

# Pingwin kontra Okna, czyli Wolne Oprogramowanie kontra...?

Łukasz Jachowicz

lukasz@jachowicz.com

Polska Grupa Użytkowników Linuksa – Ruch Wolnego Oprogramowania  
serwis informacyjny 7thGuard.net

# Pingwin kontra okna?

- Nie chodzi o konflikt produktów.

# Pingwin kontra okna?

- Nie chodzi o konflikt produktów.
- Nie chodzi o jakikolwiek konflikt.

# Pingwin kontra okna?

- Nie chodzi o konflikt produktów.
- Nie chodzi o jakikolwiek konflikt.
- Zmienia się podejście do tworzenia oprogramowania.

# Czym jest Wolne Oprogramowanie?

- Wolność uruchamiania.

# Czym jest Wolne Oprogramowanie?

- Wolność uruchamiania.
- Wolność dystrybucji.

# Czym jest Wolne Oprogramowanie?

- Wolność uruchamiania.
- Wolność dystrybucji.
- Wolność badania działania.

# Czym jest Wolne Oprogramowanie?

- Wolność uruchamiania.
- Wolność dystrybucji.
- Wolność badania działania.
- Wolność poprawiania i rozpowszechniania poprawek.



# Otwarty model oprogramowania

- Dostępne wczesne wersje programów.

# Otwarty model oprogramowania

- Dostępne wczesne wersje programów.
- Otwartość dyskusji.

# Otwarty model oprogramowania

- Dostępne wczesne wersje programów.
- Otwartość dyskusji.
- Dostępny kod źródłowy.

# Otwarty model oprogramowania

- Dostępne wczesne wersje programów.
- Otwartość dyskusji.
- Dostępny kod źródłowy.
- Łatwość wprowadzania poprawek i utrzymywania własnej wersji.

# Otwarty model oprogramowania

- Dostępne wczesne wersje programów.
- Otwartość dyskusji.
- Dostępny kod źródłowy.
- Łatwość wprowadzania poprawek i utrzymywania własnej wersji.
- Możliwość 'rozwidlania' projektów.

# Otwarty model oprogramowania

- Dostępne wczesne wersje programów.
- Otwartość dyskusji.
- Dostępny kod źródłowy.
- Łatwość wprowadzania poprawek i utrzymywania własnej wersji.
- Możliwość 'rozwidlania' projektów.
- Możliwość komercyjnej dystrybucji.

# Otwarty model oprogramowania

- Dostępne wczesne wersje programów.
- Otwartość dyskusji.
- Dostępny kod źródłowy.
- Łatwość wprowadzania poprawek i utrzymywania własnej wersji.
- Możliwość 'rozwidlania' projektów.
- Możliwość komercyjnej dystrybucji.
- Wolność działania.

# Główne zalety otwartego modelu

- Niezależność od producenta.



# Główne zalety otwartego modelu

- Niezależność od producenta.
- Możliwość wpływania na kształt programu.

# Główne zalety otwartego modelu

- Niezależność od producenta.
- Możliwość wpływania na kształt programu.
- Łatwość wprowadzania własnych poprawek.

# Główne zalety otwartego modelu

- Niezależność od producenta.
- Możliwość wpływania na kształt programu.
- Łatwość wprowadzania własnych poprawek.
- Zwiększone bezpieczeństwo.

# Główne zalety otwartego modelu

- Niezależność od producenta.
- Możliwość wpływania na kształt programu.
- Łatwość wprowadzania własnych poprawek.
- Zwiększone bezpieczeństwo.
- Długowieczność zapisanych danych.

# Główne zalety otwartego modelu

- Niezależność od producenta.
- Możliwość wpływania na kształt programu.
- Łatwość wprowadzania własnych poprawek.
- Zwiększone bezpieczeństwo.
- Długowieczność zapisanych danych.
- Możliwość wymiany danych z innymi użytkownikami.

# Ograniczenia (?) otwartego modelu

- Konieczność zmiany modelu biznesowego i sposobu myślenia.

# Ograniczenia (?) otwartego modelu

- Konieczność zmiany modelu biznesowego i sposobu myślenia.
- Niektóre licencje nakazują dawać kolejnym użytkownikom te same prawa, które my otrzymaliśmy.

# Ograniczenia (?) otwartego modelu

- Konieczność zmiany modelu biznesowego i sposobu myślenia.
- Niektóre licencje nakazują dawać kolejnym użytkownikom te same prawa, które my otrzymaliśmy.
- Zwiększona konkurencja.



