

Wolne oprogramowanie

W przeciwieństwie do zamkniętego oprogramowania, wolne oprogramowanie (ang. *free software*) posiada jasną definicję czym jest. Określenie to po raz pierwszy zostało oficjalnie użyte najprawdopodobniej w wiadomości, którą Richard Stallman wysłał do grupy dyskusyjnej net.unix wizards 28 września 1983 roku, ogłaszając w niej rozpoczęcie prac nad projektem GNU. Pierwszą próbą zdefiniowania wolnego oprogramowania był *Manifest GNU*⁸⁹ opublikowany przez Stallmana w miesięczniku *Dr. Dobbs's Journal of Software Tools* w marcu 1985 roku⁹⁰. Napisany w celu pozyskania współpracowników i poparcia dla Projektu GNU, przez wiele lat był podstawowym filozoficznym źródłem dla ruchu wolnego oprogramowania. Obecnie najbardziej precyzyjną definicję wolnego oprogramowania można przeczytać na stronach Projektu GNU⁹¹. Jest to oficjalna **definicja wg Free Software Foundation (FSF)**. Czytamy w niej:

Wolne oprogramowanie to kwestia wolności, nie ceny. By zrozumieć tę koncepcję, powinniśmy myśleć o «wolności słowa», a nie «darmowym piwie» [ang. «free» znaczy najczęściej «wolny», «swobodny», ale może też oznaczać «darmowy» - przyp. tłum.].

«Wolne oprogramowanie» odnosi się do prawa użytkowników do swobodnego uruchamiania, kopiowania, rozpowszechniania, analizowania, zmian i ulepszania programów. Dokładniej, mówimy o czterech rodzajach wolności użytkowników programu⁹²:

- Wolność uruchamiania programu, w dowolnym celu (wolność 0).
- Wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- Wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- Wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

Oprogramowanie nazywamy wolnym, jeśli wszyscy użytkownicy posiadają w pełni wszystkie te prawa. Zatem, powinniście mieć swobodę rozpowszechniania kopii programu, zmodyfikowanych bądź oryginalnych, za darmo bądź pobierając opłatę za dystrybucję, wszędzie i każdemu. Wolność robienia tego wszystkiego oznacza (między innymi), że nie musicie prosić o pozwolenie ani płacić za nie.

Wolne oprogramowanie należy wyraźnie odróżnić od freeware⁹³. Samo udostępnianie za darmo oraz zezwolenie na redystrybucję programu nie wystarczają do nazwania danego programu wolnym.

Otwarte oprogramowanie

Pokrewnym pojęciem do wolnego oprogramowania jest otwarte oprogramowanie (*open source*, oprogramowanie o otwartym kodzie źródłowym). Określenie to zostało sformułowane 3 lutego 1998 roku, podczas burzy mózgów w Palo Alto w Kalifornii. W owej sesji brali udział Todd Anderson, Christine Peterson (z Foresight Institute), John "maddog" Hall, Larry Augustin (obaj z Linux International), Sam Ockman (z Silicon Valley Linux User's Group), Michael Tiemann i Eric Raymond. Uczestnicy poszukiwali nowej nazwy dla wolnego oprogramowania: takiej, która by nie powodowała niejednoznaczności i nie kojarzyła się tylko z darmowym oprogramowaniem, i która by była bardziej przyjazna dla ludzi biznesu. Christine Peterson

wymyśliła nazwę *open source*, która przez pozostałych uczestników spotkania została uznana za najlepszą.

Podobnie jak wolne oprogramowanie, także otwarte oprogramowanie ma swoją oficjalną definicję, dostępną na stronie Open Source Initiative⁹⁴. **Definicja OSI** została oparta na Wytycznych Debiana dotyczących Wolnego Oprogramowania⁹⁵. Pełna treść definicji brzmi⁹⁶:

„Open source nie oznacza tylko dostępu do kodu źródłowego. Warunki dystrybucji oprogramowania open source muszą być zgodne z następującymi kryteriami:

1. Swoboda redystrybucji

Licencja nie może ograniczać swobody którejkolwiek ze stron do sprzedawania lub rozdawania oprogramowania jako elementu szerszej dystrybucji zawierającej programy z różnych źródeł. Licencja nie może wymagać pobierania honorariów lub innych opłat od takiej sprzedaży.

