fleire butikkar i Noreg som sel open programvare via nettet, og fleire butikkar som sel Linux-distribusjonar over disk.

Det er også mogleg å tinga Linux-distribusjonar direkte frå leverandørane. Dette tek lenger tid, og med dei ekstra kostnadene knytt til frakt og tollhandsaming er det billigare å kjøpe frå leverandørar i Noreg. Fordelen er at ein kan få versjonar som ikkje er tilgjengeleg i Noreg, og ein kan få dei nyaste versjonane tidlegare.

Når det gjeld kommersiell programvare for Linux, vert dette selt på lik line med annan kommersiell programvare. Store aktørar som IBM, SGI og Oracle har alle versjonar av programvaren sin tilgjengeleg på Linux. Desse verksemdene har også eit støtteapparat for å betene dei kundane som vel å køyre programvaren deira på Linux.

7.2 Kjøp av maskinar med ferdig installert programvare

Fleire store og små leverandørar sel maskinar med ferdig installert open programvare. Dette gjeld i hovudsak maskinar med Linux, der ein i tillegg kan få tenarprogramvare som Apache og Sendmail ferdig konfigurert på maskinane.

7.3 Konsulenttenester

Både reine konsulentselskap og dei store kommersielle programleverandørane tilbyr konsulenttenester. Blant konsulentselskapa finnast det både dei store internasjonale som dekkjer "alt", og små nisjeselskap som berre fokuserar på Linux med tilhøyrande programvare.

7.4 Kurs og opplæring

Stendig fleire tilbyr kurs og opplæring på Linux og annan open programvare. Kursa vert gjerne halda av verksemdar som sel programvare for Linux/Unix, og av universitet og høgskular. Tilbodet av kurs og opplæring er ennå ikkje så omfattande som for Windows-plattformen.

8 Bruk av open programvare som standardiserings- og utviklingstiltak

Standardar og standardisering er ein viktig del av IT-utviklinga. Open programvare har ei rolle å spele i det framtidige arbeidet innafor standardisering. Somme arenaer innafor IT-standardisering, som IETF, reflekterer kulturen til open programvare, medan andre område ynskjer å nytte open programvare som eit verkemiddel for spreiding og betre interoperabilitet av standardar.

8.1 IETF i open programvare

Måten IETF (Internet Engineering Task Force) er organisert og fungerer på, er eit godt døme på ”Basar”-utviklingsprinsippa framsett av Eric Raymond [2].

IETF definerer opne standardar for grunnleggjande protokollar på Internett samt standardar for ulike fellesbruk som tryggleik, transporttenester og administrasjon. Alle IETF-standardar er fritt og gratis tilgjengeleg på Internett.

For at ein IETF-standard som er vedteken skal fortsette å vere ein standard, må det innan eit gjeve tidsrom minst finnast to ulike implementeringar av standarden som verkar saman. I denne samanhangen er open programvare viktig for å få spreidd implementeringar av standarden og for å kunne teste ulike produkt mot kvarandre.

IETF er open for alle, og du er ”medlem” gjennom å delta på diskusjonslister og møte. Alle opptreer som einskilde personar, og ikkje som deltakarar for ei verksemd eller offentleg mynde.

Både standardiseringsprosessen og standardane vert basert på openkap.

8.2 Internett og open programvarereferanseimplementeringar

Ei referanseimplementering av ein standard er ei implementering som andre viser til, eller testar mot, når dei skal lage interoperable produkt basert på denne standarden. Ei referanseimplementering bør implementere alle eigenskapane til standarden og gjere dette på ein korrekt måte. Apache er eit døme på ei referanseimplementering av http (standarden). Nettlesarar må kunne kommunisere med Apache for at dei kan hevde at dei er ei korrekt implementering av standarden.

Litt av suksessen til Internett er at produkta basert på Internett-standardane er interoperable. Vi får dermed ein samverkande heilskap. Dei fleste grunnleggjande protokollane for Internett finnast som opne programvarereferanseimplementeringar. Referanseimplementeringane har ofte ein tidleg versjon og ein dominerande marknad, og dei andre produkta må vere interoperable med referanseimplementering for å få innpass på marknaden. Døme er TCP/IP, DNS, SMTP og HTTP.

