

Old Man GURU Magazine

*Wychodzi bardzo nieregularnie, kiedy wydaje mi się,
że mam coś ciekawego lub pożytecznego do napisania...*

Numer 34/2013

26 marzec 2013

Wielu zwłaszcza młodszych entuzjastów Linuksa zadaje dość ważne pytanie – dlaczego instytucje kupują „komercyjne” wersje systemów – np. RedHat Enterprise Linux czy też SuSE Linux Enterprise Desktop lub Server, podczas gdy istnieją i są nieodpłatnie dostępne odpowiedniki – CentOS, Fedora lub OpenSuSE? Co więcej – te odpowiedniki często wykorzystują nowocześniejsze technologie (nie dotyczy CentOS) – np. RedHat (RHEL) 6.3 to jądro 2.6.32 zaś Fedora 18 korzysta z wersji 3.8.x, zaś OpenSuSE 12.3 z wersji 3.7, gdy tymczasem wersja SuSE Linux Enterprise Desktop, z której korzystam na swoim notebooku HP 625 ma jądro 3.0.58. Warto podkreślić, że ten system (SuSE Linux Enterprise Desktop) został przeze mnie zakupiony (wersja OEM) wraz z notebookiem jako wersja preinstalowana i objęta jest opieką techniczną HP – co umożliwi mi regularne uaktualnienie systemu poprzez wprowadzanie „Service Packs” oraz gwarantuje pełną kompatybilność systemu operacyjnego ze sprzętem – jednym słowem wszystko działa bez potrzeby jakichkolwiek „walk” z mojej strony.



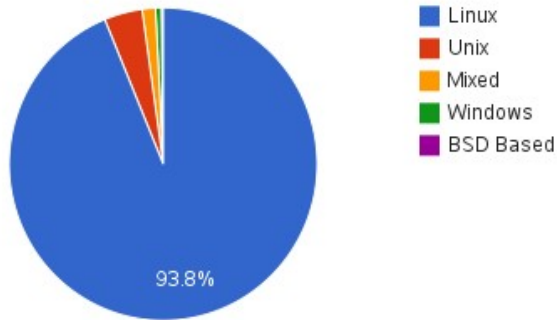
Właściwie na tym można by ten numer „GURU” zakończyć, bo mój osobisty przykład świetnie ilustruje, o co chodzi. Zajmowałem się systemami UNIX od połowy lat osiemdziesiątych ubiegłego wieku (jak to brzmi!), a od ok. 1993 r. Linuksami. Pierwszą moją dystrybucją był nieznany prawie wcale w Polsce Yggdrasil Linux, który chyba jako pierwszy występował także w wersji „Live CD”. Mogę więc świętować mały jubileusz – równo 20 lat z systemem Linux! Ale pomimo że cały czas na co dzień zajmuję się przygotowaniem specjalnych wersji Embedded Linux oraz byłem Autoryzowanym Inżynierem i

Instruktorem SCO, a jestem RedHata to wołałem do pracy kupić komputer z preinstalowanym Linuksem – i to wcale nie tylko dlatego, że mogę pochwalić się specjalną nalepką, która znajduje się na obudowie komputera w tym samym miejscu, gdzie zwykle znajduje się logo „wiodącego systemu operacyjnego” o nazwie zaczynającej się na „W”. Po prostu – ten komputer jest mi potrzebny do prac biurowych, pracy w sieci itp. - do celów „developerskich” używam innych maszyn.

Jak jeszcze pracowałem z SCO, jeden z twórców oprogramowania MPX (Multi Processor Extension) – Alec Parker powiedział mi: „Tomasz, we are super-glue company”. I do dziś w zasadzie nic się nie zmieniło! Firma SCO (i inni dostawcy Uniksov) kupowała różne technologie (choćby stack TCP, rozwiązania desktopu itp.) i „kleiła” je w całość. Plik /etc/copyrights, który występował w wielu systemach UNIX był długą listą autorów modułów, które złożyły się na system operacyjny. Dziś przecież jest tak samo! Każda

dystrybucja - „komercyjna” lub „wolna” to zbiór modułów tworzonych przez wielu autorów lub ich zespoły. Różnica dotyczy licencji – w świecie Linuksa królują licencje otwarte, które nie ograniczają w praktyce (lub ograniczają w bardzo niewielkim stopniu) praw do korzystania z oprogramowania. Rozpowszechnienie się wolnego i otwartego oprogramowania otworzyło nowe możliwości, ale jak każde nowe podejście spowodowało nowe zagrożenia.

