Open-source: o revoluţie în software?

(C) Mihai Budiu — mihaib@cs.cmu.edu, http://www.cs.cmu.edu/~mihaib

mai 1996, revizuit mai 1998

Subiect: publicarea codului sursa al programelor

Cuvinte Cheie: Free Software, Proiectul GNU, Linux

Nivel Tehnic: divertisment

Cunoştinţe necesare: programare de orice fel

Cuprins

Calculatoarele sunt o apariţie relativ recentă pe scena tehnologiilor; cu toate acestea, au devenit o componentă din ce în ce mai importantă a societăţii. Anul acesta se estimează că aproximativ 40% din produsul intern brut al Statelor Unite ale Americii provine din tehnologii înalte, ceea ce înseamnă mai ales hardware, software şi telecomunicaţii.

Importanţa fenomenului economic reprezentat de calculatoare este şi mai recentă: companiile de gen au căpătat greutate mare la bursă abia la sfîrşitul anilor ’70. Cel mai bogat om din lume este Bill Gates, patronul companiei Microsoft, al cărei capital s-a clădit cu precădere din comercializarea de software.

Programele calculatoarelor (software) reprezintă din ce în ce mai multă valoare. Dar programele sunt totodată foarte diferite de majoritatea celorlalte creaţii ale umanităţii. “Cum trebuie privite software-ul de către societate” este o întrebare al cărei răspuns nu este încă pe deplin formulat (în pofida importanţei economice fenomenale pe care o are deja!). Ca dovadă fenomenele de neimaginat în urmă cu cîţiva ani cărora le suntem martori, şi care sunt subiectul acestui articol. Mă refer cu precădere la răspîndirea paradigmei software-ului “liber”, care nu are restricţii de acces, copiere şi citire, cel mai faimos reprezentant al categoriei fiind sistemul de operare Linux împreună cu utilitarele aferente.

Partea I
Despre drept

1 Copyright, licenţe, patente şi secrete comerciale

Mecanismele juridice cu care societatea protejează software-ul sunt cu precădere adaptate de la mecanismele dezvoltate pentru a proteja alte creaţii ale spiritului uman (prin opoziţie cu bunurile materiale). De aceea vom arunca întîi o privire asupra acestor mecanisme, ca să înţelegem mai bine unele dintre dilemele contemporane legate de software. Probabil că inadecvarea acestor mecanisme “improvizate” este o cauză a tranziţiei căreia îi facem faţă.

Noţiunile pe care le discut în secţiunile următoare sunt prezentate aşa cum sunt formulate de legislaţia Statelor Unite ale Americii. Cu certitudine, această ţară este cel mai important jucător din domeniu la ora actuală, după volumul veniturilor, numărul de indivizi angrenaţi sau aproape orice altă metrică. Nu ştiu în ce măsură legislaţia din România este similară, dar cred că este oricum instructiv să aruncăm o privire la formulele americane.

1.1 Patente

Patentul este un contract între societate şi un inventator privitor la o invenţie a acestuia:

- Din partea sa, inventatorul dă societăţii toate informaţiile despre o invenţie a sa, suficient de amănunţite încît societatea să poată folosi invenţia într-o utilizare concretă.

- În schimb societatea conferă inventatorului un monopol asupra utilizării ideii: nimeni altcineva decît inventatorul nu are dreptul să aplice ideea pentru o anumită perioadă de timp; legislaţia americană garantează 17 ani de independenţă inventatorului.

Ideea este destul de naturală: dacă inventatorul nu ar fi protejat în vreun fel, ar putea veni un altcineva mai puternic, care să beneficieze de truda inventatorului, să aplice invenţia şi să strîngă toţi banii. Inventatorul are prin patent o perioadă de autonomie în care să-şi fructifice ideea, fără frica unei astfel de concurenţe.

Societatea este cîştigată, la rîndul ei, în două forme:

- După 17 ani invenţia poate fi folosită de către oricine;
- Ideea este imediat vizibilă, putînd deveni un punct de plecare pentru alte idei mai avansate.

Inventatorul are toate drepturile asupra ideii sale pentru perioada acoperită de patent; el poate decide să nu facă absolut nimic cu ideea, să încerce să o fructifice de unul singur, sau să facă o asociere cu alţi indivizi prin care să o exploateze. De pildă poate face un contract cu o firmă, prin care-şi *vinde* drepturile exclusive asupra ideii.

Societatea, prin biroul de patente, oferă inventatorului care şi-a descris ideea o dovadă de întîietate. Dacă cineva încalcă drepturile inventatorului (de pildă o firmă fabrică nişte aparate bazate pe ideea patentată), inventatorul poate da în judecată firma, iar tribunalul îi poate acorda despăgubiri proporţionale cu daunele suferite de lipsa exclusivităţii.

Primul individ care depune o înscriere la patent obţine dreptul exclusiv de folosire, chiar dacă altcineva inventează acelaşi lucru, fără a avea cunoştinţă de invenţia anterioară.

Nu orice lucru se poate patenta. De exemplu în Statele Unite nu se pot patenta idei care erau în circulaţie cu mai mult de un an de la depunerea înscrierii la patent. Nu mă pot deci trezi mîine şi patenta motorul Carnot; e prea tîrziu. Nu se pot de asemenea patenta formulele matematice sau procesele cognitive.

1.2 Copyright

Copyright-ul este o altă formă de protecţie a unei creaţii intelectuale. Copyright, literal “drept de copiere”, stabileşte că numai autorul unei creaţii are dreptul să multiplice şi distribuie acea creaţie. Drepturile acestea, în sistemul american, durează pînă la moartea autorului plus 50 de ani.