2. Kod źródłowy

Do programu musi być dołączony kod źródłowy, a licencja musi zezwalać na dystrybucję zarówno w postaci kodu źródłowego, jak i skompilowanej. Jeśli któryś produkt nie jest rozprowadzany wraz z kodem źródłowym, musi istnieć dobrze udokumentowany sposób uzyskania tego kodu źródłowego za cenę nie przekraczającą rozsądnych kosztów wykonania kopii – najlepiej poprzez darmowe pobranie z Internetu. Kod źródłowy musi być dostępny w zalecanej postaci, pozwalającej na prostą modyfikację. Nie jest dozwolone celowe gmatwanie kodu źródłowego. Formaty pośrednie, takie jak wynik działania preprocesora lub translatora, nie są dozwolone.

3. Dzieła pochodne

Licencja musi zezwalać na dokonywanie zmian oraz tworzenie dzieł pochodnych. Musi również umożliwiać dystrybucję takich dzieł na tych samych warunkach, jakie opisuje licencja oryginalnego oprogramowania.

4. Spójność kodu źródłowego autora

Licencja może ograniczać dystrybucję kodu źródłowego w zmodyfikowanej postaci tylko wtedy, jeśli dozwolona jest przy tym dystrybucja «poprawek» (ang. patch) wraz z kodem źródłowym, za pomocą których program jest potem modyfikowany w trakcie kompilacji. Licencja musi jawnie zezwalać na dystrybucję oprogramowania skompilowanego ze zmodyfikowanego kodu źródłowego. Licencja może wymagać, aby dzieła pochodne nosiły inną nazwę lub numer wersji niż oprogramowanie oryginalne.

5. Niedozwolona dyskryminacja osób i grup

Licencja nie może dyskryminować jakichkolwiek osób czy grup.

6. Niedozwolona dyskryminacja obszarów zastosowań

Licencja nie może zabraniać wykorzystywania programu w jakimś konkretnym obszarze zastosowań. Na przykład, nie może zabraniać wykorzystania programu w sposób komercyjny lub używania go do badań genetycznych.

7. Dystrybucja licencji

Określenie praw dołączone do programu musi obowiązywać wszystkich, którzy otrzymują oprogramowanie bez konieczności przestrzegania przez te osoby dodatkowych licencji.

8. Licencja nie może obejmować konkretnego produktu

Określenie praw dołączone do programu nie może zależeć od tego, że dany program stanowi część określonej dystrybucji oprogramowania. Jeśli program został pobrany z takiej dystrybucji i wykorzystywany lub rozprowadzany zgodnie z warunkami licencji, wszystkie osoby do których program trafia powinny posiadać te same prawa, które określone są dla oryginalnej dystrybucji

oprogramowania.

9. Licencja nie może ograniczać stosowania innego oprogramowania

Licencja nie może nakładać ograniczeń na inne oprogramowanie rozprowadzane wraz z oprogramowaniem objętym licencją. Na przykład, nie może wymagać aby wszystkie inne programy rozprowadzane na tym samym nośniku były programami open source.

10. Licencja musi być neutralna technologicznie

Żaden punkt licencji nie może narzucać konkretnej technologii lub stylu interfejsu.

Wolne czy Otwarte?

Obie definicje w dużej mierze się pokrywają, choć definicja OSI jest znacznie bardziej sformalizowana. Zakres pokrywania się definicji FSF i OSI ilustruje Tab. 1.

Tab. 1: Porównanie definicji wg FSF i OSI. (Źródło: opracowanie własne.)

