Fri programvara (FOSS), Öppen källkod (Open Source), öppna standarder och decentralisering

vad det betyder och varför det är viktigt

Elias Rudberg

Glasklar Teknik AB
elias@glasklarteknik.se

Maj 2025

Intro

Vem är jag?

Elias Rudberg

- Jobbar på Glasklar Teknik AB
- Engagerad i Föreningen för Digitala Fri- och Rättigheter (DFRI)
- Tidigare jobbat som forskare i beräkningsvetenskap vid Uppsala universitet
- Tidigare jobbat med utveckling på Bahnhof

Två delar

Två delar av föredraget:

(1) Allmänt om Fri programvara (FOSS), Öppen källkod (Open Source), o.s.v. i samhället idag

(2) Vad FOSS, Open Source o.s.v. betyder specifikt för den som jobbar som utvecklare/programmerare

Två delar: del 1

(1) Allmänt om Fri programvara (FOSS), Öppen källkod (Open Source), o.s.v. i samhället idag

Översikt

Vi ska prata om:

- Problem som finns idag
- Fri programvara och öppna standarder
- Decentralisering
- Olika perspektiv:
  - individ
  - företag
  - anställd
  - staten
- Exempel på vad man kan göra

Problem som finns idag

Ett fåtal väldigt stora företag (“Big Tech”) har stor makt över människor och samhälle.

Massövervakning

Många upplever att de inte har något val, tror att det inte går att själv ha kontroll över tekniken.

“Big Tech”

https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-capitalization/
“largest companies in the world by market capitalization in 2024”

Topp 7:

- Microsoft
- Apple
- Nvidia
- Alphabet (Google)
- Amazon
- Saudi Aramco (ojlebolag)
- Meta (Facebook)

Fri programvara och öppna standarder

Fri programvara gör det möjligt för den som använder programmet att själv ha kontroll, genom tillgång till källkoden.

Öppna standarder för kommunikation (t.ex. internet-protokoll, TCP/IP osv) gör det möjligt att kommunicera fritt, med hjälp av fri programvara.

Fri programvara – vad är det?

Definition enligt Free Software Foundation (FSF):

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

Fri programvara

Fördelar

Kunna kontrollera vad programmet faktiskt gör

Möjlighet att själv lösa problem

Öppenhet ger bättre kvalitet och säkerhet

Programmets ägare/utvecklare har inte makt över användarna, eftersom varje användare själv kan ändra programmet

Fri programvara

Fördelar, forts.

Möjlighet att själv lösa problem – jaha, måste man verkligen själv göra det, alldeles ensam? Kan man inte ta hjälp av någon?

— Jo, såklart!

En fördel är också just att när det finns många användare som alla har möjlighet att granska och förbättra koden, då höjs kvaliteten.

Fri programvara

Kärt barn har många namn

Olika namn som betyder samma sak:

- Free software
- Libre software
- Free and Open Source Software (FOSS)
- Free/Libre and Open Source Software (FLOSS)

Namn som ibland betyder samma sak som Free software: “Open Source Software”.
OBS: fri programvara betyder mer än bara öppen källkod.

“Open Source” vs “Free Software”

Vilket begrepp som används beror ofta på vad man vill fokusera på:

“Open Source Software”: fokus på just att källkoden finns öppet tillgänglig, inte så mycket fokus på vilken frihet det ger. Kanske ger det inte så mycket frihet beroende på licensvillkor.

“Free Software”: fokus på *frihet* kopplat till programvaran, där öppen källkod är en nödvändig förutsättning.

Motsatsen till FOSS: “Proprietary Software”

Programvara som är stängd och där någon “äger” och hemlighåller källkoden, kallas “proprietär programvara”, på engelska “Proprietary Software”.

De som gillar FOSS brukar betrakta “Proprietary Software” som något negativt som bör undvikas.

$\sim$“stängd”, $\sim$“inte öppen”

Ägaren, den som är “proprietor”, kan distribuera binärer, sälja licenser för programvaran, o.s.v. och tjäna pengar på det.