Spre deosebire de patente, copyright-ul nu protejează *o idee*, ci *modul de exprimare* al ideii. Cu alte cuvinte nu pot proteja la copiere idila între Romeo şi Julieta, dar Shakespeare ar fi putut proteja piesa cu pricina. Scenariul unui film despre Romeo şi Julieta este o cu totul altă creaţie decît piesa, şi autorul scenariului are drepturi exclusive de copiere asupra sa.

Iarăşi, spre deosebire de patente, copyright-ul apare automat. De îndată ce am fixat într-un mediu permanent expresia mea, am drept de copyright asupra ei, fie că este vorba de un text literar, un manual şcolar, o simfonie sau o pictură.

Drept de copiere exclusiv am asupra creaţiilor mele; problema este cum pot să dovedesc în faţa instanţei acest lucru, la nevoie, dacă altcineva îmi ia manuscrisele, le multiplică şi le vinde? Pentru a beneficia de astfel de protecţie pot să-mi *înregistrez* creaţia la un organism oficial; ştampila lor este dovada de care am nevoie.

Faptul că un obiect este protejat de copyright este indicat de obicei de semnul (C) (c într-un cerculeţ). Pentru înregistrările muzicale avem de-a face cu fonograme în loc de litere (phonograms), aşa că semnul folosit este (P).

Copyright-ul este mai puţin restrictiv decît patentele: alţi indivizi au dreptul să folosească fragmente din creaţia mea dacă o fac în scopuri dezinteresate şi nu îmi cauzează daune. Aceasta este aşa numita clauză *fair use*.

De exemplu un autor poate fi citat, fără ca prin asta să se încalce copyright-ul. Care este limita între "fair use" şi încalcare, tribunalele stabilesc de la caz la caz, în funcţie de circumstanţe.

Ca şi în cazul patentelor, un autor poate decide să vîndă altora o parte din drepturile sale exclusive asupra obiectului în schimbul unor compensaţii.

Anumite lucruri nu pot fi protejate de copyright; în pofida aparenţelor, titlurile (cărţilor, filmelor, produselor) *nu* pot intra în această categorie. Sloganuri, obiecte de uz general (forma lor mai exact), cărţile de telefon nu pot fi protejate de copyright. Obiectele protejate trebuie să fie într-o oarecare măsură originale. (Un slogan sau un simbol pot fi protejate însă printr-o *marcă comercială*, dar asta e altă poveste care ne interesează mai puţin.)

1.3 Domeniul public

Ce se întîmplă după expirarea perioadei de validitate a unui patent sau copyright? Bunurile protejate intră în *domeniul public*: oricine are voie să le folosească fără a fi îndatorat cuiva. Sunt bunuri ale întregii societăţi.

Astfel, piesele lui Shakespeare pot fi jucate pe scenă de către oricine, fără ca prin aceasta să fie încălcate drepturile sale de copyright.

1.4 Licenţe

Licenţa este un contract prin care posesorul unor drepturi exclusive vinde o parte din aceste drepturi altcuiva. Cele două părţi în mod normal semnează un contract, prin care se specifică întinderea transferului (de exemplu un autor poate licenţia o carte pentru a se face o piesă după ea, dar nu pentru un film).

1.5 Secrete comerciale

Cînd ai pus mîna pe o găselniţă cu potenţial ai două posibilităţi:

- O patentezi şi eşti apărat de lege în monopolul tău;
- Ţii ideea secretă şi o "storci" cît poţi. Ideea devine atunci un *secret comercial*.

Secretele comerciale au avantaje şi dezavantaje faţă de patente. Un avantaj este că atîta vreme cît reuşeşti să păstrezi secretul, beneficiezi singur de el, pe cînd un patent expiră după o vreme; de pildă compoziţia băuturii "Coca-Cola" este un secret comercial păzit cu grijă de mai bine de un secol.

Un dezavantaj este că oricine poate — teoretic — avea aceeaşi idee şi o poate patenta. Atunci adio avantaje, ba chiar dimpotrivă.

Legea apără într-o oarecare măsură secretele comerciale. Anumite activităţi, ca spionajul, sunt ilegale tocmai din astfel de motive. Companiile care au secrete comerciale se apără de proprii lor angajaţi folosind contracte speciale numite *Non-Disclosure Agreements, NDA*: înţelegeri de ne-divulgare. Practic angajatul semnează cu compania un act cum că nu va divulga nimănui informaţiile secrete, etc., iar compania îl poate da în judecată pentru încălcarea contractului dacă o face. (Pe de altă parte odată secretul divulgat e bun dus; "trădătorul" va plăti atunci daune.)

2 Situaţia în software

După atîtea ocolişuri "legale" să vedem care este situaţia în domeniul care ne interesează — în software.

Ei bine, toate ingredientele de mai sus apar şi în software. Astfel:

- Algoritmii se pot patenta (dar nu programele!); un exemplu este algoritmul de criptografie RSA, care este patentat de autorii săi; ei sunt singurii care au dreptul să-l implementeze (patentul va expira în curînd);
- Programele sunt protejate prin copyright;
- Multe firme decid să facă tehnologiile folosite secrete comerciale (atît programele cît şi algoritmii); astfel de firme semnează cu angajaţii lor NDA-uri foarte stricte;
- Algoritmi şi programe pot fi licenţiate (între creatorii lor şi firme); de exemplu sistemul de operare Unix a fost licenţiat multor firme de către AT&T; cea mai notabilă este Sun Microsystems, al cărei sistem de operare Solaris este construit pe baza unor bucăţi de cod licenţiate de la AT&T (între timp AT&T a vîndut drepturile sale exclusive şi marca comercială Unix lui Novell, care le-a vîndut apoi lui SCO).