Operating system Family System Share



Niewątpliwie największą zaletą systemów rodziny UNIX, a później LINUX jest ogromna elastyczność. Dzięki otwartej strukturze, rozproszonej strategii rozwoju technologii (wielu autorów poszczególnych modułów, które muszą ze sobą współpracować), powszechnemu stosowaniu otwartych standardów itp. możliwe stało się powstawanie różnych dystrybucji. Niestety ta elastyczność, która jest powszechnie doceniana przez

fachowców stała się również największą słabością Linuksa. Stosunkowo mało entuzjastów komputerów wie, że popularność Linuksa na wielkich maszynach (lista www.top500.org) jest większa niż rozpowszechnienie się MS Windows na desktopach!

To właśnie dzięki tej ogromnej elastyczności specjaliści mogą przystosować system operacyjny do pracy na wielkich komputerach o ogromnych zasobach. Na przykład pracująca w Oak Ridge National Lab. maszyna Titan Cray XK7 ma ponad 500000 rdzeni (ponad pół miliona!), pamięć 710 TB oraz pobiera z sieci energetycznej moc ponad 8000 kW! I oczywiście pracuje pod kontrolą Linuksa! Trudno o lepszy przykład skalowalności systemu, bo przecież Linux znakomicie sobie radzi również na starych komputerach z wolnymi procesorami i może się zadowolić nawet kilkunastoma MB pamięci RAM.