Öppna standarder

Exempel

Viktig skillnad:

(1) Alice har en hemsida på internet. Vem som helst som har en webbläsare (öppen standard) kan läsa info där. $\longrightarrow$ **bra!**

(2) Bob har istället en Facebook-sida. "Alla har ju ändå Facebook" tänker Bob. Facebook äger kommunikationen och bara den som lyder Facebooks villkor får läsa info där. $\longrightarrow$ **dåligt!**

Decentralisering

Internet har i grunden en decentraliserad uppbyggnad.

Olika delar kan fungera oberoende av varandra

All kommunikation behöver inte passera en central punkt

Decentralisering

Fördelar

Robusthet: det händer alltid att saker går sönder, men ett decentraliserat system fortsätter fungera även om vissa delar har problem.

Skydd mot angrepp: om någon vill sabotera systemet finns ingen central punkt som kan angripas för att slå ut allt.

Skalbarhet: systemet kan byggas ut effektivt på många ställen samtidigt, ingen central punkt blir bromskloss.

Fördelar med FOSS – olika perspektiv

individ

Möjlighet att skydda personlig integritet och privatliv.

T.ex. kanske du har filer i din dator som är privata, som du inte vill att någon annan, varken staten eller företag, ska veta något om.

Om du använder bara fri programvara på din dator, kan du ha kontroll.

Om du installerar stängd (ofri) programvara med okänd källkod, tappar du kontrollen. Du kan då inte veta vad programmet gör.

Fördelar med FOSS – olika perspektiv

företag

Fri programvara $\longrightarrow$ företaget kan självt ha kontroll över sin verksamhet.

Stängd programvara $\longrightarrow$ företaget blir beroende på den som utvecklar koden. Förutsättningar kan ändras när som helst.

dessutom: inlåsning

Fördelar med FOSS – olika perspektiv

anställd

Fri programvara $\longrightarrow$ fokus på anställdas kompetens att lösa problem själv. Kompetens lönar sig.

Stängd programvara $\longrightarrow$ mindre viktigt med anställdas kompetens.

Fördelar med FOSS – olika perspektiv

staten

Fri programvara $\longrightarrow$ **transparens**, i linje med offentlighetsprincipen. Förbättring av programvaran innebär ett bidrag till det allmänna bästa, gemensamt ägt.

Stängd programvara $\longrightarrow$ folk vet inte vad staten gör. Staten själv vet inte heller, istället har ett fåtal företag makten. Inlåsning. Slöseri med skattepengar.

Få företag med teknikmonopol tjänar mycket på detta. Andra företag, civilsamhälle, offentlig sektor och medborgarna förlorar.

Fördelar med FOSS – olika perspektiv

staten, forts

Diktatur $\longrightarrow$ vill gärna ha centralisering och stängda system.

Demokrati $\longrightarrow$ bör sträva efter att skydda människor från statens övergrepp, kan uppnås med fri programvara, öppen källkod, öppna standarder.

Fördelar med FOSS – olika perspektiv

Utbildning

Vad bör elever/studenter få lära sig?

Kunskap om stängda system ägda av företag $\longrightarrow$ bra för det företaget, dåligt för studenten, dåligt för samhället

Kunskap om fri programvara och öppna standarder $\longrightarrow$ bra för studenten och bra för samhället

Fördelar med FOSS – olika perspektiv

Hur vi lär oss saker

Människor lär sig bäst genom att undersöka hur något fungerar, ta isär, titta inuti, testa, osv.

Det möjliggörs och uppmuntras av fri programvara, man lär sig mycket mer.

(jämför med en stängd låda som det är förbjudet att titta inuti, hur ska man då lära sig hur den fungerar?)

Fördelar med FOSS – olika perspektiv

Forskning

Inom forskning forskning och i universitets-världen är öppenhet självklart.

Forskningsresultat publiceras och kan citeras. Kunskap byggs genom öppenhet.

$\longrightarrow$ öppen källkod naturligt inom forskning och utbildning.

Fördelar med FOSS – olika perspektiv

skydd mot inlåsning

Nackdel med inköp av stängda system: **inlåsning**

Stora pengar betalas till det företag man köper systemet från

Vid problem tvingas man betala ännu mer till samma företag (koden är ju stängd och ägd av dem så ingen annan kan lösa problem)

Har företaget då alls incitament att göra ett bra jobb?