Să observăm că (lucru valabil şi pentru patente şi copyright în general) o firmă poate angaja indivizi care să inventeze/creeze pentru ea, iar produsele devin posesiunea firmei, şi nu a indivizilor care le-au creat. Acest lucru este de obicei specificat în contractele de muncă ale inşilor. Majoritatea firmelor software aplică acest model.

2.1 Tipuri de protecţie legală a creatorilor de software

Varietatea de protecţii aplicate în software este însă foarte mare, în parte şi pentru că un program poate fi văzut la două nivele: al codului sursă, în limbajul în care este scris (C, Pascal, Java, etc.), sau la nivelul programului *obiect, executabil*, aşa cum iese dintr-un compilator. Iată unele dintre categoriile cele mai răspîndite de software[1]:

Public domain: sunt programe care nu sunt supuse nici unei protecţii, pentru că autorii lor nu au luat măsurile necesare de protecţie. Oricine poate folosi aceste programe în orice fel, inclusiv le poate include în programe mai mari care sunt secrete sau sunt protejate de copyright.

[1]Descrierile de aici sunt adaptate din pagina de web a proiectului GNU, despre care voi vorbi mai jos: `http://www.gnu.org`; reproducerea este permisă de copyright-ul din acea pagină.

Proprietary: (în proprietatea cuiva) sunt programe a căror modificare, folosire şi redistribuire sunt supuse la restricţii de către posesorii lor. Semi-free software, de mai jos, intră în această categorie.

Semi-free: anumite programe sunt disponibile fără constrîngeri dacă sunt folosite în scopuri non-profit, dar sunt supuse la restricţii pentru alte utilizări. Un exemplu este pachetul PGP (Pretty Good Privacy).

Shareware: shareware este un tip de program care se poate folosi într-o formă restrînsă (fără funcţionalitate completă, sau doar pentru o anumită vreme) gratuit, dar care trebuie cumpărat în versiunea completă. Practic un fel de software cu mostre gratuite pentru a momi cumpărătorii. În general "shareware" este o formă de software "proprietary".

Freeware: acest termen este ambiguu şi nu se bucură de o interpretare unanim acceptată.

Comercial: software-ul comercial este vîndut de o companie pentru bani. În mod paradoxal, software-ul comercial nu este acelaşi lucru cu software-ul "proprietary"! Poate exista software comercial free, cum explică următoarea categorie!

Free/Open Source: aceasta este de fapt creatura cea mai interesantă, care este scopul central al acestui articol. Software free este orice software care:

- Este distribuit fără restricţii de copiere sau utilizare pentru oricine;
- Este întotdeauna distribuit (şi) în limbajul sursă;
- A cărui *modificare* şi distribuire sunt permise oricui.

Termenii "open source" şi "free software" sunt sinonimi. Cel de-al doilea este mai tradiţional, dar puţin mai confuz, pentru că "free" înseamnă în engleză atît "liber" (de restricţii) cît şi "gratuit". Free software *nu* înseamnă gratuit, ci "liber de restricţii".

2.2 Garanţiile software-ului

E interesant de văzut ce fel de garanţii oferă firmele în general atunci cînd livrează programe, cel puţin în cazul consumului de masă (ce se întîmplă cu software-ul scris pentru a monitoriza centrale nucleare este o altă poveste).

Cînd cumperi soft de la Microsoft (sau orice alt mare producător) vei avea surpriza să observi că *nu ţi se oferă nici o garanţie* asupra calităţii sale! Dacă aţi avut răbdarea să citiţi un astfel de contract, care de obicei apare cînd porneşti prima oară un program, aţi observat că firma producătoare nu promite că programul face ceva bine cu siguranţă 100%.

Desigur, programele nu sunt în realitate inutile, pentru că nu le-ar cumpăra nimeni. Ele nu sunt însă nici complet fiabile. În general merg, însă producătorul nu este răspunzător de nici o malfuncţie.

Situaţia programelor este complet diferită de cea a tuturor celorlalte bunuri de larg consum; o fabrică de maşini este *obligată* să garanteze ca motorul nu poate lua foc (desigur, în condiţii rezonabile).

Această diferenţă de atitudine este posibilă pentru că în general software-ul nu are un impact imediat asupra lumii materiale: ţi se blochează calculatorul, îl scoţi din priză, îi dai drumul, poate ai pierdut ceva din munca ta, dar merge din nou. Complexitatea programelor a ajuns atît de mare încît este virtual imposibil să fie garantată corectitudinea lor, nici măcar din punct de vedere statistic. O viaţă de om nu ajunge pentru a citi toate cele 20 de milioane de linii sursă ale sistemului de operare Windows NT, cu atît mai puţin pentru a garanta corectitudinea lor.

Este pe deplin acceptat că fiabilitatea unui program este funcţie (crescătoare) de vechimea lui: cu cît un program a fost folosit mai mult şi peticit mai tare, cu atît este mai sigur. Din această cauză marile bănci rulează în continuare software scris prin anii '70, scris în Cobol: riscul unei pierderi este prea mare pentru a încerca o altă soluţie.

Partea II
Free Software

Noţiunea de free-software (sau open-source) este aparent paradoxală: dacă oricine poate lua un program şi-l poate multiplica, cum poate acel program să mai poarte valoare pentru cel care l-a creat? Cum se pot face bani scriind program pe care apoi oricine le poate folosi?

Orice economist o să exclame: "nu se poate"! Absurd! Aşa ceva nu există.