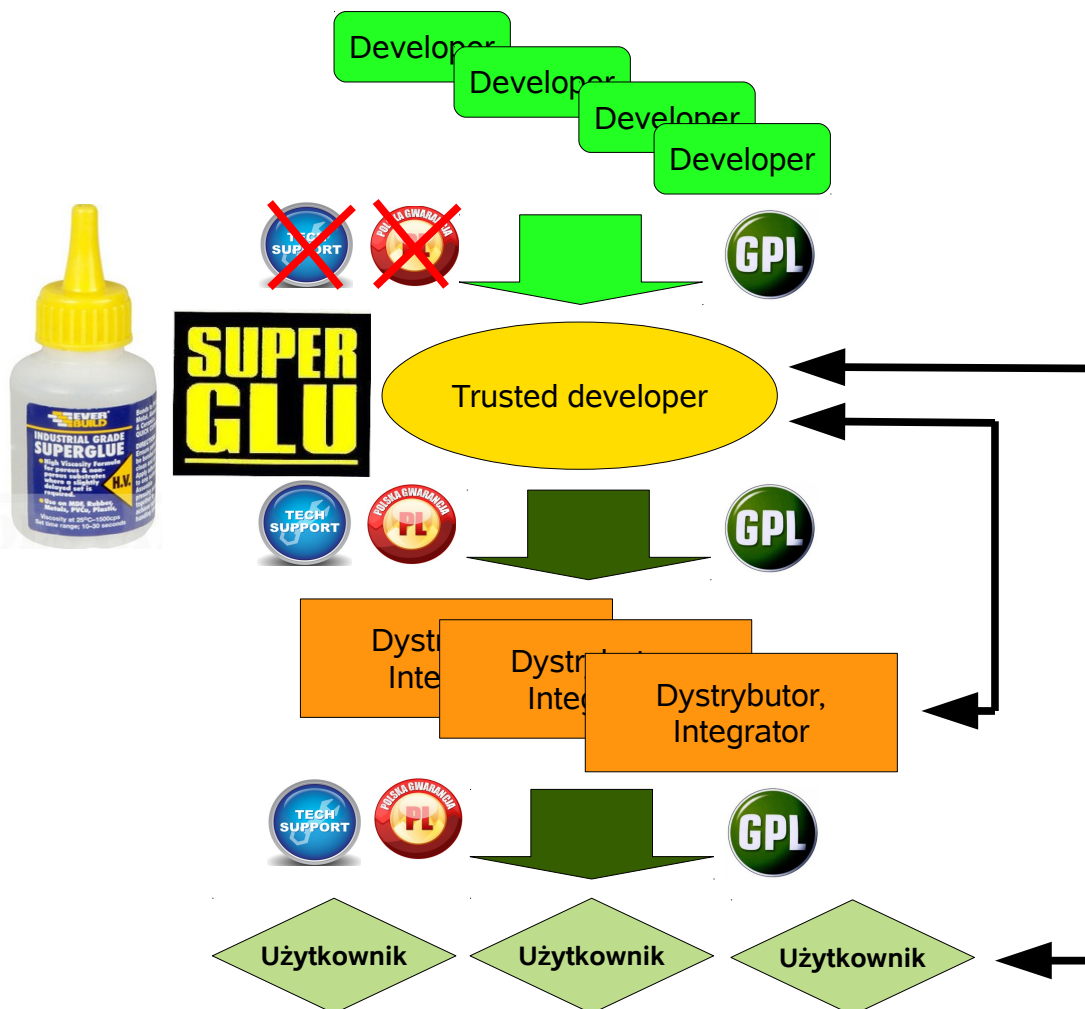
Niestety, co jest zaletą dla fachowca może stać się problemem dla przeciętnego użytkownika. Mnogość dystrybucji, pełny dostęp administracyjny do systemu prowadzi często do kłopotów. Ileż to razy musiałem odpowiadać na pytanie „jaka dystrybucja jest najlepsza?”. Nic się nie zmieniło od czasów Uniksa! Wówczas inicjatywę Open System Foundation (OSF) nazywano „Oppose SUN Forever”! Co gorsza, popularność dystrybucji Linuksa zmienną jest. Jeszcze niedawno bardzo popularna była dystrybucja Slackware (równieśnik „mojego” Yggdrasila) – dziś króluje Ubuntu i jego pochodne. Taka zmienność ma wiele przyczyn, jednak ich omówienie wymaga osobnego opracowania. Skutki są jednak dość poważne – zwłaszcza dla informatyki profesjonalnej. Częste wprowadzanie zmian w nawet w systemach średniej wielkości, które są wykorzystywane do celów profesjonalnych zawsze wiąże się ze sporymi kosztami, które często przewyższają korzyści. Niedawno otrzymałem zadanie opracowania końcówek sieciowych do współpracy ze starym systemem HP/UX, na którym pracuje znakoma aplikacja obsługi szpitala wymagająca specjalnego zestawu czcionek z obsługą semigrafiki (bez wsparcia UTF). Końcówki te służą równocześnie jako terminale graficzne serwerów MS Windows, jednak stary system UNIX spisuje się do dnia dzisiejszego znakomicie i pracujące na nim oprogramowanie jest podstawą pracy szpitala oraz zawiera dane o pacjentach z długiego okresu. Samo działanie szpitala w zasadzie się nie zmienia (za wyjątkiem przepisów NFZ!) - nie ma więc powodu, aby „na siłę” wymieniać sprawny i efektywny system.

Rozwój wolnego i otwartego oprogramowania opiera się na dwóch zasadach:

- pełnej dostępności do kodu oprogramowania, możliwości analizowania jego pracy oraz wprowadzania poprawek,
- ochronie autora przed zobowiązaniami wynikającymi z tytułu udostępnienia oprogramowania – czyli udostępnianiu go „AS IS” bez zobowiązań gwarancyjnych oraz bez obowiązku świadczenia wsparcia technicznego.