$\longrightarrow$ Öppen källkod ger mycket bättre förutsättningar för **konkurrens**

Operativsystem

Exempel på fria operativsystem:

GNU/Linux (Debian, Ubuntu, Fedora, ...)

FreeBSD, OpenBSD

OBS: med stängt operativsystem har du inte kontroll, även om du startar fritt program där

Även viktigt för smartphones, två dominerar: iOS (Apple har kontrollen) eller Android (Google har kontrollen)
GNU/Linux för smartphones finns, även “degoogled Android”

Vem äger din dator?

Om du själv inte kan bestämma vilken programvara som körs på den, är det då verkligen ”din” dator?

Vem äger din smartphone? (Du? Google? Apple?)

Vem äger din bil?

...

Exempel på problem: e-legitimation i Sverige

Närapå de-facto krav i Sverige idag: alla förväntas ha "Bank-ID"

"Bank-ID" är stängd programvara, "proprietary software", ägd av bankerna och fungerar endast på stängda operativsystem ägda av Apple/Google/Microsoft. De bolagen har alltså kontroll över allas identiteter i Sverige, de kan fejka signaturer o.s.v.

$\longrightarrow$

Det svenska samhället är idag helt "ägt" av de tre USA-bolagen.

Vad händer vid problem?

Med fri programvara kan du själv undersöka och lösa problem som uppstår.

Kontrast till stängd programvara, där kan du inte göra någonting alls själv, annat än att “ringa support” och hoppas på hjälp. Får du inte svar därifrån är du helt hjälplös.

Vem har kontrollen?

Citat, Richard Stallman:

> *"Either the users control the program, or the program controls the users"*

Exempel: Linux

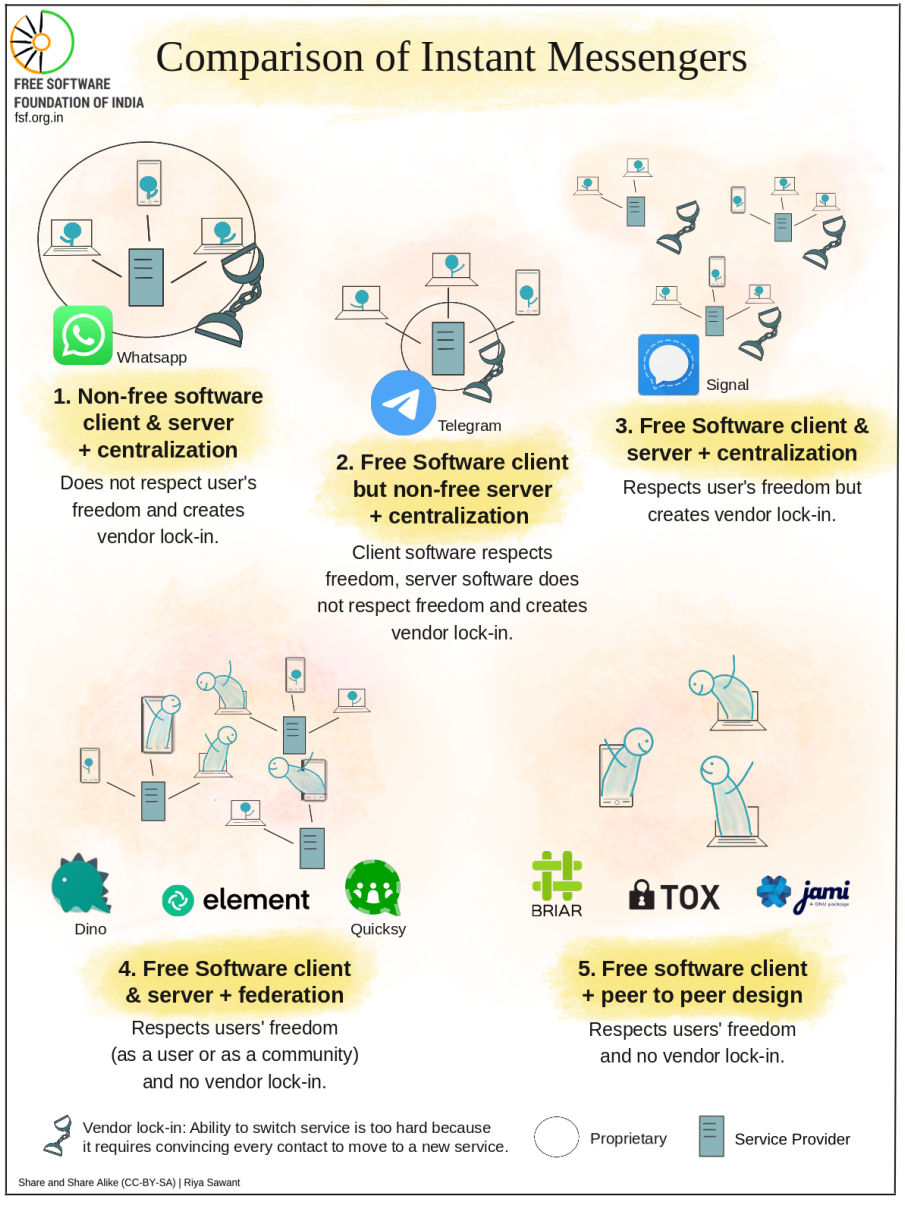
Linux är en kernel (kärnan i ett operativsystem), skapad av Linus Torvalds 1991 när Linus var student, publicerad som fri programvara (öppen källkod, FOSS).