Bine, dar atunci cum explicăm cele 7 milioane de servere din lume pe care sunt instalate Linux?

Pentru a vedea geneza acestui fenomen misterios trebuie să facem un pic de istorie.

3 Apariţia valorii în software

La început a existat numai hardware. La începutul anilor '60 calculatoarele erau mari, puţine şi extrem de scumpe (milioane de dolari). Timpul lor era foarte preţios, şi cei care ştiau să le "descînte" erau puţini.

Cu vremea hardware-ul a devenit din ce în ce mai ieftin; "încrucişarea" s-a produs undeva la începutul anilor '70, cînd oamenii au realizat că programele care se execută pe computere sunt cel puţin la fel de importante economic ca şi computerele însele.

Dacă programele au valoare economică atunci ele trebuie protejate, nu? Cum? Făcîndu-le secrete comerciale. Practica ne-publicării surselor programelor şi interzicerii multiplicării datează din această perioadă.

Aparent modelul funcţionează de minune: companiile care scriu software fac o grămadă de bani din asta; programatorii sunt în mare cerere şi salariile lor sunt impresionante. Costul de producţie al software-ului nu este deloc scăzut: randamentul unui programator este destul de mic. Cea mai mare parte a efortului este cheltuită în *întreţinerea* şi *depanarea* programelor. Microsoft are pentru fiecare programator doi indivizi care testează! Ceea ce face software-ul profitabil este uşurinţa cu care este multiplicat. Un program popular este scris o singură dată (şi întreţinut cu un efort continuu), dar este vîndut în milioane de copii.

O altă mare sursă de venit al companiilor de software este "customer support". Compania oferă o formă limitată de asistenţă tehnică utilizatorilor săi, răspunzînd la întrebări şi ajutîndu-i în anumite proceduri (service). Compania nu promite, dar încearcă, să corecteze malfuncţiile programelor.

Deşi acest statut juridic/economic al software-ului funcţionează de minune, poate de fapt nu este cel mai potrivit. Obiecţii pertinente se ivesc una cîte una.

4 Contracurentul: Richard Stallman şi proiectul GNU

Cea mai importantă figură a opoziţiei noţiunii de "proprietary software" (software secret, cu restricţii de publicare şi copiere) este cu certitudine Richard Stallman. Dacă nu este cea mai importantă figură, Stallman este cu siguranţă cea mai pitorescă.

Stallman este un "hacker" cu o experienţă impresionantă: a început să programeze încă din anii '60. Stallman a lucrat o vreme îndelungată la Universitatea MIT (Massachusetts Institute of Technology[2]) ca programator de sistem în laboratorul de Inteligenţă Artificială. După aceea şi-a dat demisia ca să se consacre în întregime proiectului său GNU despre care o să vorbesc ceva mai jos.

Cea mai răspîndită creaţie a lui este editorul de texte Emacs, pe care l-a conceput cît timp era la MIT. Astăzi Emacs este probabil unealta preferată pentru interacţiunea cu calculatorul a majorităţii profesioniştilor din calculatoare din lume. (Un articol despre Emacs am publicat în PC Report din mai 1997; articolul este disponibil din pagina mea de web.)

De asemenea Stallman este arhitectul şi realizatorul excelentului compilator de C numit GNU C Compiler, pe scurt GCC. Este singurul compilator care, ca şi Emacs dealtfel, funcţionează pe cîteva zeci de maşini diferite arhitectural (de la PC-uri şi staţii de lucru la supercalculatoare).

Stallman este însă extraordinar prin proiectul său GNU, şi prin fundaţia pe care a creat-o pentru a-l duce la capăt. Despre proiectul GNU puteţi afla o sumedenie de lucruri din pagina sa de web de la `http://www.gnu.org`.

Stallman este cel puţin un excentric. Cei care l-au văzut cu ocazia vizitei sale la Bucureşti în 1994, la conferinţa ROSE, nu-l vor uita prea curînd. Despre raţiunile sale şi concepţia sa despre programare puteţi afla din paginile mai sus citate; excelent reprezentativ este textul unei conferinţe ţinute în 1986 la Institutul Regal de Tehnologie din Stockholm.

Pe scurt, Stallman crede că a face software-ul secret şi a împiedica diseminarea sa este imoral, pentru că privează societatea de accesul la valoare. Mai mult decît atît, acest lucru înseamnă risipă: pentru că nu poţi copia (şi îmbunătăţi) software, trebuie să-l rescrii. Asta se vede destul de clar în sumedenia de sisteme de tip Unix care există: fiecare firmă mare a dezvoltat propria ei versiune, uşor incompatibilă cu toate celelalte. (Un articol despre istoria sistemului de operare Unix şi variantele sale am publicat în BYTE din august 1996; vedeţi şi pagina mea de web.)

Stallman spunea:

> Pentru ca astfel de lucruri să nu se mai poată întîmpla m-am hotărît să scriu un set complet de software de sistem, pe care să îl pot da oricui îl vrea să îl folosească. [Pachetul ăsta de programe poartă denumirea de GNU]. Odată ce GNU va fi scris, oricine va fi capabil să obţină software de sistem de calitate, fără a face vreun efort.

Nu-i aşa că sună nebuneşte?

Proiectul GNU a debutat în 1984. GNU urma să fie un set complet de programe necesare pentru a utiliza un sistem de tip Unix, plus un nucleu de sistem de operare stil Unix. Iar totul urma să fie "free software", spre deosebire de Unix, care este o marcă comercială.

[2]MIT este considerată liderul universităţilor americane în cercetare în calculatoare.