8.3 Open programvare som katalysator for utbreiing av nye standardar

For å få utvikla produkt basert på standardar er det viktig at det finnast implementeringar av standarden som ein kan byggje vidare på. Her er det viktig med

open programvare. Finnast standarden implementert som ei open programvarereferanseimplementering, kan alle nytte denne implementeringa av standarden for å utvikle nye produkt. Finnast det berre proprietære implementeringar av standarden, må folk som ynskjer å utvikle nye produkt basert på standarden, fyrst implementere sin eigen versjon av standarden, eller kjøpe rettar til å bruke ei proprietær utgåve. Ho vil vere eit hinder for nyutvikling og kan sakke prosessen med å få teke i bruk nye standardar.

I samband med utviklinga av nye standardar for å støtte implementeringa av EU-direktivet om elektroniske signaturar, har EU-kommisjonen bedt om å få lagd opne referanseimplementeringar av nokre av dei standardane frå ETSI som allereie er vedtekne. Dette for å lette produktutviklinga og forskinga på området.

I Noreg kan Noregs forskingsråd og Statens nærings- og distriktsutviklingsfond vere moglege aktørar for å fremje bruken av open programvare i utviklingsarbeid. Noregs standardiseringsforbund kan vere ein aktør for å fremje open programvare innafør arbeidet med IT-standardisering.

9 Vurdering av bruksområda til open programvare

9.1 Vurderingskriterium for open programvare

Vi har skilt mellom open programvare som er meint for tenarmaskinar, og for infrastruktur og open programvare som primært er tenkt nytta av vanlege sluttbrukarar, når vi har valt vurderingskriterium.

Når det gjeld vurdering av open programvare for tenarmaskinar og infrastruktur, har vi lagd vekt på

- *innkjøp*. Programvaren skal vere enkel å kjøpe.
- *installasjon*. Det skal vere lett å installere programvaren, både fyrste gong og ved seinare oppgraderingar.
- *support*. Det skal finnast profesjonell support på programvaren.
- *administrasjon og drift av systemet*
 - fleksibelt, enkelt og høve for tilpassing
 - kompatibelt med andre system
 - marknadsposisjon
 - driftskostnader
- *standardar*. Følgjer programmet gjeldande standardar på området?
- *opplæring*. Det må finnast høve for opplæring i systemet.
- *brukelegskap for offentleg sektor*. Kor brukeleg er programvaren for offentleg sektor?

Når det gjeld vurdering av open programvare for sluttbrukarar (operativsystem med program som teksthandsamar og e-post), har vi lagd vekt på

- *innkjøp*. Programvaren skal vere rimeleg og enkel å kjøpe.
- *installasjon*. Det skal vere lett å installere programvaren, både fyrste gong og ved seinare oppgraderingar. Nye program skal kunne installerast og fjernast enkelt.
- *brukarstøtte og support*. Det må vere mogleg å få brukarstøtte for den installerte programvaren.
- *opplæring*. Det må finnast høve for opplæring i bruk av dei ulike programma.
- *brukarvenlegskap*. Programma må vere like ”brukarvenlege” som tilsvarande kommersiell programvare. For vanlege kontorstøtteprogram vil Microsoft Office vere eit naturleg program å samanlikne med.
- *språk*. Programma må vere tilgjengeleg på norsk.
- *driftskostnader*. Kostnader for drift av programvaren over tid.
- *brukelegskap for offentleg sektor*. Kor brukeleg er programvaren for offentleg sektor?

Val av kriterium er gjord ut frå kva vi trur er mest sentralt, og dekkjer alle fasar av kjøp og bruk av programvare. Ut frå kriteria skal det vere mogleg å gjennomføre ein analyse av dei totale kostnadene for kjøp og bruk av programvaren²⁰.

²⁰ Dette vert gjerne omtalt som total cost of ownership (TCO for IT-produkt

9.2 Kva for delar av forvaltninga er best eigna for bruk av open programvare?

Forvaltninga er svært variert. Ikkje alle delar er like eigna for å ta i bruk open programvare. Etatar og sektorar med store, tunge program som dei har utvikla sjølv, eller der ein er avhengig av eigne sakshandsamingssystem, er ikkje så velegna for å ta i bruk open programvare fordi

- det ikkje finnast den typen open programvare
- den typen programvare ikkje er eigna til å utvikle som open programvare.