Każdy może wykorzystywać takie oprogramowanie w dowolnym celu, może go powielać i przekazywać osobom trzecim – lecz czyni to na własny koszt i ryzyko i nie może dochodzić żadnych praw od jego Autora (developera). Jest to bardzo atrakcyjne dla specjalistów, jednak nie sprawdza się w większości zastosowań – zwłaszcza określanych jako „zastosowania biznesowe”, ponieważ wówczas duże znaczenie przywiązuje się do stabilności rozwiązania oraz gwarancji jego pracy przez dłuższy czas i istnienia podmiotu odpowiedzialnego za świadczenie opieki nad oprogramowaniem (dostęp do poprawek, uaktualnień itp.).

Rolę taką pełni w tym przypadku „zaufany dostawca” (Trusted Developer). „Biznesowy” model rozwoju i dystrybucji wolnego oprogramowania przedstawia poniższy diagram:



„Trusted Developer” wykorzystuje oprogramowanie tworzone i udostępniane na zasadach „AS IS” przez wielu niezależnych deweloperów, scala (skleja „super klejem”) określone moduły w jedną całość i przygotowuje oprogramowanie w formie gotowej (lub prawie gotowej) do użycia oraz zapewnia usługi związane z jego obsługą na poziomie co najmniej zbliżonym do usług oferowanych dla oprogramowania własnościowego (zwanego potocznie komercyjnym). Usługi te najczęściej obejmują:

- świadczenie gwarancji na dostarczany produkt przez dłuższy (kilkuletni) okres,
- objęcie produktu wsparciem technicznym (pomoc przy eksploatacji, gwarancja dostaw poprawek i uaktualnień, dostęp do linii „Help-Desk”),
- przygotowanie i organizację specjalizowanych szkoleń związanych z produktem.

Warto zaznaczyć, że „Trusted Developer” nie może naruszać warunków licencji otwartych – na przykład GPL, na których otrzymuje poszczególne moduły od developerów. Warto podkreślić, że wbrew dość powszechnej opinii licencja GPL nie wymaga udostępniania oprogramowania w Internecie! Uprawnienia wynikające z licencji GPL muszą być natomiast przekazane użytkownikowi, z którym zawierana jest umowa licencyjna. Nie można na przykład wprowadzać ograniczeń z użytkowaniu oprogramowania lub w jego rozpowszechnianiu. Tak więc użytkownik po otrzymaniu oprogramowania na takiej licencji dysponuje „4 wolnościami GNU”.

Kanał dystrybucji otwartego oprogramowania przedstawiłem schematycznie na diagramie. Kanał ten jest i musi być „przeźroczysty” dla warunków licencyjnych (na diagramie symbol GPL). Warunki licencji są przekazywane bez zmian od „Developerów” do użytkowników końcowych. Zysk „Trusted Developera” oraz Dystrybutora lub Integratora wynika wprost z zakresu oferowanych przez nich usług.

Aby zabezpieczyć pokryć koszty świadczenia takich usług i zabezpieczyć swoje interesy „Trusted Developerzy” stosują specjalne umowy subskrypcyjne. Umowy te muszą uwzględniać fakt, że oprogramowanie otwarte może być powielane i instalowane w dowolnej liczbie kopii bez żadnych ograniczeń. Z zasady więc takie umowy zawierają klauzulę, że muszą obejmować wszystkie wykorzystywane przez użytkownika egzemplarze oprogramowania. Technicznie jest to realizowane poprzez rejestrację oprogramowania (a więc podobnie, jak w przypadku oprogramowania własnościowego). Umowy subskrypcyjne nie zabraniają (bo po prostu nie mogą!) korzystania z niezarejestrowanych egzemplarzy oprogramowania – lecz w przypadku wykrycia takiego faktu automatycznie są wypowiedzane.