Numele este o abreviere recursivă, (Stallman este un mare programator în Lisp): GNU = Gnu is Not Unix (G este prescurtare la rîndul său de la Gnu).

Programele din sistemul GNU însă încearcă să remedieze unele din deficienţele tradiţionale de implementare a UNIX-ului (în multiplele variante ale diferitelor firme), păstrînd multele sale calităţi.

Stallman a înfiinţat o societate non-profit, Free Software Foundation (FSF), pentru a duce la bun sfîrşit proiectul GNU.

Primele programe scrise de Stallman pentru proiectul GNU au fost GCC, o implementare complet "free" a editorului Emacs, şi un generator de parsere Bison. În timp proiectul GNU a dezvoltat majoritatea utilitarelor necesare oricărui sistem Unix tradiţional. Şi într-adevăr, acestea sunt excelent implementate, cu mai puţine bug-uri şi eficienţă mai mare decît cele pe care se dau bani grei. Toate aceste programe pot fi copiate gratuit prin Internet sau pot fi cumpărate de la FSF.

În 1996 a fost lansată şi prima versiune a nucleului sistemului de operare GNU, numit "The Herd" (ciurda). Din raţiuni pe care le vom evidenţia mai jos (recte popularitatea Linux), The Herd nu este încă foarte populară.

FSF angajează programatori care scriu pentru proiectul GNU. Cele mai multe din programele GNU însă sunt rodul muncii unor entuziaşti din întreaga lume, care scriu programe pe care le pun la dispoziţia fundaţiei, sau care contribuie cu optimizări sau indicaţii de bug-uri la programele existente. Fundaţia scrie şi documentaţii şi manuale pentru produsele ei, pe care le tipăreşte şi le vinde. Periodic face CD-ROM-uri pe care ultimele versiuni ale programelor GNU se află în totalitate, şi care pot fi cumpărate la preţuri derizorii. Fundaţia mai trimite cui vrea contra cost benzi cu programe. Acestea sunt în principiu mijloacele ei de subzistenţă.

Stallman s-a aşteptat şi la suport financiar din partea unor corporaţii mari care dezvoltă soft pentru Unix:

> Mă aşteptam ca toţi să mă sponzorizeze pentru ca eu să le fac soft de excelentă calitate pe care să-l folosească toţi, iar ei să se poată ocupa de lucruri mai serioase, decît să refacă fiecare acelaşi lucru în felul lui.

În viziunea lui Stallman, cea mai importantă calitate a software-ului este că poate fi citit de oricine şi îmbunătăţit de oricine. Or asta se poate numai dacă software-ul este "free".

Free înseamnă două lucruri esenţiale independente:

- Programele pot fi copiate de către oricine şi distribuite;
- Programele pot fi citite de către oricine şi modificate.

Desigur, acestea sunt atribute adevărate ale oricărei creaţii care este în domeniul public: oricine o poate copia, modifica şi distribui. Dar asta nu e suficient pentru a o face utilă: ce folos dacă cineva ia un program public, îl îmbunătăţeşte şi apoi îl face secret?

Acesta este pericolul de care trebuie să ne apărăm. Dar cum?

4.1 GNU Public License (GPL)

Pentru a proteja soft-ul pe care FSF îl scrie, acesta nu este pus pur şi simplu în domeniul public, unde oricine şi l-ar putea însuşi pentru scopuri personale. Soft-ul GNU este protejat de exact mijlocul pe care marile firme îl folosesc pentru a-şi apăra produsele împotriva copierii: copyright-ul şi licenţele.

Stallman e elaborat celebra GPL, "General Public License", care se aplică programelor GNU. Programele GNU sunt deci protejate de copyright, dar licenţa GPL dă următoarele drepturi utilizatorilor:

- Oricine are voie să folosească, copieze, modifice şi distribuie (eventual pentru bani) soft GNU;
- Oricine construieşte programe folosind bucăţi din programe GNU *trebuie* să ofere aceleaşi drepturi tuturor celorlalţi (copiere, distribuire, etc.).

Pentru Stallman, a face cunoştinţele "secrete" este profund imoral. El observa, corect, că existentă suportului tehnic şi a garanţiilor nu are absolut nici o legătură cu proprietatea software-ului de a fi secret sau "free":

> Utilizatorii cumpără software de firmă pentru că li se spune că vor căpăta asistenţă tehnică şi garanţii. Este desigur un lucru foarte util, dar dacă se vor gîndi ce legătura poate să aibă asta cu vînzarea de soft, sau cu interdicţia de a-l studia, vor realiza că nu e prea clar. Nu există nici o garanţie ca soft-ul de firmă va primi suport tehnic bun.
>
> [...] În realitate lumea crede că GNU Emacs este mai bine întreţinut decît variante comerciale ale programului. Unul dintre motive este eu că sunt un hacker mai bun decît cei care scriu celelalte Emacs-uri, dar celălalt motiv este că toată lumea are sursele şi sunt aşa de mulţi interesaţi să-l folosească la tot felul de lucruri încît de fapt nu trebuie să fie întreţinut de mine.

Acesta este un lucru deosebit de interesant, asupra căruia o să revenim mai jos, în secţiunea consacrată modelului "open-source".

5 Linux

5.1 Geneza

În 1987 un profesor de la Universitatea Vrije din Amsterdam, pe nume Andrew S. Tanenbaum,[3] scrisese o carte excelentă despre arhitectura sistemelor de operare, care era însoţită de codul unui nucleu complet, numit Minix. Minix arăta pentru un utilizator ca un Unix foarte mic (de aici şi numele), dar era construit folosind o arhitectură complet deosebită de cea a Unix-ului tradiţional (Minix era un micro-nucleu).