FSI	OSI									
	P.1	P.2	P.3	P.4	P.5	P.6	P.7	P.8	P.9	P.10
Wolność 0					x	x	x			x
Wolność 1		x								x
Wolność 2	x						x	x	x	
Wolność 3	x	x	x							x

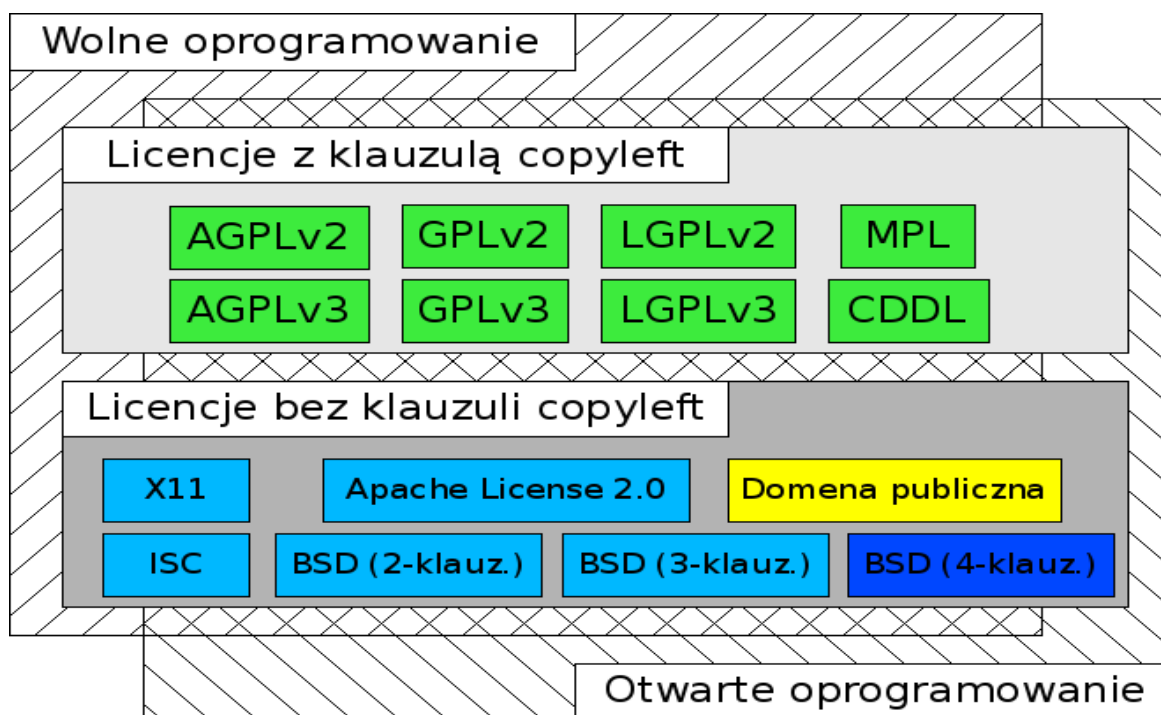
Definicja FSF nic nie wspomina o możliwości dystrybucji modyfikacji tylko w postaci poprawek (p. 4 definicji OSI). Choć definicja OSI wyraźnie zakazuje nakładania wielu typów ograniczeń, to ani razu nie pada w niej słowo „wolność„. Jest to główny punkt sporny pomiędzy zwolennikami FSF i OSI. Obie definicje wyraźnie zezwalają na pobieranie opłat za dystrybucję oprogramowania i jednocześnie wymagają aby w ten sposób pozyskany program można było bezpłatnie redystrybuować. Obie definicje żądają dostarczenia kodu źródłowego programu oraz zezwolenia na jego modyfikację i dystrybucję zmodyfikowanych wersji. W praktyce niemal każdy program open source jest jednocześnie wolnym oprogramowaniem i vice versa. FSF i OSI publikują na swoich stronach listy licencji zgodnych z ich definicjami⁹⁷. OSI dodatkowo przyznaje znak „Open Source Initiative certified” („certyfikowane przez OSI”, zob. Rys. 3) wszystkim programom, których licencja jest zatwierdzona przez OSI jako zgodna z definicją oprogramowania open source⁹⁸.