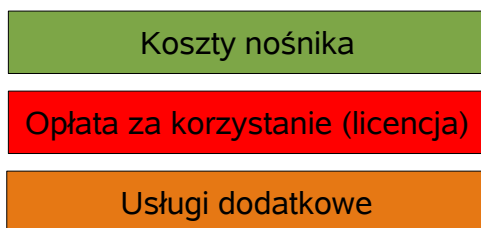
W praktyce wygląda to tak, że jeśli użytkownik korzysta np. z 5 egzemplarzy oprogramowania to aby korzystać z usług z nim związanych konieczne jest posiadanie aktualnych umów subskrypcyjnych na wszystkie egzemplarze. Jeśli wygaśnie umowa choćby na jeden egzemplarz to powinna ona zostać przedłużona – w przeciwnym przypadku wszystkie umowy subskrypcyjne zostaną one automatycznie anulowane.

Jeśli użytkownik zdecyduje się nie przedłużać umów subskrypcyjnych, to może w dalszym ciągu korzystać z oprogramowania, jednakże oczywiście straci prawo do korzystania z usług objętych subskrypcją.

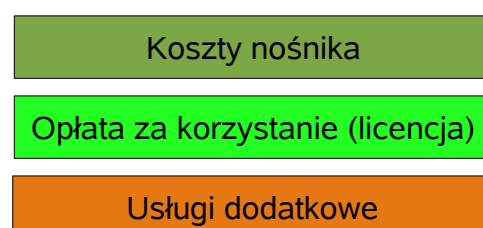
Podobne modele stosuje obecnie wielu dostawców - na przykład firma RedHat, a nawet wielu dostawców oprogramowania komercyjnego (Network Instruments, 2X, NoMachine itp.). Niektórzy dostawcy (np. firma ABA) oferują także ryczałtowe umowy subskrypcyjne obejmujące wszystkie egzemplarze wykorzystywanego oprogramowania.

„Trusted Developer” najczęściej dostarcza oprogramowanie wraz z początkową umową subskrypcyjną, co często powoduje wiele nieporozumień. Tymczasem nie jest to opłata za korzystanie z oprogramowania („Right to Use”) lecz za dostęp do usług dodatkowych! Lepiej zilustrować różnicę pomiędzy oprogramowaniem własnościowym i otwartym na kolejnym diagramie:

Oprogramowanie własnościowe



Oprogramowanie wolne



Koszty wytworzenia nośnika oraz jego dystrybucji tracą na znaczeniu, ponieważ coraz więcej oprogramowania jest dystrybuowane elektronicznie. Nie znaczy to jednak, że całkowicie znikają – pozostaje koszt utrzymania serwera, wynajmu łącza itp. mogą jednak one mogą być pokryte przez sponsorów (reklamy itp.).

Opłaty za korzystanie to podstawowa część kosztów związanych z korzystaniem z oprogramowania własnościowego. Przykładem może być fragment cennika firmy Network Instruments, która otwarcie ujawnia te składniki:

OBSBASE16X	Observer BASE kit (For use with Right To Use (RTU) for Observer Standard, Expert or Suite)	€ 100,00
OBS16XRTU	Observer Standard 16.x Right To Use license (Requires OBSBASE16X)	€ 1 000,00
OBSMAINT	1 Year Observer Standard Maintenance Contract* - Includes upgrades and telephone support	€ 220,00

Widać, że pozycja „Right To Use license” stanowi największy składnik kosztów posiadania i eksploatacji oprogramowania (TCO – Total Costs of Ownership). Jest to typowe dla oprogramowania własnościowego. Warto zwrócić uwagę, że w większości przypadków (wbrew deklaracjom dostawców) opłata za prawa do korzystania nie obejmuje gwarancji i wsparcia technicznego (lub oferowana usługa jest bardzo ograniczona).

W przypadku wolnego oprogramowania opłata „Right to Use license” najczęściej nie jest pobierana (choć sama licencja GPL pobierania takich opłat nie zabrania). Dzięki temu TCO są dla wolnego oprogramowania znacznie niższe.

Wiele osób zadaje w tym miejscu pytanie – „no to po co w ogóle płacić? Samą kopię oprogramowania mogę najczęściej pobrać za darmo, a posiadam (lub przynajmniej uważam, że posiadam) wystarczającą wiedzę, aby go eksploatować”.