Sedan dess har $\sim$ 20 000 andra utvecklare bidragit och idag dominerar Linux i världen när det gäller operativsystem på servrar och även smartphones (Android bygger på Linux).

Till och med Microsoft har senaste tiden börjat använda Linux.

$\longrightarrow$ open source är tydligen ett bra sätt att utveckla kod.

Exempel: instant messengers



Exempel: kompilatorer

Kompilatorer som är fri programvara:

- gcc: GNU Compiler Collection
- LLVM/clang

Fri programvara $\longrightarrow$ du kan ha kontroll över programmet som byggs. Du kan felsöka även när det gäller hur kompileringen går till. $\longrightarrow$ mer egenmakt och möjligheter för dig som programmerare.

Exempel: Mastodon

Mastodon är fri programvara.

Den som vill kan starta en egen instans (server) och kommunicera med andra instanser.

Fritt, decentraliserat alternativ till Twitter.

Fler exempel

- Jitsi (video-konf)
- BigBlueButton (video-konf)
- Matrix protocol, Element m.fl. klienter
- PeerTube
- Tor-nätverket
- ssh
- pgp / gpg (kryptering)
- git
- apache and nginx (web servers)

Frågor?

Frågor efter del 1?

Två delar: nu kommer del 2

(2) Vad FOSS, Open Source o.s.v. betyder specifikt för den som jobbar som utvecklare/programmerare

git

"**git**" är en fri programvara (GPL) för decentraliserad versionshantering, skapad av Linus Torvalds för att det behövdes för Linux.

Med git kan man göra "**git clone**", "**git pull**", "**git commit**", "**git push**", o.s.v.
Man kan enkelt hosta egen git-server för sina egna git-repon om man vill.

Det finns också centraliserade tjänster som använder git, t.ex.: github.com, gitlab.com, codeberg.org

OBS: Github är *inte* samma sak som git!

privacy, security, freedom

Tre saker som öppen källkod kan hjälpa till med:

- frihet
- säkerhet
- privacy

Friheten gör det möjligt att uppnå säkerhet och privacy.

Varför FOSS ger frihet

Om man inte har tillgång till källkoden är man “ägd” av den som har källkoden. Man är inte fri.

FOSS ger frihet för att om programvaran inte gör det användaren vill så kan man ändra den så att den gör det man vill.

Varför FOSS kan ge säkerhet

För att kunna säkerställa att ens system är säkra behöver man tillgång till källkoden.

Tillgång till källkoden gör att man kan granska den, och man kan få hjälp av många andra att granska den.

Bäst säkerhet uppnås om väldigt många granskar källkoden väldigt noga.

Mer om säkerhet

Två synsätt:

(1) Källkoden måste hållas hemlig så att inte skurkar får veta hur programmet fungerar. (den som tror detta, tror att FOSS är dåligt)

vs

(2) Källkoden måste vara öppen så att så många som möjligt kan granska den. $\longrightarrow$ FOSS är bra!