Område der hovudbruken er ein kombinasjon av teksthandsaming, e-post og Internett, kan vere velegna for bruk av open programvare. I denne kategorien finn vi mellom anna mange departement, nokre direktorat og store delar av utdanningssektoren. Desse vil kunne ta i bruk ein eller annan form av open programvare. Utdanningssektoren har store variasjonar i den programvaren som vert nytta, medan direktorat og departement i stor grad nyttar den same programvaren²¹. Evna til å ta i bruk ny programvare verker dermed større i utdanningssektoren enn i departementa og direktorata. I tillegg har utdanningssektoren som føremål å lære opp brukarane sine, noko som er ein føresetnad for å ta i bruk nye typar av programvare. På høgskular og universitet finnast det også veldig mykje IT-kompetanse blant undervisningspersonalet. Sett under eitt er utdanningssektoren den sektoren der det er mest fornuftig å starte for å ta i bruk open programvare i forvaltninga.

Vurderinga vår er at open sluttbrukarprogramvare ikkje per i dag kan erstatte dei kommersielle alternativa når det gjeld funksjon og integrasjonshøve.

9.3 Kva for type open programvare er eigna i forvaltninga?

Som med annan programvare er det store variasjonar i kor brukarvenleg og brukleg open programvare er. I samband med denne vurderinga har vi sett på programvare for sluttbrukarar og tenarprogramvare. Programvare for sluttbrukarar omfattar det grafiske brukarmiljøet, teksthandsaming, rekneark, nettlesar og e-postklient. Tenarprogramvare omfattar programvare for e-post, etablering av nettsider og vanleg filhandsaming.

9.3.1 Sluttbrukarprogramvare

Det finnast fleire gode teksthandsamarar og rekneark for Linux. Ulempa er at desse ikkje er open programvare. Dermed har ein framleis dei same ulempene som med annan proprietær programvare. Dei teksthandsamarane og reknearka som finnast som open programvare, manglar diverre den heilskapen og dei integrasjonane som dei proprietære produkta har. Når det gjeld nettlesarar og e-postklientar, finnast mange av dei same produkta tilgjengeleg både på Linux og Windows. Ikkje alle av desse produkta finnast som open programvare, men det er ofte ein låg eller ingen lisenskostnad knytt til bruken av produkta.

Per i dag manglar teksthandsamarar rekneark og e-postklientar utvikla som open programvare, den heilskapen og gagnlegskapen som finnast hjå dei kommersielle produkta. Om vi tek omsyn til at det for dei fleste brukarane vil medføre opplæring og omstilling å gå over til nye produkt, bør dei nye produkta ha tilnærma same

²¹ Dette kan lesast ut av bakgrunnstala til rapporten "IT i staten 1999"

gagnlegskap som det produktet ein skiftar frå. Pris vert her mindre viktig for ein sluttbrukar. I situasjonar der ein skal lærast opp og ein ikkje har gjort seg avhengig av gagnlegskapen til dei proprietære produkta, er open programvarealternativa veleigna. Vurderinga vår er at open programvare meint for sluttbrukarar vil eigne seg best i utdanningssektoren og hjå spesielt motiverte og interesserte brukarar.

9.3.2 Tenarprogramvare

Når det gjeld tenarprogramvare er situasjonen annleis. Programvare for drift og utvikling av nettsider er i stor utstrekning allereie open programvare. I tillegg finnast det fleire døme på open tenarprogramvare for e-post. På desse områda er det heilt klårt eit stort potensial for open programvare i forvaltninga. Diverre er det også på desse områda ein tilbyr rimeleg eller gratis proprietær programvare, slik at overgangen til open programvare ikkje medfører store økonomiske innsparingar. Bruk av open programvare kan vere fornuftig for å hindre den einsrettinga av programvare som spesielt skjer innafør departement og direktorat.

På tenarprogramvare er den opne programvaren funksjonelt og administrativt minst like god som dei proprietære alternativa. I noka utstrekning kan det mangle support på programvaren frå leverandørane. Dette er i ferd med å verte endra, sidan fleire store leverandørar tilbyr support på open tenarprogramvare som Apache, Samba og Sendmail.

Vurderinga vår er at Linux er ein veleigna tenarmaskin for offentlig forvaltning. Dette gjeld spesielt i kombinasjon med Apache som tenar for eigne nettsider.

