

Nyílt forrás az üzletben?

Nagyon sokszor írunk nyílt forrású fejlesztésekről – a Nyugatról, hogy ott mennyire hatékonyan oldanak meg komoly feladatokat nyílt eszközökkel, a fejlett országok minisztériumairól, ahol – gyakran biztonsági okokra hivatkozva – a nyílt rendszereket részesítik előnyben, akár a tengeren innen, akár túl. Mégis úgy érzem, idehaza még mindig megdöbbenően kevesen gondolkodnak „nyíltan”. Tényleg nincs létjogosultsága ennek a világszerte ünnepeelt és használt elméletnek kis hazánkban?

Pontosítsuk egy kicsit, mire gondolok. Ahogy viccesen mondani szokták, két-féle ember van: az egyik, aki mindent két külön csoportba oszt, és a többiek. Komoly ellentét alakult ki a nyílt és a zárt rendszerek pártolói között, már-már kívülről tudjuk mindkét oldal először felhozott érveit. A zárt programok követői szerint például: a zárt rendszer egy kézben összefogott fejlesztés, ha a cég vásárol egy könyvelőprogramot, elvárja, hogy a program ilyen és ilyen szabályoknak feleljen meg, megfelelő leírást, támogatást vár, ha nem, visszakövetelheti az árát, pereskedhet. A zárt rendszer kódjába nem lát bele senki, a hozzáértők készítik el, és nem is kívánatos, hogy a felhasználók belenyúljanak. A zárt rendszerhez a követést az eladó készíti, egy-egy termék mögött gyakran egy egész vállalat áll.

Természetesen a nyílt oldal is felsorolja gyorsan a legtöbb említett témát, de pont a másik oldalról megvilágítva: a zárt rendszer kulcsa egyetlen kézben van, így a vásárló kiszolgáltatja magát a programozónak. Ha a rendszer nem tud valamit, és a beszállító tojik a fejére, a vásárló futhat a kifizetett nagy kalap pénze után, akár pereskedhet is. A zárt rendszer kódjába nem lát bele senki – lehet, hogy akik készítették, beleírtak még pár sort, hogy minden hónap tizedikén biztosítva legyen a havi fizetésük. Netán következetesen néhány hibát is ejtettek, hogy a vevőnek a követés megvásárlására is legyen oka. A zárt rendszert készítő cég közvetlen érdeke, hogy sokat keressen, és nem az, hogy a felhasználó minél gyorsabban, kényelmesebben tudjon dolgozni a termékkel. És így tovább...

A minap például egy cégvezetővel beszéltem, aki hosszú évek óta készít komoly programrendszereket mind windowsos, mind linuxos környezetben. Beszélgetésünk során szóba került ez a kérdés is, hogy vajon neki milyen tapasztalatai vannak a nyílt fejlesztésekkel kapcsolatban. Az egyik legérdekesebb tapasztalatot, amit megosztott velem, el is neveztem kosarasnak: a „nagyhalak” világában egy nagyvállalat számára ajánlott rendszerek közül az igazgató többnyire nem szakmai szempont szerint választ. Sőt gyakran nem is a hozzáértő választ az ajánlatok közül, hanem az igazgató „egység sugarú D-kosaras titkárnője”. Más szavakkal megfogalmazva, nem biztos, hogy a program használhatósága a fontos, sokkal inkább az, hogyan adjuk el. És itt van az egyik legnagyobb különbség a két tábor között.

Nyílt rendszert árusítani?