Argument för (2): angripare kan få tag i källkoden även om den inte är tänkt att vara "öppen". Bättre då att alla har tillgång.

Mer om säkerhet

kryptografi

De som jobbar seriöst med säkerhet använder krypterings-algoritmer som är öppna, med implementationer som är **öppen källkod**.

Det är alltså *in*te hemligt hur krypterings-algoritmerna fungerar, det som är hemligt är krypterings-nycklarna.

Riskabelt att använda en hemmasnickrad "hemlig" lösning som inte har granskats lika mycket som de mest använda öppna algoritmerna.

Mer om säkerhet

reproducible builds

Ett kraftfullt sätt att förbättaer säkerheten är att använda **reproducible builds**:

Se till att källkoden finns öppet tillgänglig och att kompilering och bygge av koden är helt **reproducerbart**:

Om jag bygger på min dator och du bygger på din, från samma källkod, ska vi få samma binär som resultat.

$\longrightarrow$ kan verifiera att man har rätt binär genom att kolla en checksumma.

$\longrightarrow$ "den här binären är byggd från den officiella öppna källkoden"

Varför FOSS kan ge privacy

Om man inte har tillgång till källkoden kan man inte veta vad programvaran gör, man kan inte veta om ett program t.ex. samlar in information om en själv och skickar det någonstans.

Mer om säkerhet/privacy

motargument

"Men Apple/Google/Microsoft lovar ju att det här är säkert/privat, jag litar på dem!"

Tre problem:

- De kanske lurar dig
- De kanske gör sitt bästa men misslyckas
- Osund koncentration av makt om allt hänger på dem

Öppen källkod gör det möjligt att själv ta ansvar för säkerhet/privacy.

Mer om operativsystem

För kontroll behöver man öppen källkod för allt, från operativsystemet och uppåt.

OBS: allt ditt program gör går via operativsystemet.

Exempel:

- Vad som visas på datorskärmen
- Input från användaren (tangentbord/pekskärm)
- Kamera, mikrofon
- Nätverkstrafik

Enkelhet är viktigt

För kontroll krävs två saker:

- Tillgång till källkoden
- Möjlighet att granska och förstå källkoden

$\longrightarrow$ för mycket komplexitet är ett hot.

$\longrightarrow$ bäst med program som är små och enkla att förstå.

Öppen hårdvara och ”right to repair”

Mer generellt: viktigt vem som har kontroll över tekniken, inte bara mjukvara utan även hårdvara.

För full kontroll behövs inte bara öppen källkod till programvaran utan också *öppen hårdvara* (“open hardware”) d.v.s. öppen information, ritningar o.s.v. som visar hur allt är byggt så att man kan bygga det själv.

Licenser för öppen källkod

Två huvud-typer av FOSS-licenser:

(1) **“Copyleft”**-licenser. Exempel: GPL (GNU Public License)
Innebär att koden är fri att använda, ändra och dela med sig av, med villkoret att nya versioner av koden också ska vara FOSS. Linux-kärnan har en GPL-licens.

(2) **“Permissive”**-licenser. Exempel: MIT license
Innebär att koden är fri att använda, ändra och dela med sig av, men licensen tillåter också att man stänger in koden. Det går alltså att ta en öppen kod, ändra den lite och “äga” resultatet som en egen “proprietary software”.

Begrepp: fork

Begreppet “**fork**” betyder att någon startar en oberoende utvecklingsgren av ett FOSS-projekt.

Kan göras om man inte är nöd med riktningen huvudversionen utvecklas i, man vill göra annorlunda val.

Kan även göras om den som utvecklar huvudversionen vägrar acceptera bra ändringar som andra föreslår, då kan de andra göra en “fork” och skapa en (enligt dem) bättre version.

Begrepp: upstream/downstream

Den som har hand om “den riktiga” versionen av källkoden för ett FOSS-projekt kallas “**upstream**”.

Andra som använder FOSS-projektet och kanske gör egna ändringar är “downstream”. De kan (och bör!) skicka sina bästa ändringar och bugfixar till “upstream” så att fler får nytta av de ändringarna.