Rys. 3: Znak "OSI certified". (Źródło: <http://opensource.org/trademarks/osi-certified/>)

Wg Stallmana oba terminy opisują niemal tę samą kategorię oprogramowania, ale oznaczają poglądy oparte na fundamentalnie odmiennych wartościach⁹⁹. Dla niego open source to metodologia tworzenia oprogramowania, a wolne oprogramowanie to ruch społeczny. Poza tym zwraca on uwagę, że chociaż twórcom terminu *open source* udało się uniknąć dwuznaczności terminu *free software* (wolny vs darmowy), to wpadli w inną, gorszą pułapkę, bo w powszechnym mniemaniu wielu ludzi (przy zastosowaniu zwykłych reguł języka angielskiego) *open source* oznacza tylko możliwość obejrzenia kodu źródłowego programu, zaś ewentualna możliwość jego modyfikacji nie jest z tym pojęciem powiązana. Stallman nie uważa open source za wroga, tylko za jedno ze skrzydeł tego samego ruchu. Wrogiem dla niego jest zamknięte oprogramowanie. Zaś wg przedstawicieli ruchu open source, taka konfrontacyjna postawa Stallmana jest szkodliwa¹⁰⁰. Otwarty kod źródłowy jest dla nich tylko sposobem na tworzenie lepszego oprogramowania. Rishab Aiyer Ghosh (członek zarządu OSI) zaproponował Free/Libre Open-Source Software (w skrócie FLOSS, ew. FOSS) jako neutralne pojęcie, bez przechylania się na którąkolwiek ze stron, ale popularność tego terminu jest niska.

Licencje wolnego oprogramowania można podzielić ze względu na obecność klauzuli *copyleft*. Ilustruje to Rys. 4.



Rys. 4: Podział licencji wolnego oprogramowania. (Źródło: opracowanie własne.)

Przypisy:

89.R. M. Stallman, L. Lessig, J. Gay, op. cit., s. 31.

90.[http://www.math.utah.edu/ftp/pub/tex/bib/toc/dr-dobbs-1980.html#10\(3\):March:1985](http://www.math.utah.edu/ftp/pub/tex/bib/toc/dr-dobbs-1980.html#10(3):March:1985)

91.R. M. Stallman, „Czym jest Wolne Oprogramowanie?”, tłum. Grupa tłumaczy witryny Projektu GNU, <http://www.gnu.org/philosophy/free-sw.pl.html>

92.Stallman w swojej definicji numeruje wolności od zera, tak jak programiści numerują np. zmienne przy pisaniu swoich programów.

93.Zob. rozdz. 2.1.2.

94.Oficjalna definicja jest dostępna na stronie OSI: <http://opensource.org/docs/osd>

95.http://www.debian.org/social_contract#guidelines

96.Polskie (nieoficjalne) tłumaczenie pochodzi z archiwalnej wersji strony OSI:

<http://www.free-soft.org/mirrors/www.opensource.org/docs/osd-polish.php>

Punkt 10 definicji: tłumaczenie własne.

97. <http://www.fsf.org/licensing/licenses/>

<http://www.opensource.org/licenses/category>

98. http://www.opensource.org/docs/certification_mark.html

99. <http://www.gnu.org/philosophy/open-source-misses-the-point.html>

100. <http://opensource.org/history>

Autor: Przemysław Kulczycki

Tekst opublikowany jest na licencji [Creative Commons Uznanie Autorstwa 2.5](#), co w skrócie oznacza, że możesz go dowolnie modyfikować, publikować i rozpowszechniać również dla użytku komercyjnego. Jeśli skorzystasz z tego tekstu prosimy Cię jedynie o link zwrotny do wortalu jakilinux.org i informację o autorze oraz o tym, że został on pierwotnie opublikowany w naszym serwisie.