10 Konklusjonar, moglege tiltak og tilrådingar

10.1 Konklusjonar

Det offentlege er ikkje tent med for stor einsretting av den programvaren som vert nytta. Kvar einskild verksemd bør så langt som mogleg stå fritt til å velje den programvaren som verksemda finn mest gagnleg for å løyse oppgåvene sine. Dette krev at verksemdene har kunnskap om, og kjennskap til ulike produkt og tenester. *Open programvare er ein billig måte folk kan gjere seg kjend med ny programvare på.*

Open programvare er eit konsept som forvaltninga bør vere medveten å utnytte. Tilgang til kildekoden betyr at fleire kan få kjennskap til den grunnleggjande teknologien, og dette kan gjere det enklare å få utbreidd og utvikla ny teknologi. *Det er spesielt innafør utdanning, forskning og utvikling at open programvare vil gjere seg gjeldande.*

Open programvare er dominerande når det gjeld implementeringa av Internett-infrastrukturen. Både utviklinga og implementeringa av standardar for Internett er basert på ideane og tankane rundt open programvare. *Norske mynde bør vere medvetne om dei høva som open programvare gjev for ei rask spreiding av nye standardar i marknaden.*

For å utvikle open programvare via nettet treng dei som er involverte, ei felles forståing av og innsikt i det som skal utviklast. *Det er derfor veileigna å utvikle tenar- og infrastrukturprogramvare som open programvare.* Årsaka er at det her som oftast føreligg klare spesifikasjonar og standardar som ein kan bruke når ein skal lage produkt. Som regel har dei som skal implementere ein standard, også vore med på å definere han. Sluttbrukarprogramma er ikkje så veileigna å utvikle som open programvare, sidan det ikkje finnast standardar for dette, og at det ikkje er så lett å kome opp med ein klår spesifikasjon av det som skal lagast. Dette gjenspeglar seg også i kva program som finnast som open programvare.

Det er ikkje alltid den beste teknologien som går vinnande ut av ein tvikamp. Dette er det viktig å ha med seg. Det er fleire faktorar enn dei reint tekniske som ligg til grunn når folk skal velje eit operativsystem, eller tekniske apparat. Andre viktige ting er marknadsføring av produktet, design, pris, tilgjengeleg utstyr (for operativsystem vil dette seie kva for program som finnast, og kva for maskinvare som vert støtta), kva du forbind med produktet, og ikkje minst kven det er som bruker det.

Vurderinga vår av Linux og open programvare vert dermed ikkje ei rein teknisk vurdering, men ei heilagvurdering der vi også tek omsyn til erfaringane til brukarane og dei investeringane som allereie er gjort på IT-området i offentleg sektor.

Det er ikkje gjeve at open programvare er billigare enn kommersiell programvare. *Etter vurderingane våre vil det vere kostnadssparande å nytte open programvare i verksemdar der dei totale IT-kostnadene i hovudsak er knytt til kjøp av lisensar, og kostnadene ved opplæring av brukarane på "ny" programvare er relativt liten.*

Linux er i hovudsak eit operativsystem som er meint nytta på tenarmaskinar. Dette gjenspeglar også korleis dei fleste Linux-distribusjonar vert marknadsførte, og korleis dokumentasjonen som følgjer med programma, er²².

Linux er ein Unix-variant, og det er dermed naturleg at Linux fyrst og fremst vil verte nytta på dei områda der andre Unix-operativsystem tidlegare har vorte nytta. Unix-system har i hovudsak vorte nytta som tenarmaskinar, og vi ser at Linux også i hovudsak vert teke i bruk som tenarmaskinar. Dei miljøa der sluttbrukarane har nytta Unix, vil også vere dei fyrste miljøa der sluttbrukarane tek i bruk Linux.

Det er spesielt innafor undervisningssektoren at Linux kan spele ei rolle både som tenarmaskinar, og som operativsystem for sluttbrukarar. Gjennom å oppmuntre til bruk av forskjellige programvare i opplæring kan brukarane få eit breiare kunnskapsgrunnlag og der igjennom eit betre grunnlag når dei seinare skal velje programvare sjølv.

Linux er i dag velegna som operativsystem for tenarmaskin, også for offentleg sektor. Vurderinga vår er at open sluttbrukarprogramvare per i dag ikkje kan erstatte dei kommersielle alternativa når det gjeld funksjon og integrasjonshøve.