Minix era distribuit în categoria "semi-free" de mai sus, adică fără restricţii pentru cei care-l foloseau non-profit. Codul era folosit pentru instruirea în sisteme de operare în universităţile din întreaga lume. Grupurile de discuţii pe Internet (news) despre arhitectura nucleului, unde lumea sugera variante şi îmbunătăţiri, se bucurau de un interes crescînd.

În 1991 proiectul GNU pornise deja de 6 ani şi dezvoltase toate utilitarele necesare, mai puţin nucleul unui sistem de operare. Utilitarele GNU erau folosite de multă lume, dar nimeni nu avea încă la dispoziţie ceea ce Stallman îşi dorise: un sistem complet free, de la un cap la altul.

[3] Da, este acelaşi Tanenbaum care a scris cartea de Reţele de Calculatoare publicată de Editura Agora în limba română.

Sudura s-a produs datorită unui student finlandez de la universitatea din Helsinki. Linus Torvalds a adăugat iniţial un modul de memorie virtuală la Minix şi şi-a făcut publică creaţia. Foarte mulţi entuziaşti au început să folosească îmbunătăţirile aduse de Linus.

Linus a început încetul cu încetul şi a rescris feluritele părţi din Minix folosind o arhitectură monolitică[4]. Noul sistem era dezvoltat iniţial folosind Minix (compilatoarele, utilitarele pentru scrierea sistemului se executau pe Minix). A urmat o ceartă publică (pe Internet) între Linus şi Tanenbaum, care era nemulţumit de evoluţia arhitecturii nucleului său. Tanenbaum este un cercetător renumit în sistemele de operare distribuite, şi crede că arhitectura micro-nucleu este mai potrivită pentru evoluţia sistemelor viitoare. Linus pe de altă parte s-a bazat foarte mult pe arhitectura Unix-ului tradiţional (despre care s-a publicat foarte mult, şi care este relativ bine înţeleasă; trebuie reţinut că design-ul Unix-ului tradiţional nu este patentat; codul său este sub copyright, dar asta nu înseamnă deloc că altcineva nu poate scrie noi sisteme care să arate ca Unix). În momente de înfierbîntare Tanenbaum a declarat că Linus ar pica dacă ar fi student la cursul său de sisteme de operare.

Linus atunci şi-a luat jucăriile şi a plecat, redenumind propriul lui nucleu Linux[5]. După o vreme Linux avea un nucleu suficient de complet încît să nu mai fie nevoie de mediul de dezvoltare oferit de Minix. În locul utilitarelor de pe Minix, Linus a ales să folosească sculele de dezvoltare de software de la GNU. De asemenea, Linus a pus codul Linux sub protecţia licenţei GPL, făcînd codul "free".

Aşa a început ceea ce ar putea fi "noua revoluţie în software"...

5.2 Explozia

Linux a început să strîngă tot mai mult aderenţi, în dauna Minix-ului. Unul dintre motive era entuziasmul cu care Linus accepta modificările sugerate de alţii şi le introducea în codul lui (Tanenbaum era mult mai conservator). Un alt motiv era faptul că Linus a implementat anumite trăsături foarte utile unui sistem de operare, cum ar fi memoria virtuală. (Tanenbaum a spus că aşteaptă vremea cînd RAM-ul o să fie atît de ieftin încît nu va mai fi nevoie de memorie virtuală.)

Cert este că evoluţia numărului de utilizatori a fost explozivă, exponenţială. În 1995 cînd Linus a fost la Bucureşti la conferinţa ROSE a prezentat un grafic care arăta cam aşa:

An	Număr de utilizatori Linux
1991	50
1992	500
1993	5.000
1994	50.000
1995	500.000

De atunci încoace numărul de utilizatori a continuat să crească spectaculos. Cifra actuală (1998) este estimată undeva între 4 milioane şi 24 de milioane, dar este imposibil de afirmat cu precizie cîţi. Unul dintre motive este că Linux este distribuit "free", şi oricine poate să-l citească de pe

[4] Am publicat un întreg articol despre lupta dintre micronuclee şi monoliţi într-un PC Report din septembrie 1997.

[5] Cronologia mea este puţin imprecisă privitor la aceste evenimente "primordiale"; ele s-au petrecut cu siguranţă, dar nu sunt sigur de ordinea lor.

Internet, fără a da socoteală cuiva. Nu se pot deci contabiliza încasările pentru a afla popularitatea sa.

Sistemul a crescut din ce în ce mai mult, şi a căpătat din ce în ce mai multă funcţionalitate. Modelul de dezvoltare este în esenţă acelaşi ca cel din urmă cu 7 ani: Linus menţine un control relativ strîns al arhitecturii interne a sistemului şi proiectează interfeţele. Contribuţiile unor programatori din întreaga lume sunt înglobate în nucleu şi rapoartele despre erori ale celor care testează sunt imediat luate în considerare.

În clipa asta Linux este pe locul doi în lume ca număr de sisteme de operare multi-utilizator instalate (nu socotim Windows95, DOS şi alte sisteme mono-utilizator), după Windows NT al lui Microsoft[6]. Linux a bătut toate Unix-urile comerciale ca popularitate (care există de 25 de ani)! E natural la preţul pe care îl are...o distribuţie Unix de firmă costă între cîteva sute şi cîteva zeci de mii de dolari, depinzînd de utilitarele livrate.

Linux mai deţine şi alte recorduri interesante: este cel mai portabil nucleu de tip Unix, care rulează nu numai pe PC-uri cu procesoare din familia Intel Pentium, ci şi pe calculatoare Sun cu procesoare SPARC, calculatoare Digital cu procesoare Alpha, pe procesoare PowerPC de la IBM şi Motorola, Motorola m68k, MIPS de la Silicon Graphics, ARM de la Intel (fost la Digital), etc.