A Debian és az RPM alapú Linux-változatok közti egyik legnagyobb különbség, hogy a Debian mögött nem áll egy óriás-cég, amelyik – sok más tevékenysége mellett – erős reklámtevékenységet folytat, tárgyal, kiejárl, megken, rábeszél, elad. De ha összehasonlítjuk a linuxos cégeket a piacon zsíros falatokat lenyelő cégekkel, gyakran egy alapvető különbséggel találkozunk: a nyílt megoldásokat szállító cégek nem abból akarnak megélni, hogy kijárnak a vevőhöz, újratelepítgetnek, majd a vastagabbik ceruzájukkal megírják a számlát. Egy hozzáértő szakember egyszer üzembe helyez egy gépet, és utána néha rászagol, bírják-e még az alaplap kondenzátorai. Meg lehet akkor élni a nyílt rendszerekből? Nézőpont kérdése: attól függ, mit tekintünk célnak. Ha a cél az, hogy minél nagyobb összegeket kérjünk el ugyanazért a termékért, akkor a nyílt fejlesztésekben szereplők éhhalálra vannak ítélve. Ebből a szempontból arany életük van például a könyvelési rendszereket készítő és forgalmazó cégeknek: a tíz éve megalkotott, azóta már relikviának számító ősszállat-eszközökkel megírt, mondjuk paradox táblán dolgozó könyvelőprogram azóta is működik, néha-néha módosítgatunk rajta itt-ott, ezt eladjuk párszáz fel-

használónak, majd évente kiszabunk egy arcátlan követési díjat. De természetesen külön díjat szabhatunk ki például azért, ha valaki több tételt akar felvinni. Igaz, a rendszernek hihetetlenül mindegy, hogy tízezer vagy kétszázötvenezer tétel kerül feldolgozásra, de valamiből meg kell élni, nem igaz? Ezt a folyamatos bevételt biztosító elméletet irigyelte meg sok óriás-cég is, akik már évek óta azon küzdenek, hogy valami fondorlatos módszerrel mindenki folyamatosan köteles legyen nekik nagy pénzeket fizetni a – felhasználók oldaláról nézve azért a szolgáltatásért, amit egyszer már megvásároltak. Nézzük, hogyan épül fel a nyílt megoldásokat szállító cégek világa. A vállalatnál felmerül az igény egy új vezérlőprogramra, ami már többcsapágas gumipitypang-ültetvény locsolórendszerét is EU-szabványoknak megfelelően vezérli. A cégnek több lehetősége is van: a) elmegy a boltba, és vásárol egy dobozos locsolóvezérlő programot; b) egy céget bíz meg a fejlesztéssel; illetve c) felvesz néhány programozót belső program fejlesztésére. Melyikkel jár a legjobban? Az a) változat esetén kérdés, hogy a dobozos termék EU-megfelelő program-e. Ha nem, akkor póruul járt. Esetleg visszaveszekedheti a pénzét, vagy reménytelen kísérletet tehet a gyártónál arra, hogy újírják a programot, hogy az ő ültetvényén is használható legyen. Ez a megoldás tehát csak akkor működik, ha a program minden feladatot ellát, és hiba sem merül fel. Erre pedig vajmi kevés remény van. A második lehetőség lényegében még komolyabb kiszolgáltatottságot jelent, hiszen a cég az adott rendszer teljes fejlesztését egy külső cégre bízta. Ebben az esetben végképp a külső cég becsületességén és piaci megbízhatóságán múlik, mikor mennyit számláz ki – a kód itt is zárt, és a felület meg azt a biztonságot sem adja meg, hogy „a dobozos terméket legalább többen használják, így ha valami gond van vele, a gyártó jó eséllyel már rendelkezik megoldással. Sőt ha az elején elfogadott rendszertervből hiányzik mondjuk a gumipitypang számára fontos hőmérséklet figyelése, könnyen kaphat a cég a beszállítótól pármillió forintos ajánla-

tot egy hőmérsékletet-is-figyelő programváltozat elkészítésére. Ennek pedig többnyire az a vége, hogy a cég felvesz egy nyugdíjast, akinek az a feladata, hogy óránként megnézze a hőmérőt. Lényegében a harmadik változat tűnik a legérdekesebbnek, feltéve, hogy a vállalat az Ég vezérletével megtalálja azt a pár programozót, akik pont az ő számára megfelelő eszközökkel szeretnek dolgozni, rendszerszervező ismeretekkel