W przypadku instalacji Linuksa na wielkim klastrze obliczeniowym lub farmie serwerów internetowych takie podejście jest uzasadnione, ponieważ system zainstaluje, skonfiguruje i będzie nadzorował fachowiec. Musi on być jednak do ciągłej dyspozycji – pełni więc podobną rolę jak „Trusted Developer”. Model ten jest właściwie jedyny, jeśli chcemy w pełni wykorzystać elastyczność systemu i potrzebujemy indywidualnego rozwiązania, lecz całkowicie zawodzi w przypadku dużej liczby instalacji. W takich przypadkach znakomicie sprawdza się podejście zbliżone do kanałów dystrybucyjnych wykorzystywanych przez komercyjnych dostawców oprogramowania. Wzorcowym przykładem takiego podejścia jest dystrybucja systemu RedHat. Szczegóły subskrypcji RedHat zawierają odpowiednie umowy oraz informacje dostępne w Internecie (np. na stronie Linux Polska). Warto jednak ponownie podkreślić, że posiadanie aktualnej subskrypcji nie jest równoważne z licencją „Right to Use”. W przypadku wolnego oprogramowania prawa użytkownika do korzystania z oprogramowania określa bowiem licencja – np. General Public License (GPL), zaś umowa subskrypcyjna zawiera jedynie zobowiązania do świadczenia wsparcia dla użytkownika oprogramowania.

Podobny model stosuje firma ABA – oprogramowanie rodziny ABA-X4 jest oprogramowaniem otwartym, wykorzystującym wiele modułów udostępnianych przez wielu deweloperów z całego Świata, które zostały „sklejone” w całość przez firmę ABA. Działa ona jako „Trusted Developer” oferując szeroki zakres usług, uaktualnienia, poprawki itp. To właśnie dzięki takiemu modelowi „czas życia” terminali z oprogramowaniem firmy ABA jest tak długi (w wielu przypadkach już ponad 10 lat!) i użytkownicy eksploatujący terminale dostarczone w 2003 i posiadający ważne kontrakty wsparcia technicznego mają w 2013 dostęp do najnowszej wersji oprogramowania (z obsługą protokołu RDP 7, a więc dostępem do maszyn MS Windows 7 i Windows Server 2008/12), która może być wykorzystywana na starych platformach sprzętowych. Jest to najlepszy dowód, że przyjęty model rozwoju i dystrybucji oprogramowania oparty o istnienie „Trusted Developera” sprawdza się znakomicie w praktyce i gwarantuje użytkownikom ciągłość wsparcia technicznego nieosiągalną dla rozwiązań indywidualnych.

Na zakończenie anegdotka (ale autentyczna):

Odwiedził mnie kiedyś w firmie V-Prezes sporego Banku, który był naszym klientem i zadał pytanie: Panie Tomku, jak „zbackupować” Jarka?. Zdziwiłem się, ponieważ Jarek był świetnym informatykiem w tym Banku i twórcą kilku znakomitych rozwiązań opartych o wolne i otwarte oprogramowanie.

Jednak obawy osoby odpowiedzialnej w zarządzie Banku za dział IT były uzasadnione – otóż rozwiązania wdrożone przez Jarka były bardzo wyrafinowane i właściwie nikt poza Jarkiem ich nie znał. Po prostu działały – ale w sumie tak, jak zamknięte oprogramowanie komercyjne, jednak poza Jarkiem nie istniała żadna organizacja, która mogłaby w razie potrzeby zapewnić Bankowi odpowiednią pomoc. Problem „back-upu” Jarka nie był więc problemem „wydumany” i istniał rzeczywiście.

Jeśli rozejrzeć się wśród wielu wdrożeń znajdziemy wiele podobnych sytuacji. Sprawdzony model „Trusted Developer” pozwala uniknąć problemów związanych z „backupem” Panów Jarków. I to jest odpowiedź na zadane na początku tego „Guru” pytanie.