10.2 Moglege tiltak

Det finnst mange moglege tiltak for å få ei større utbreiing av open programvare i offentleg forvaltning. I dette punktet omtalar vi nokre av desse:

- Påleggje bruk av open programvare
- Støtte bruken av open programvare
- Distribuere og utvikle open programvare
- Støtte forskning og utvikling av open programvare

10.2.1 Påleggje bruk av open programvare

Eit kraftig verkemiddel er å påby bruk av open programvare i gitte samanhengar. Pålegg om bruk av ein bestemt teknologi vil alltid vere risikabelt. Er det den rette teknologien ein påbyr? Er verksemdene som vert pålagt bruk, klare for å ta teknologien i bruk? Er mottakarane av pålegget motiverte for å ta teknologien i bruk?

I hovudsak bør ein vere tilbakehalden med å pålegge bruk av ein bestemt teknologi, men heller framheve fordelar og ulemper med teknologien, slik at verksemdene sjølv kan velje å ta teknologien i bruk.

10.2.2 Støtte bruken av open programvare

Med å støtte bruken av open programvare meiner vi her å førebu omgivingane slik at det vert enklare å ta i bruk open programvare. Tiltaka kan spenne frå å forenkla

²² Dette sitatet er henta frå den trykte dokumentasjonen som følgjer med Caldera OpenLinux 2.3: "Today, Linux is first and foremost a server operating system. Although many applications are now appearing that allows Linux to be used as a primary workstation or desktop system, most users of Linux focus on the server capabilities of the operating system."

innkjøpsprosessen for open programvare gjennom opplæringstilbud for bruk av open programvare til hjelp til drifting av open programvare.

Ved å førebu for bruk av open programvare kan ein stimulere til større bruk av denne typen programvare utan å tvinge nokon til å bruke ein type programvare dei ikkje ynskjer.

10.2.3 Distribuere og utvikle open programvare

Eit konkret tiltak for å støtte opp under bruken av open programvare er på den eine sida om staten sjølv distribuerar ein eigen versjon av Linux med annan tilhørande open programvare. Dette vil kunne gjere det enklare for verksemdar å ta i bruk open programvare. På den andre sida vil det kunne verke marknadsvridande om det offentlege har sin eigen Linux-distribusjon. Kva distribusjonen skulle innehalde, vil også vere eit sentralt spørsmål. Eit anna spørsmål er om ein eigen distribusjon vil vere kostnadssvarande sidan det allereie er gratis å vidaredistribuere dei eksisterande distribusjonane av Linux.

Ei mellomløysing ville vere at ein eller annan i det offentlege kopierte opp og distribuerte eksisterande Linux-distribusjonar, slik at dei som ynskjer å prøve ut open programvare, kunne få han på ein enkel måte ved å vende seg til ein offentlig etat.

Det offentlege bør også vurdere om det kan tilby programvare som det i dag har eigarrettane til, som open programvare.

10.2.4 Støtte forskning og utvikling av open programvare

Ved å støtte utvikling av open programvare kan ein hjelpe til med å vidareutvikle infrastrukturen samstundes som ein gjer teknologien tilgjengeleg for alle. Som nemnt tidlegare er det ein trend at implementering av nye standardar skjer som open programvare. Set ein krav frå dei som ytar pengar til forskning og utvikling, at resultatet skal vere tilgjengeleg som open programvare, får ein som ein ytterlegare effekt at resultatet av forskinga enklare kan nyttiggjerast av andre. Vi trur at ved tildeling av forskings- og utviklingsmidlar bør det vere eit krav at programvare som vert utvikla, vert gjort tilgjengeleg som open programvare.

10.3 Tilrådingar

- Linux er eit produkt som det offentlege bør støtte for å skape eit press for vidareutvikling av, og som eit potensielt alternativ til Microsofts operativsystem. Linux er i dag best eigna som eit tenaroperativsystem.
- Det offentlege bør støtte utviklinga av open programvare for å skape eit alternativ til eksisterande programvare. Ny open programvare kan skape eit press på vidareutvikling av eksisterande programvare og kan hindre ei for stor einsretting av den programvaren som vert nytt i offentlig sektor. Støtta kan kome gjennom forskings- og utviklingstiltak.