Linux a devenit un sistem de operare redutabil, înzestrat cu toată funcţionalitatea necesară dezvoltării de software, rulării de programe pentru servicii Internet (servere Web, de poştă electronică, etc.), cercetare, etc. Nucleul este în întregime supus licenţei GPL, ca şi majoritatea utilitarelor sistemului.

Pe de altă parte din ce în ce mai multe firme încep să dezvolte software care merge (şi) pe Linux; grupul de discuţii `comp.os.linux.announce` este foarte populat cu noi anunţuri de software, cel mai adesea software free. Anumite programe esenţiale folosite nu sunt neapărat supuse licenţei GPL, cum ar fi sistemul grafic de ferestre X Windows. Acesta este "aproape free": cei care l-au scris îl oferă gratuit în surse, însă oferă posibilitatea dezvoltării de versiuni "proprietary" (şi există astfel de versiuni).

Din ce în ce mai multe companii adoptă Linux pentru a rula servicii care au nevoie de fiabilitate (Windows NT este din punctul ăsta de vedere mult inferior). Soluţia server-ului de Web Apache peste un nucleu Linux este ideală din punctul de vedere al unui administrator de sistem. Multă publicitate s-a făcut în jurul folosirii sistemului Linux de către compania Digital Domain pentru a realiza efectele speciale din filmul Titanic. Avînd de ales între Windows NT, IRIX (de la Silicon Graphics) şi Linux a fost preferat cel din urmă. NASA plăteşte programatori care dezvoltă software pentru a transforma Linux într-un mediu pentru programare distribuită (proiectul Beowulf). Poşta Americană foloseşte 900 de staţii cu Linux pentru a face recunoaşterea scrisului de pe plicuri şi cartarea automată. Şi exemplele de aplicaţii "serioase" ar putea continua. Universităţile găsesc Linux un sistem din ce în ce mai tentant: e ieftin, e public, e bine întreţinut, este răspîndit şi are o bază largă de aplicaţii. Plus faptul că sursele sunt complet disponibile îl face un vehicul ideal pentru cercetare: cînd faci cercetare pe nuclee obţinute semnînd NDA (non-disclosure agreements) anumite rezultate nu pot fi publicate.

[6] Cineva observa că cifrele lui Microsoft pentru NT includ toate PCurile care sunt vîndute cu NT, chiar dacă după aceea utilizatorii aleg să şteargă NT şi să pună Linux.

6 Modelul Open-Source

Aşa ceva este imposibil, ştie toată lumea. Microsoft este o firmă cu un buget uriaş, mii de programatori şi testeri, care investeşte enorm în dezvoltarea de software şi cercetare, cu o experienţă remarcabilă în domeniu, care angajează oameni plătiţi full-time pentru a scrie programe, şi nu a fost în stare să producă un sistem de aceeaşi calitate. De unde atunci valoarea din Linux? Cine produce valoare dacă recompensa este aparent imposibilă datorită modelului de distribuţie liber, şi dacă restricţiile la copiere nu împiedică “furturile”?

Sau poate software-ul are o natură complet diferită, şi criteriile despre conservarea valorii din alte domenii nu i se aplică?

De fapt Linux nu este decît exemplul cel mai evident dintr-o suită întreagă de alte exemple. Modelul acesta de dezvoltare este mai răspîndit decît pare.

Unul dintre dezvoltatorii de software “free”, Eric Raymond a încercat să explice modelul de supravieţuire şi înflorire al Linux-ului dintr-o perspectivă cît mai largă: economică, tehnică, socială. Un articol de-al său pe această temă a devenit faimos: “Catedrala şi Bazarul”. Dl. Raymond contrapune în acest text (`http://www.earthspace.net/~esr/writings/cathedral-bazaar`) modelul de dezvoltare “tradiţional” (catedrala) cu modelul Linux (bazarul) şi încearcă să explice meritele celui din urmă. Acest articol a fost foarte important pentru a conceptualiza fenomenul.

Pe data de 6 martie 1998 s-au întîlnit în Silicon Valley cîţiva dintre cei mai importanţi actori din arena software-ului “free”. S-a putut observa cu această ocazie că o mare parte din software-ul care constituie *infrastructura* Internet-ului este de fapt free! (Conferinţa de presă era disponibilă la `http://www.news.com/News/Item/0,4,20913,00.html?st.ne.fd.mdh`.) Iată cei mai importanţi participanţi:

Linus Torvalds, creatorul **Linux**;

Larry Wall, creatorul limbajului **Perl**, folosit în majoritatea serverelor de Web din lume;

Brian Behlendorf, fondator al grupului **Apache**, al căror server web free rulează pe mai mult din 50% din serverele din Internet;

Eric Allman, creatorul programului **sendmail** care asigură livrarea poştei electronice în mai mult de 70% din Internet;

Paul Vixie, creator al programului **bind** (Berkeley Internet Name Daemon), piesa centrală din managementul numelor din Internet;

Phil Zimmermann, creatorul pachetului de criptografie Pretty Good Privacy **PGP**;

John Gilmore, întemeietor al firmei **Cygnus Solutions**, prima companie care trăieşte exclusiv din întreţinerea şi vînzarea de software “free”;

Guido van Rossum, creator al limbajului **Python**;

John Ousterhout, creator al interpretorului **Tcl** folosit pentru dezvoltarea rapidă de interfeţe cu utilizatorii şi generarea de pagini de web;

Tim O'Reilly, patron la editura **O'Reilly & Associates**, care nu scrie software free, dar face bani grei publicînd cărţi despre software free;

Tom Paquin şi Jamie Zawinski, reprezentînd site-ul `mozilla.org` al companiei **Netscape**[7].