# Ograniczenia (?) otwartego modelu

- Konieczność zmiany modelu biznesowego i sposobu myślenia.
- Niektóre licencje nakazują dawać kolejnym użytkownikom te same prawa, które my otrzymaliśmy.
- Zwiększona konkurencja.
- Brak gwarancji rozwoju produktu.

# Ograniczenia (?) otwartego modelu

- Konieczność zmiany modelu biznesowego i sposobu myślenia.
- Niektóre licencje nakazują dawać kolejnym użytkownikom te same prawa, które my otrzymaliśmy.
- Zwiększona konkurencja.
- Brak gwarancji rozwoju produktu.
- Wielość standardów programistycznych.

# Czego potrzebuje użytkownik?

- Wysokiej jakości.

# Czego potrzebuje użytkownik?

- Wysokiej jakości.
- Bezpieczeństwa.

# Czego potrzebuje użytkownik?

- Wysokiej jakości.
- Bezpieczeństwa.
- Niezależności od dostawcy.

# Czego potrzebuje użytkownik?

- Wysokiej jakości.
- Bezpieczeństwa.
- Niezależności od dostawcy.
- Niskiej ceny.

# Czego potrzebuje użytkownik?

- Wysokiej jakości.
- Bezpieczeństwa.
- Niezależności od dostawcy.
- Niskiej ceny.
- Długowieczności danych.

# Czego potrzebuje użytkownik?

- Wysokiej jakości.
- Bezpieczeństwa.
- Niezależności od dostawcy.
- Niskiej ceny.
- Długowieczności danych.
- Możliwości wymiany z innymi użytkownikami.



# Czy dziś to otrzymuje?

- Wysoka jakość.
- Bezpieczeństwo.
- Niezależność od dostawcy.
- Niska cena.
- Długowieczność danych.
- Możliwość wymiany z innymi użytkownikami.

# Czego potrzebuje programista

- Dostępu do wiedzy.

# Czego potrzebuje programista

- Dostępu do wiedzy.
- Możliwości pracy.

# Czego potrzebuje programista

- Dostępu do wiedzy.
- Możliwości pracy.
- Środowiska programistycznego.

# Czego potrzebuje programista

- Dostępu do wiedzy.
- Możliwości pracy.
- Środowiska programistycznego.
- Stabilności wykorzystywanych rozwiązań.

# Czy zamknięty kod to daje?

- Borland (największy producent narzędzi dla programistów) wycofuje się z tego rynku.

# Czy zamknięty kod to daje?

- Borland (największy producent narzędzi dla programistów) wycofuje się z tego rynku.
- Nie można rozwijać bibliotek programistycznych bez dostępu do ich kodu.

# Czy zamknięty kod to daje?

- Borland (największy producent narzędzi dla programistów) wycofuje się z tego rynku.
- Nie można rozwijać bibliotek programistycznych bez dostępu do ich kodu.
- Kto z nas pamięta Ami Pro? ChiWitera? TAGa? Word Perfect?



# Czy oprogramowanie jest wieczne?

- Borland i narzędzia dla programistów.

# Czy oprogramowanie jest wieczne?

- Borland i narzędzia dla programistów.
- Lotus (wcześniej: Samna Corp.) i Ami Pro

# Czy oprogramowanie jest wieczne?

- Borland i narzędzia dla programistów.
- Lotus (wcześniej: Samna Corp.) i Ami Pro
- IBM i system operacyjny OS/2

# Czy oprogramowanie jest wieczne?

- Borland i narzędzia dla programistów.
- Lotus (wcześniej: Samna Corp.) i Ami Pro
- IBM i system operacyjny OS/2
- Commodore, Atari, Sinclair Research, Amstrad, Cray Research, Digital Equipment Corporation...

# Otwarty model w praktyce

- Systemy operacyjne: Linux, \*BSD.
- Serwery: Apache (www),  
Exim/Postfix/Sendmail/qmail/qpopper (poczta).
- Bazy danych: MySQL, PostgreSQL, MaxDB  
(dawniej: SAP DB).
- Narzędzia użytkownika: OpenOffice.org, GIMP,  
Evolution, Firefox.
- Aplikacje sieciowe: systemy zarządzania treścią,  
webmaile, systemy pracy grupowej.
- (Prawie) każda inna kategoria oprogramowania.

# Dziękuję

Niniejsza prezentacja została przygotowana przy użyciu wyłącznie wolnego oprogramowania, a w szczególności pakietu  $\text{\LaTeX}$  działającego pod kontrolą systemu GNU/Linux. Może zostać obejrzana na każdym komputerze z zainstalowaną przeglądarką PDF – nieważne, ile ma lat i czy działa pod systemem Linux, Windows, FreeBSD czy innym.

Łukasz Jachowicz <lukasz@jachowicz.com>