Varför företag väljer att jobba med öppen källkod

Exempel: VPP (vector packet processing), från början utvecklat av Cisco

Cisco valde att göra koden öppen. Varför?

- Lättare att underhålla
- Får mer feedback och bidrag från utvecklare utifrån
- Leder till bättre kvalitet på koden

Andra som använder VPP vill gärna skicka sina bidrag “upstream” till den publika versionen.

Ha kontroll, kunna undersöka och ändra om det behövs

Exempel, i Debian: vad gör programmet `ssh` ?

Hämta källkoden, undersök och ändra, bygg och installera:

```
apt-get source openssh-client
cd openssh*
sudo apt-get build-dep .
DEB_BUILD_OPTIONS=nocheck dpkg-buildpackage -us -uc -b
cd ..
sudo dpkg --install openssh-client_8.4*.deb
```

För att återställa:

```
sudo apt reinstall openssh-client
```

Ha kontroll, kunna undersöka och ändra om det behövs

Exempel, i Debian: vad gör programmet `ssh` ?

Hämta källkoden, undersök och ändra, bygg och installera

Demo?

Öppen källkod och jobb/rekrytering

Öppen källkod drar ofta till sig bra utvecklare. Varför?

- Roligt att bidra till något som många använder
- Mycket granskning ger hög kvalitet
- Dålig kod är oftast inte välkommen
- Stolthet över det man gjort
- Att bidra ger möjlighet att påverka

$\longrightarrow$ bra att kunna visa upp på jobb-intervju att man bidragit till FOSS-projekt!

Öppen källkod och jobb/rekrytering

Fördelar

Kunskap om och erfarenhet av FOSS-utveckling ger fördelar jämfört med att jobba i stängda miljöer.

Den som jobbar med stängda saker kan påstå att den gjort saker, men det finns inga bevis att peka på. Kanske har hen varit en dålig utvecklare, svårt att veta.

Den som jobbat med FOSS och kan visa på sina bidrag i öppna projekt har något tydligt att visa på, mer trovärdigt.

Öppen källkod och jobb/rekrytering

Fördelar, forts.

Kunskaper inom FOSS är ofta brett användbara, inte begränsat till ett visst företags system.

Om man är expert på ett visst företags stängda produkt har man en begränsad kompetens, kan bara jobba på ställen där just den produkten används.

Den som jobbat med FOSS får en mer generell kunskap, mer förståelse för hur saker fungerar, eftersom öppen källkod ger mycket bättre möjligheter till felsökning och till att lära sig saker.

Öppen källkod och jobb/rekrytering

Ännu fler fördelar

Politisk uitveckling i EU senaste åren $\longrightarrow$ större intresse för "digital suveränitet" för Europa, EU vill försöka göra sig mindre beroende av USA-bolag.

$\longrightarrow$ ökad efterfrågan på FOSS-utvecklare.

Organisationer

kring FOSS och digitala rättigheter

- Föreningen för Digitala Fri- och Rättigheter (DFRI) – `https://www.dfri.se/`
- Fripost – `https://fripost.org/`
- Free Software Foundation (FSF) – `https://www.fsf.org/`
- Free Software Foundation Europe (FSFE) – `https://fsfe.org/`
- Electronic Frontier Foundation (EFF) – `https://www.eff.org/`
- Software Freedom Conservancy – `https://sfconservancy.org/`
- European Digital Rights (EDRi) – `https://edri.org/`
- Framasoft – `https://framasoft.org/`
- Uppsala Linux User Group (ULUG) – `http://ulug.se/`

Bonus-länkar

- Årlig FOSS-konferens i Göteborg: “foss-north”: `https://foss-north.se/`
- `https://www.torproject.org/`
- `https://switching.software/`
- `https://publiccode.eu/`
- `https://www.mojeek.com/`
- `https://noyb.eu/`
- Swedish FOSS celebrity: Daniel Stenberg, author of `curl`: `https://daniel.haxx.se/`
- `https://foss-gbg.se/`
- `https://www.foss-sthlm.se/`

Tack!

Frågor?

Mejla gärna frågor senare också: `elias@glasklarteknik.se`