Unul dintre rezultatele întîlnirii a fost propunerea de a schimba numele ambiguu de "free" software (care poate sugera "gratuit", ceea ce am văzut că nu e adevărat) cu "open-source". Eric Raymond a înregistrat marca comercială **Opensource** şi a înfiinţat un "site" web consacrat acestui subiect; îl puteţi vizita la `http://www.opensource.org`. Acest nume a fost ales pentru că cea mai importantă caracteristică a software-ului de acest gen este că trebuie publicat în "surse" (în limbajul în care a fost scris).

În acest "site" Eric Raymond expune o serie de motive foarte convingătoare care încearcă să explice succesul modelului de dezvoltare open-source. Argumentele încearcă să fie justificate din cît mai multe puncte de vedere (în contrast cu viziunea lui Stallman, pentru care prerogativele morale erau dominante: "nu e cinstit să ascunzi software", spunea el).

O să trec în revistă argumentaţia sa pe scurt; pentru detalii parcurgeţi paginile de web indicate.

6.1 Meritele "open-software"

Iată o serie de motive pentru care software-ul poate fi dezvoltat (mai bine) în stil "liber" decît ca secret comercial. Care dintre ingrediente este majoritar, rămîne de văzut.

1. "Dacă ai destui ochi, orice "bug" devine evident". Altă formulare a acestui dicton este "depanarea se poate face în paralel". Cu cît mai mulţi inşi utilizează un program cu atît e mai simplu de depistat ce nu merge bine. Atunci cînd utilizatorii au la dispoziţie sursele programului pot investiga cauzele reale ale erorii, şi adesea le pot înlătura fără intervenţia autorului programului.

2. Linux a introdus un model de dezvoltare software cu reacţie extrem de rapidă; Linus pune în circulaţie o versiune nouă de nucleu odată la 2-3 zile (cîteodată apar două nuclee pe zi!). În felul acesta reparaţiile şi sugestiile sunt foarte rapid implementate.

3. Fiabilitatea programelor free este mult mai bună decît a celor comerciale, datorită faptului că practic oricine poate căuta şi depana bug-uri. Două studii prezentat la conferinţa "Usenix" a dezvoltatorilor de Unix din 1990, respectiv 1995[8], comparau cantitativ robusteţea utilitarelor din mai multe sisteme de operare comerciale, a celor produse de GNU şi a celor folosite cu Linux (cele din urmă se suprapun). Studiile observa două lucruri oarecum surprinzătoare:

   - Robusteţea programelor "free" era mult mai mare: cînd erau "stresate" cu date de intrare incorecte, în medie 9% din programele de pe Linux şi 6% din utilitarele GNU manifestau erori. Sistemele Unix comerciale se plasau între 12% (IRIX) şi 43% (NeXT)!
   - Anumite bug-uri erau prezente atît în versiunile din 1990 cît şi din 1995 ale programelor comerciale, în pofida raportării lor (publice!) de către primul studiu!

[7] Ca urmare a lobby-ului dl-ui Raymond în sprijinul software-ului "free", compania Netscape a decis să facă publice sursele celebrului lor browser de web, devenind astfel cel mai important (din punct de vedere economic) susţinător al modelului open-source.

[8] Studiul din 1995 este disponibil din pagina de web de la GNU.

Asta demonstrează că firmele nu sunt de fapt cu adevărat capabile sau interesate să înglobeze reparaţiile sugerate de utilizatori, lucru mai puţin adevărat în cazul programelor "free". Atunci cînd codul este secret *numai* firma care îl posedă îl poate repară; un program public poate fi reparat de către oricine se pricepe.

4. Contrar aparenţelor, se pot cîştiga bani din dezvoltarea de software "open". Există mai multe companii de succes care manipulează numai software "free"; pe piaţa Linux una dintre cele mai celebre este RedHat software, care vinde CD-ROM-uri, documentaţie şi asistenţă tehnică cu o cantitate mare de software pentru Linux. Deşi produsele software RedHat pot fi copiate direct de pe reţea de către oricine, gratuit, compania a vîndut peste 400.000 de CD-ROM-uri cu Linux, pentru mai multe platforme.

5. O observaţie foarte interesantă se impune despre natura creaţiei de software: mai mult (cel puţin pe scara temporală) decît activitate creatoare, scrierea de software este întreţinere şi depanare! Cu alte cuvinte, a livra software de calitate seamănă mai mult cu prestarea de servicii pentru clientul tău. Dar aceste din urmă activităţi sunt cu siguranţă ajutate de prezenţa codului sursă şi de o bază largă de utilizatori (care sunt atraşi de disponibilitatea software-ului).

6. Rata de inovare este mult sporită de prezenţa software-ului free; o soluţie odată descoperită este disponibilă tuturor, şi pe baza ei se poate construi o alta şi mai eficace.

7. O motivaţie de natură socială: scriind software "free" poţi deveni faimos. Cîtă lume ştie numele arhitectului şef al sistemului de operare NT (David Cutler)? Şi cîtă lume a auzit de Linus? O companie mare "înghite" individualităţile angajaţilor ei.

Ce ne rezervă viitorul în privinţa modelelor de dezvoltare software este foarte greu de prevăzut. Tot ce putem face este să aşteptăm şi să vedem.

Ah, şi să contribuim la îmbunătăţirea unor programe "free" pe care le folosim!