-
- Ved tildeling av forskings- og utviklingsmidlar kan det vere eit krav at programvare som vert utvikla, vert gjort tilgjengeleg som open programvare.
 - Det offentlege bør vurdere nærare om det kan offentleggjere kildekode, som det har eigarrette til, som open programvare.
 - Det offentlege bør oppmuntre til bruk av Linux og open programvare innafør skule- og utdanningssystemet. Årsaka er at elevar og studentar skal få kjennskap til fleire ulike produkt, slik at dei seinare kan velje dei produkta dei sjølv ynskjer på eit betre grunnlag.
 - For å spare lisenskostnader kan brukte PC-ar som vert gjeve til skular, verte utstyrt med open programvare.
 - Infrastrukturen bør baserast på opne standardar implementert som open programvare. Offentleg sektor bør setje krav til bruk av opne standardar implementert som open programvare i den infrastrukturen den nyttar. Eit område der bruk av opne standardar og løysingar basert på open programvare kan vere fornuftig, er ved ei eventuell etablering av ein infrastruktur for distribusjon og handtering av digitale sertifikat.

Vedlegg I

Referansar

- [1] Chris. Dibona, Sam Ockman & Mark Stone (Editors): Open Sources – Voices from the Open Source Revolution
- [2] Eric S. Raymond: The Cathedral & the bazaar. O*Reilly & Associates, 1999. ISBN 1-56592-582-3
- [3] The Open Source Initiative. The open source definition, 1999 Tilgjengeleg på <http://www.opensource.org/osd.html>
- [4] Free Software/Open Source: Information Society Opportunities for Europe? Version 1.2 April 2000
Working group on Libre software²³ Tilgjengeleg på <http://eu.conecta.it/paper.pdf>
- [5] Linux Journal. Juni 2000 “The new beginning”
- [6] NOU 2000:24 Et sårbart samfunn
- [7] Open Source the unauthorized white papers. Donald K. Rosenberg M&T books 2000. ISBN 0-7645-4660-0

²³ Arbeidsgruppa for Libre Software vart satt saman på initiativ av EU-kommisjonen sin kommisjonær for området ”Information Society”

Vedlegg II

Forkortingar

BIND	Berkeley Internet Naming Deamon
BSD	Berkeley Software Distribution
DNS	Domain Name System
ESMTP	Extended Simple Mail Transfer Protocol
ETSI	European Telecommunications Standards Institute
GNOME	The GNU Network Object Model
GNU	Gnu's Not Unix
GPL	GNU General Public License
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IEC	International Electro technical Committee
IEEE	The Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Taste Force
IMAP	Internet Message Access Protocol
IP	Internet Protocol
KDE	The K Desktop Environment
OSI	Open Systems Interconnection
PKI	Public Key Infrastructure
POP	Post Office Protocol
POSIX	Portable Operating System Interface for unIX
RFC	Request For Comments
RPM	RedHat Package Manager
SMTP	Simple Mail Transfer Protocol
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol

Vedlegg III

GPL (GNU General Public License)

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, INC
59 Temple Place – Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (The GNU Library General Public License covers some other Free Software Foundation Software instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, software patents threaten any free program constantly. We wish to avoid the danger that re distributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work, which contains a notice placed by the copyright holder saying it, may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: That is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true, depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy and appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distributes such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary

way, to print or display an announcement including an appropriate copyright notice and a notice there is no warranty (or else, saying that you provide a warranty) and that users may re distribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: If the Program itself is interactive but does not normally print such an announcement,)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for non-commercial executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering, access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program are restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License, which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software, which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING

RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software that everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

One line to give the program’s name and to have an idea of what it does. Copyright © yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, writ to the Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you be electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright © year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w`. This is free software, and you are welcome to redistribute it under certain conditions; type `show c` for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-click or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc, hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

Signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Return to GNU's home page.

Send FSF & GNU inquiries & questions to gnu@gnu.org. Other ways contact the FSF.

Comments on these web pages to webmasters@www.gnu.org, send other questions to gnu@gnu.org.

When Copyright notice above.
Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111,
USA

Updated: 3 Jan 2000 rms

Vedlegg IV

The BSD License (Berkeley Software Distribution)

Copyright © <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

REFERANSAR

Tittel:	Open programvare. Bruken av Linux og annan open programvare i statleg forvaltning
Forfattar	Endre Grøtnes
Statskonsults rapportnummer:	2001:7 (NYN)
Prosjektnummer:	43 488
Prosjektnamn:	Open programvare
Prosjektleiari:	Endre Grøtnes
Oppdragsgjevar:	Statskonsult
Resymé:	
Arbeidsområde:	Informasjonsteknologi
Emneord:	Linux, Open Source, open programvare, opne standardar
Dato:	Desember 2001
Sider:	46
Utgjevar:	Statskonsult Eit direktorat for kommunikasjon, kompetanse og forvaltning Postboks 8115 Dep 0032 OSLO