is rendelkeznek, remek kódolók és hosszú távon egy helyen akarnak dolgozni. Mondjuk fél év alatt kifejlesztik a Pitypang 1.0 változatot, ami már működik is, és folyamatos üzemeltetés mellett igény szerint továbbfejlesztik. A veszélyeket itt is nyugodtan sorolhatjuk. Egyrészt a megfelelő ember megtalálása nem két nap. Másrészt szakember kell ahhoz, hogy eldöntsük, valóban jó embert találtunk-e (és a szakember szintén tévedhet); harmadrészt, ha tévedünk, azt csak fél, egy, vagy éppen két év múlva tudjuk meg. Negyedrészt fent kell tartanunk egy teljes fejlesztőkörnyezetet, és sajnos még fehér hollónak számít az a programozó, aki egy ilyen Pitypang 1.0 rendszert esetleg GNU-eszközökkel akar fejleszteni. Tehát mint láttuk, mindhárom modellnek komoly veszélyei vannak. Nem beszélve arról, hogy mindhárom modellnél szerepel még egy erős veszélyforrás: mi van, ha a beszállító csődbe megy, és úgy dönt, hogy nem nyújt megfelelő fejlesztést, támogatást vagy a programozó feláll az asztaltól, mert jobb ajánlatot kapott egy Borókaföld 1.0 megírásá-

sára? Baj. Méghozzá nem kicsi. Az amerikaiak jellemző módon erre a helyzetre is kitaláltak egy nevet, úgy hívják: kamionhatás: „Mi lesz, ha a programozónkat, aki az egész rendszert fejlesztette és üzemeltette, holnap elcsapja egy kamion?” A három modell tulajdonságait és veszélyeit szem előtt tartva több cég egy negyedik modellt, a nyílt fejlesztésű modellt választotta, melynek kialakulásához tehát piaci igények vezettek.

összehozlak titeket, és elkészítünk egy olyan rendszert, ami mindhármótok igényét kielégíti. Így igaz, hogy három helyzetfelmérés van, de már csak egy rendszertervet kell készíteni, nem is beszélve arról, hogy a kipróbálás párhuzamosan mehet, így a kényesebb feladatoknak csak az egyharmada terhelődik rád. Sőt az összköltségnek is csak az egyharmadát kell állnod. A teljes fejlesztést GNU-eszközökkel végezzük, és utána a rendszert kódostul, kutyafülestül mindhárman megkapjátok.” Ez is egy lehetőség, de most jön az ötödik változat, a tényleges „nyílt rendszer”: „Mondok neked még egy lehetőséget. A fejlesztés után az alapkódot nyissátok meg a külvilág felé! Így amit elkészítettünk, azt a világon bárhol használhatják, ahol a tiedhez hasonló feladatokat kell ellátni. Hogy ez miért jó neked, hiszen minden ellenfeled is elkezdheti használni? Hát éppen azért jó! Na figyelj, ha Kelet-Üzbegisztánban is ezt a rendszert kezdi el használni egy pitypangos, és neki szüksége van egy páratartalom-vezérlő modulra is, megírhatja magának, majd felküldi ezt is a rendszerbe. Te nem is gondoltál rá, hogy a páratartalomra is figyelj, de ott a modul, kipróbálsz, és rájössz, mennyivel szebbek lesznek a pitypangjaid. Kész nyeremény, nem?” Természetesen ez az elmélet sok kérdést felvet; a támogatást másként kell megoldani, a követés is más rendszert igényel, valamint a hivatalos szervek felé is másképp kell fordulni. Nincs sajnos hírem olyan számlázórendszeréről, amelyet az APEH elfogadott volna, és a forrása nyílt volna. De kíváncsi vagyok, vannak-e olyan cégek idehaza, akik nyílt fejlesztésben gondolkodnak? Mondjuk egy nyílt számlázórendszeren?



Zsolt György

(Szy.Gyorgy@linuxvilag.hu)
A Linuxvilág főszerkesztője,
a Kiskapu Kiadó vezetője.
Mindenki levelét
örömmel várja.

Felhívás!

Ha vállalatod nyílt fejlesztésben gondolkodik, mondjuk egy számlázó- vagy raktárkezelő rendszer kifejlesztésén, kérlek, írd nekem! Háttha össze tudjuk hozni! Szívesen adnék hírt az első ilyen típusú komolyabb fejlesztésről!